aws

Guía del usuario

Amazon EKS

Amazon EKS: Guía del usuario

Table of Contents

¿Qué es Amazon EKS?

Amazon Elastic Kubernetes Service (Amazon EKS) es un servicio administrado que elimina la necesidad de instalar, operar y mantener su propio plano de control de Kubernetes en Amazon Web Services (AWS). Kubernetes es un sistema de código abierto que automatiza la administración, el escalado y la implementación de aplicaciones en contenedores.

Características de Amazon EKS

Estas son algunas de las principales características de Amazon EKS:

Redes y autenticación seguras

Amazon EKS integra sus cargas de trabajo de Kubernetes con redes de AWS y servicios de seguridad. También se integra con AWS Identity and Access Management (IAM) para proporcionar autenticación para los clústeres de Kubernetes.

Fácil escalado de clústeres

Amazon EKS le permite escalar y reducir verticalmente los clústeres de Kubernetes fácilmente en función de la demanda de las cargas de trabajo. Amazon EKS admite el escalado automático de Pod horizontal en función de la CPU o de métricas personalizadas, y el escalado automático de clústeres en función de la demanda de toda la carga de trabajo.

Experiencia de Kubernetes administrada

Puede realizar cambios en los clústeres de Kubernetes mediante `eksctl`, AWS Management Console, AWS Command Line Interface (AWS CLI), la API, `kubectl` y Terraform.

Alta disponibilidad

Amazon EKS proporciona alta disponibilidad para su plano de control en múltiples zonas de disponibilidad.

Integración con servicios de AWS

Amazon EKS se integra con otros servicios de AWS, proporcionando una plataforma integral para implementar y administrar sus aplicaciones en contenedores. También puede solucionar los problemas de las cargas de trabajo de Kubernetes con diferentes herramientas de observabilidad.

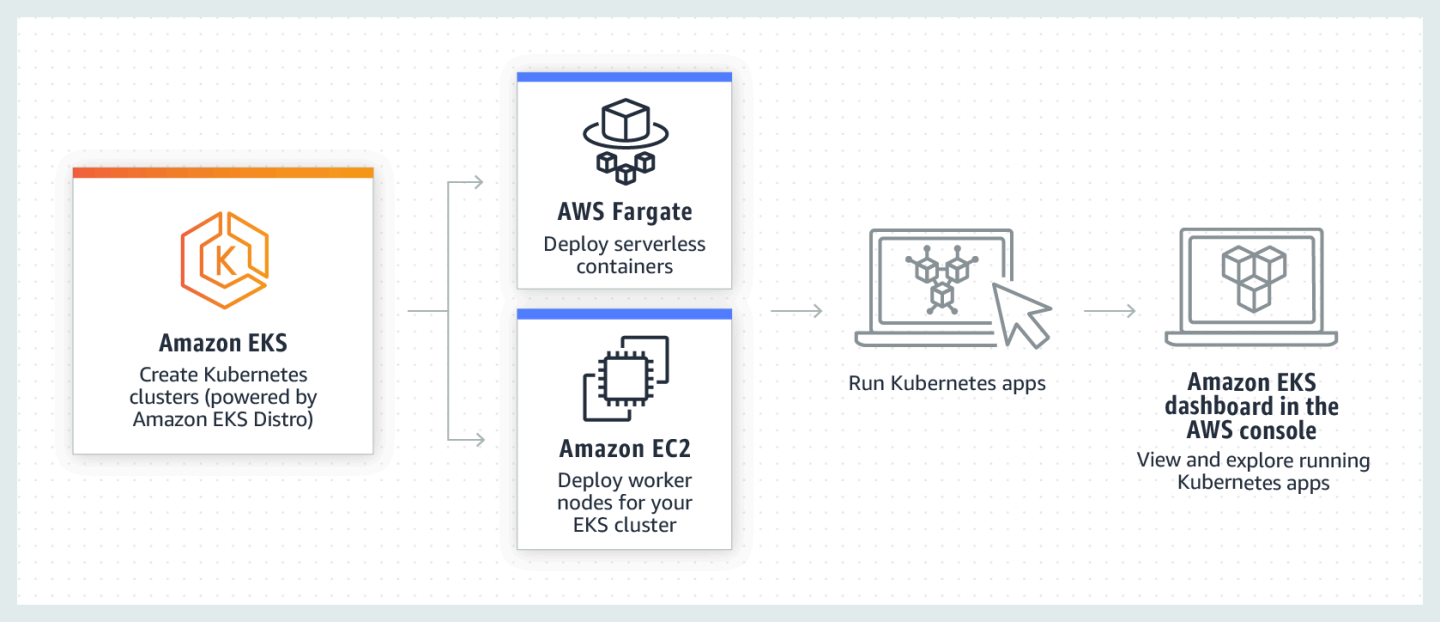
Para obtener más información sobre otras características de Amazon EKS, consulte Características de Amazon EKS.

Introducción a Amazon EKS

Para crear el primer clúster y sus recursos asociados, consulte Introducción a Amazon EKS. En general, empezar a utilizar Amazon EKS implica los siguientes pasos.

1. Creación de un clúster: comience por crear su clúster usando `eksctl`, AWS Management Console, AWS CLI o uno de los SDK de AWS.
2. Elección del enfoque para calcular los recursos: decida entre AWS Fargate, Karpenter, grupos de nodos administrados y nodos autoadministrados.
3. Configuración: configure los controladores y servicios necesarios.
4. Implementación de cargas de trabajo: personalice las cargas de trabajo de Kubernetes para utilizar mejor los recursos y las capacidades del tipo de nodo que elija.
5. Administración: supervise las cargas de trabajo mediante la integración de los servicios de AWS para simplificar las operaciones y mejorar el rendimiento de la carga de trabajo. Puede ver información acerca de las cargas de trabajo mediante la AWS Management Console.

El siguiente diagrama muestra un flujo básico de ejecución de Amazon EKS en la nube. Para obtener más información sobre otras opciones de implementación de Kubernetes, consulte Opciones de implementación.



Precios de Amazon EKS

Un clúster de Amazon EKS consta de un plano de control y de la informática de Amazon Elastic Compute Cloud (Amazon EC2) o Fargate donde ejecuta los Pods. Para obtener más información acerca de los precios del plano de control, consulte Precio de Amazon EKS. Tanto Amazon EC2 como Fargate proporcionan lo siguiente:

instancias bajo demanda

Pague por las instancias que usa por segundos, sin compromisos a largo plazo ni pagos iniciales. Para obtener más información, consulte Precios de Amazon EC2 bajo demanda y Precios de AWS Fargate.

, Savings Plans

Puede reducir los costos comprometiéndose a una cantidad de uso constante, en USD por hora, durante un período de uno o tres años. Para obtener más información, consulte Precios con Savings Plans.

Casos de uso comunes en Amazon EKS

Amazon EKS ofrece sólidos servicios administrados de Kubernetes en AWS que se han diseñado para optimizar las aplicaciones en contenedores. A continuación, se incluyen algunos de los casos de uso más comunes de Amazon EKS y le ayuda a aprovechar sus puntos fuertes para sus necesidades específicas.

Implementación de aplicaciones de alta disponibilidad

Al usar Elastic Load Balancing, puede asegurarse de que sus aplicaciones estén altamente disponibles en varias zonas de disponibilidad.

Creación de arquitecturas de microservicios

Utilice las características de detección de servicios de Kubernetes con AWS Cloud Map o Amazon VPC Lattice para crear sistemas resilientes.

Automatización del proceso de lanzamiento de software

Administre las canalizaciones de integración e implementación continuos (CICD) que simplifican el proceso de creación, prueba e implementación automatizadas de aplicaciones.

Ejecución de aplicaciones sin servidor

Utilice AWS Fargate con Amazon EKS para ejecutar aplicaciones sin servidor. Esto significa que puede centrarse únicamente en el desarrollo de aplicaciones, mientras que Amazon EKS y Fargate se encargan de la infraestructura subyacente.

Ejecución de cargas de trabajo de machine learning

Amazon EKS es compatible con los marcos de machine learning más populares, como TensorFlow, MXNet y PyTorch. Gracias a la compatibilidad con la GPU, puede gestionar incluso tareas complejas de machine learning de forma eficaz.

Implementación uniforme en las instalaciones y en la nube

Utilice Amazon EKS Anywhere para operar clústers de Kubernetes en su propia infraestructura mediante herramientas compatibles con Amazon EKS en la nube.

Ejecución de cargas de trabajo rentables de procesamiento por lotes y macrodatos

Utilice instancias de spot para ejecutar sus cargas de trabajo de procesamiento por lotes y macrodatos, como Apache Hadoop y Spark, a una fracción del costo. Esto le permite aprovechar la capacidad no utilizada de Amazon EC2 a precios reducidos.

Proteger la aplicación y garantizar el cumplimiento

Implemente prácticas de seguridad sólidas y mantenga el cumplimiento con Amazon EKS, que se integra con servicios de seguridad de AWS como AWS Identity and Access Management(IAM), Amazon Virtual Private Cloud (Amazon VPC) y AWS Key Management Service (AWS KMS). Esto garantiza la privacidad y la protección de los datos según los estándares del sector.

Arquitectura de Amazon EKS

Amazon EKS se alinea con la arquitectura general de clústeres de Kubernetes. Para obtener más información, consulte Componentes de Kubernetes en la documentación de Kubernetes. En las siguientes secciones se resumen algunos detalles adicionales de la arquitectura de Amazon EKS.

Plano de control

Amazon EKS garantiza que cada clúster tenga su propio plano de control de Kubernetes. Este diseño mantiene la infraestructura de cada clúster separada, sin superposiciones entre los clústeres o cuentas de AWS. La configuración incluye:

Componentes distribuidos

El plano de control coloca al menos dos instancias de servidor de la API y tres instancias `etcd` que se ejecutan en tres zonas de disponibilidad de AWS dentro de una Región de AWS.

Rendimiento óptimo

Amazon EKS supervisa y ajusta activamente las instancias del plano de control para mantener el máximo rendimiento.

Resiliencia

Si una instancia del plano de control falla, Amazon EKS la reemplaza rápidamente y utiliza una zona de disponibilidad diferente si es necesario.

Tiempo de actividad consistente

Al ejecutar clústeres en varias zonas de disponibilidad, se ha conseguido un Acuerdo de nivel de servicio (SLA) de disponibilidad de puntos de conexión de servidores API.

Amazon EKS utiliza Amazon Virtual Private Cloud (Amazon VPC) para limitar el tráfico entre los componentes del plano de control dentro de un único clúster. Los componentes del clúster no pueden ver ni recibir comunicaciones de otros clústeres u otras cuentas de AWS, excepto según lo autorizado por las políticas de control de acceso basado en roles (RBAC) de Kubernetes.

Cálculo

Además del plano de control, un clúster de Amazon EKS tiene un conjunto de máquinas de trabajo denominadas nodos. La selección del tipo de nodo de clúster de Amazon EKS adecuado es fundamental para cumplir sus requisitos específicos y optimizar la utilización de los recursos. Amazon EKS ofrece los siguientes tipos de nodos principales:

AWS Fargate

Fargate es un motor de procesamiento sin servidor para contenedores que elimina la necesidad de administrar las instancias subyacentes. Con Fargate, usted especifica las necesidades de recursos de su aplicación y AWS aprovisiona, escala y mantiene automáticamente la infraestructura. Esta opción es ideal para los usuarios que priorizan la facilidad de uso y desean concentrarse en el desarrollo y la implementación de aplicaciones en lugar de en la administración de la infraestructura.

Karpenter

Karpenter es un escalador automático de clústeres de Kubernetes flexible y de alto rendimiento que ayuda a mejorar la disponibilidad de las aplicaciones y la eficiencia de los clústeres. Karpenter lanza recursos de computación del tamaño correcto en respuesta a los cambios en la carga de las aplicaciones. Esta opción puede aprovisionar recursos informáticos justo a tiempo que cumplan con los requisitos de su carga de trabajo.

Grupos de nodos administrados

Los grupos de nodos administrados son una combinación de automatización y personalización para administrar un conjunto de instancias de Amazon EC2 dentro de un clúster de Amazon EKS. AWS se encarga de tareas como la aplicación de parches, la actualización y el escalado de los nodos, lo que facilita los aspectos operativos. Paralelamente, se admiten los argumentos de `kubelet` personalizados, lo que abre la posibilidad de aplicar políticas avanzadas de administración de CPU y memoria. Además, mejoran la seguridad mediante roles de AWS Identity and Access Management (IAM) para las cuentas de servicio y, al mismo tiempo, reducen la necesidad de permisos independientes por clúster.

Nodos autoadministrados

Los nodos autoadministrados ofrecen un control total sobre sus instancias de Amazon EC2 dentro de un clúster de Amazon EKS. Usted se encarga de administrar, escalar y mantener los nodos, lo que le proporciona un control total sobre la infraestructura subyacente. Esta opción es adecuada para los usuarios que necesitan un control detallado y una personalización de sus nodos y que están dispuestos a invertir tiempo en la administración y el mantenimiento de su infraestructura.

Conceptos de Kubernetes

Amazon Elastic Kubernetes Service (Amazon EKS) es un servicio administrado de AWS basado en un proyecto de código abierto de Kubernetes. Si bien hay cosas que debe saber sobre cómo el servicio Amazon EKS se integra con la nube de AWS (especialmente cuando crea un clúster de Amazon EKS por primera vez), una vez que esté en funcionamiento, utilizará su clúster de Amazon EKS de la misma manera que lo haría con cualquier otro clúster de Kubernetes. Por lo tanto, para comenzar a administrar clústeres de Kubernetes y a implementar las cargas de trabajo, necesita al menos una comprensión básica de los conceptos de Kubernetes.

Esta página divide los conceptos de Kubernetes en tres secciones: Por qué Kubernetes, Clústeres y Cargas de trabajo. La primera sección describe el valor de ejecutar un servicio de Kubernetes, en

particular como un servicio administrado como Amazon EKS. La sección Cargas de trabajo describe cómo se crean, almacenan, ejecutan y administran las aplicaciones de Kubernetes. La sección Clústeres describe los diferentes componentes que componen los clústeres de Kubernetes y cuáles son sus responsabilidades a la hora de crear y mantener los clústeres de Kubernetes.

Temas

A medida que vaya leyendo este contenido, los enlaces le llevarán a descripciones más detalladas de los conceptos de Kubernetes, tanto en Amazon EKS como en la documentación de Kubernetes, por si quiere profundizar en alguno de los temas aquí tratados. Para obtener más información sobre cómo Amazon EKS implementa el plano de control de Kubernetes y las características informáticas, consulte Arquitectura de Amazon EKS.

¿Por qué Kubernetes?

Kubernetes se diseñó para mejorar la disponibilidad y la escalabilidad al ejecutar aplicaciones contenerizadas esenciales y con calidad de producción. En lugar de ejecutar Kubernetes en una sola máquina (aunque eso es posible), Kubernetes logra esos objetivos al permitir ejecutar aplicaciones en conjuntos de computadoras que pueden expandirse o contraerse para satisfacer la demanda. Kubernetes incluye funciones que le ayudan a:

- Implementar aplicaciones en varios equipos (mediante contenedores implementados en pods)
- Supervisar el estado de los contenedores y reiniciar los que estén defectuosos
- Escalar los contenedores hacia arriba y hacia abajo en función de la carga
- Actualizar los contenedores con nuevas versiones
- Mover recursos entre contenedores
- Equilibrar el tráfico entre las máquinas

Cuando Kubernetes automatiza este tipo de tareas complejas, los desarrolladores de aplicaciones pueden centrarse en crear y mejorar las cargas de trabajo de sus aplicaciones, en lugar de preocuparse por la infraestructura. El desarrollador suele crear archivos de configuración, formateados como archivos YAML, que describen el estado deseado de la aplicación. Esto

podría incluir qué contenedores ejecutar, los límites de recursos, el número de réplicas del pod, la asignación de CPU o memoria, las reglas de afinidad, etc.

Atributos de Kubernetes

Para lograr sus objetivos, Kubernetes cuenta con los siguientes atributos:

- En contenedores: Kubernetes es una herramienta de orquestación de contenedores. Para usar Kubernetes, primero debe tener sus aplicaciones en contenedores. Según el tipo de aplicación, puede ser un conjunto de microservicios, trabajos por lotes o de otras formas. De este modo, sus aplicaciones pueden aprovechar un flujo de trabajo de Kubernetes que abarca un enorme ecosistema de herramientas, en el que los contenedores pueden almacenarse como imágenes en un registro de contenedores, implementarse en un clúster de Kubernetes y ejecutarse en un nodo disponible. Puede crear y probar contenedores individuales en su equipo local con un entorno de tiempo de ejecución de contenedores Docker u otro contenedor antes de implementarlos en su clúster de Kubernetes.
- Escalable: si la demanda de sus aplicaciones supera la capacidad de las instancias en ejecución de esas aplicaciones, Kubernetes podrá escalar verticalmente. Según sea necesario, Kubernetes puede determinar si las aplicaciones requieren más CPU o memoria y responder ampliando automáticamente la capacidad disponible o utilizando una mayor capacidad existente. El escalado se puede llevar a cabo en el pod, si hay suficiente computación disponible para ejecutar más instancias de la aplicación (escalado automático horizontal del pod), o en el nodo, si es necesario instalar más nodos para gestionar el aumento de la capacidad (escalador automático de clústeres o Karpenter). Como la capacidad ya no es necesaria, estos servicios pueden eliminar los pods innecesarios y cerrar los nodos innecesarios.
- Disponible: si una aplicación o un nodo deja de funcionar o no está disponible, Kubernetes puede mover las cargas de trabajo en ejecución a otro nodo disponible. Para forzar el problema, basta con eliminar una instancia en ejecución de una carga de trabajo o un nodo en el que se estén ejecutando sus cargas de trabajo. La conclusión es que las cargas de trabajo se pueden almacenar en otras ubicaciones si ya no se pueden ejecutar donde están.
- Declarativo: Kubernetes utiliza la reconciliación activa para comprobar constantemente que el estado que se declara para el clúster coincida con el estado real. Al aplicar objetos de Kubernetes a un clúster, normalmente a través de archivos de configuración con formato YAML, puede, por ejemplo, solicitar que se inicien las cargas de trabajo que desea ejecutar en su clúster. Puede cambiar las configuraciones más adelante para llevar a cabo otras acciones, como usar una versión posterior de un contenedor o asignar más memoria. Kubernetes hará lo necesario para establecer el estado deseado. Esto puede incluir activar o desactivar los nodos, detener y reiniciar las cargas de trabajo o extraer contenedores actualizados.

- Compatible con la composición: dado que una aplicación suele constar de varios componentes, es recomendable poder administrar un conjunto de estos componentes (que suelen estar representados por varios contenedores) de forma conjunta. Si bien Docker Compose ofrece una forma de hacerlo directamente con Docker, el comando Kompose de Kubernetes puede ayudarte a hacerlo con Kubernetes. Consulte Translate a Docker Compose File to Kubernetes Resources para ver un ejemplo de cómo hacerlo.
- Extensible: a diferencia del software propietario, el proyecto de Kubernetes de código abierto está diseñado para que usted pueda ampliar Kubernetes como desee para satisfacer sus necesidades. Las API y los archivos de configuración están abiertos a modificaciones directas. Se recomienda a terceros que diseñen sus propios controladores para ampliar las características de infraestructura y de usuario final de Kubernetes. Los webhooks le permiten configurar reglas de clúster para hacer cumplir las políticas y adaptarlas a las condiciones cambiantes. Para obtener más ideas sobre cómo ampliar los clústeres de Kubernetes, consulte Extendiendo Kubernetes.
- Portátil: muchas organizaciones han estandarizado sus operaciones de Kubernetes porque les permite administrar todas sus necesidades de aplicaciones de la misma manera. Los desarrolladores pueden usar las mismas canalizaciones para crear y almacenar aplicaciones en contenedores. Luego, esas aplicaciones se pueden implementar en clústeres de Kubernetes que se ejecutan en las instalaciones, en nubes, en terminales de puntos de venta de restaurantes o en dispositivos de IoT dispersos por los sitios remotos de la empresa. Su naturaleza de código abierto permite a las personas desarrollar estas distribuciones de Kubernetes especiales, junto con las herramientas necesarias para administrarlas.

Administrar Kubernetes

El código fuente de Kubernetes está disponible de forma gratuita, por lo que puede instalarlo con su propio equipo y administrar Kubernetes por su cuenta. Sin embargo, autoadministrar Kubernetes requiere una profunda experiencia operativa, y su mantenimiento requiere tiempo y esfuerzo. Por estas razones, la mayoría de las personas que implementan cargas de trabajo de producción eligen un proveedor de nube (como Amazon EKS) o en las instalaciones (como Amazon EKS Anywhere) con su propia distribución de Kubernetes probada y el apoyo de expertos de Kubernetes. Esto le permite librarse de gran parte del trabajo pesado e indiferenciado necesario para el mantenimiento de sus clústeres, que incluye:

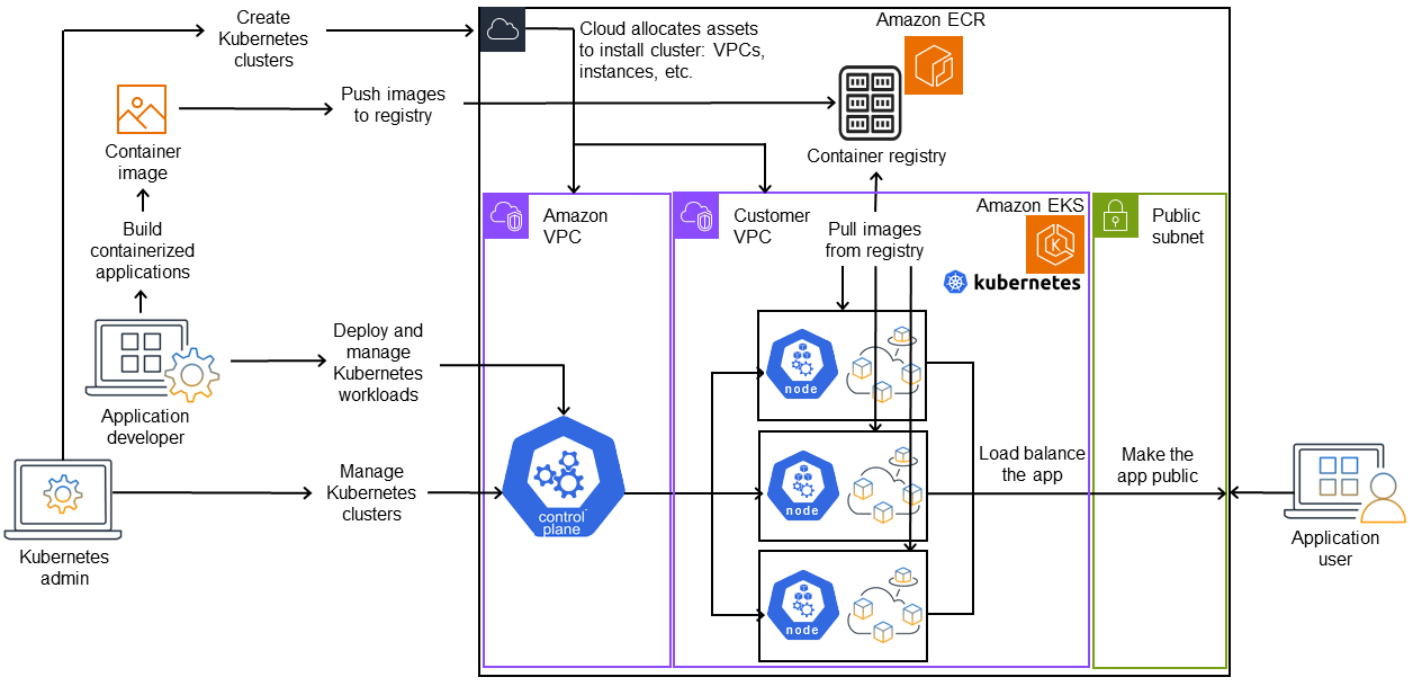
- Hardware: si no tiene hardware disponible para ejecutar Kubernetes según sus necesidades, un proveedor de nube como AWS Amazon EKS puede ahorrarle costos iniciales. Con Amazon EKS, esto significa que puede consumir los mejores recursos de nube que ofrece AWS, incluidas las instancias de computación (Amazon Elastic Compute Cloud), su propio entorno privado

(Amazon VPC), la administración central de identidades y permisos (IAM) y el almacenamiento (Amazon EBS). AWS administra las computadoras, las redes, los centros de datos y todos los demás componentes físicos necesarios para ejecutar Kubernetes. Del mismo modo, no tiene que planificar su centro de datos para gestionar la máxima capacidad en los días de mayor demanda. En el caso de Amazon EKS Anywhere u otros clústeres en las instalaciones de Kubernetes, usted es responsable de administrar la infraestructura utilizada en sus implementaciones de Kubernetes, pero puede confiar en que AWS le ayudará a mantener Kubernetes actualizado.

- Administración del plano de control: Amazon EKS administra la seguridad y la disponibilidad del plano de control de Kubernetes alojado en AWS, que se encarga de programar contenedores, administrar la disponibilidad de las aplicaciones y otras tareas clave, para que pueda centrarse en las cargas de trabajo de sus aplicaciones. Si el clúster se interrumpe, AWS debería disponer de los medios necesarios para restaurarlo a un estado de ejecución. En el caso de Amazon EKS Anywhere, usted mismo administraría el plano de control.
- Actualizaciones probadas: cuando actualiza sus clústeres, puede confiar en Amazon EKS o Amazon EKS Anywhere para proporcionarle versiones probadas de sus distribuciones de Kubernetes.
- Complementos: hay cientos de proyectos diseñados para ampliarlos y trabajar con Kubernetes que puede agregar a la infraestructura de su clúster o utilizarlos para facilitar la ejecución de sus cargas de trabajo. En lugar de crear y administrar esos complementos por su cuenta, AWS proporciona complementos de Amazon EKS que puede usar con sus clústeres. Amazon EKS Anywhere ofrece paquetes seleccionados que incluyen compilaciones de muchos proyectos populares de código abierto. Por lo tanto, no tiene que crear el software ni administrar parches de seguridad, correcciones de errores o actualizaciones críticas. Del mismo modo, si los valores predeterminados se ajustan a sus necesidades, es habitual que se necesite muy poca configuración de esos complementos. Consulte Extend Clusters para obtener más información sobre cómo ampliar su clúster con complementos.

Kubernetes en acción

El siguiente diagrama muestra las actividades clave que llevaría a cabo como administrador de Kubernetes o desarrollador de aplicaciones para crear y usar un clúster de Kubernetes. En el proceso, ilustra cómo los componentes de Kubernetes interactúan entre sí, con la nube de AWS como ejemplo del proveedor de nube subyacente.



Un administrador de Kubernetes crea el clúster de Kubernetes mediante una herramienta específica para el tipo de proveedor en el que se creará el clúster. En este ejemplo, se utiliza la nube de AWS como proveedor, que ofrece el servicio administrado de Kubernetes denominado Amazon EKS.
El servicio administrado asigna automáticamente los recursos necesarios para crear el clúster, lo que incluye la creación de dos nuevas nubes privadas virtuales (Amazon VPC) para el clúster, la configuración de las redes, la asignación de permisos de Kubernetes a las mismas para administrar los activos en la nube, la supervisión de que los servicios del plano de control tengan lugares donde ejecutarse y la asignación de cero o más instancias de Amazon EC2 como nodos de Kubernetes para ejecutar cargas de trabajo de AWSadministra una Amazon VPC por sí misma para el plano de control, mientras que la otra Amazon VPC contiene los nodos del cliente que ejecutan las cargas de trabajo.

En el futuro, muchas de las tareas del administrador de Kubernetes se harán mediante herramientas de Kubernetes como kubectl. Esa herramienta envía las solicitudes de servicios directamente al plano de control del clúster. Por lo tanto, las formas en que se hacen las consultas y los cambios en el clúster son muy similares a las formas en que se harían en cualquier clúster de Kubernetes.

Un desarrollador de aplicaciones que desee implementar cargas de trabajo en este clúster puede llevar a cabo varias tareas. El desarrollador debe crear la aplicación en una o más imágenes de

contenedor y, a continuación, enviarlas a un registro de contenedores al que pueda acceder el clúster de Kubernetes. AWS ofrece Amazon Elastic Container Registry (Amazon ECR) para ese fin.

Para ejecutar la aplicación, el desarrollador puede crear archivos de configuración con formato YAML que indiquen al clúster cómo ejecutar la aplicación, incluidos los contenedores que deben extraerse del registro y cómo empaquetarlos en pods. El plano de control (programador) programa los contenedores en uno o más nodos y el tiempo de ejecución del contenedor en cada nodo extrae y ejecuta los contenedores necesarios. El desarrollador también puede configurar un equilibrador de carga de aplicaciones para equilibrar el tráfico hacia los contenedores disponibles que se ejecutan en cada nodo y exponer la aplicación al mundo exterior para que esté disponible en una red pública. Una vez hecho esto, cualquier persona que desee utilizar la aplicación puede conectarse al punto de conexión de la aplicación para acceder a ella.

En la siguiente sección se analizan los detalles de cada una de estas características, desde la perspectiva de los clústeres de Kubernetes y las cargas de trabajo.

Clústeres

Si su trabajo consiste en iniciar y administrar clústeres de Kubernetes, debe saber cómo se crean, mejoran, administran y eliminan los clústeres de Kubernetes. También debe saber cuáles son los componentes que componen un clúster y qué debe hacer para mantenerlos.

Las herramientas para administrar los clústeres gestionan la superposición entre los servicios de Kubernetes y el proveedor de hardware subyacente. Por esa razón, la automatización de estas tareas la suele llevar a cabo el proveedor de Kubernetes (como Amazon EKS o Amazon EKS Anywhere) mediante herramientas específicas del proveedor. Por ejemplo, para iniciar un clúster de Amazon EKS puede utilizar `eksctl create cluster`, mientras que para Amazon EKS Anywhere puede usar `eksctl anywhere create cluster`. Tenga en cuenta que, si bien estos comandos crean un clúster de Kubernetes, son específicos del proveedor y no forman parte del proyecto de Kubernetes en sí.

Herramientas de creación y administración de clústeres

El proyecto de Kubernetes ofrece herramientas para crear un clúster de Kubernetes manualmente. Por lo tanto, si desea instalar Kubernetes en una sola máquina o ejecutar el plano de control en una máquina y agregar nodos manualmente, puede usar herramientas de la CLI como kind, minikube o kubeadm que se enumeran en Instalar herramientas de Kubernetes. Para simplificar y automatizar todo el ciclo de vida de la creación y administración de clústeres, es mucho más fácil

utilizar herramientas compatibles con un proveedor de Kubernetes establecido, como Amazon EKS o Amazon EKS Anywhere.

En la nube de AWS, puede crear clústeres de Amazon EKS mediante herramientas de la CLI, como eksctl, o herramientas más declarativas, como Terraform (consulte Amazon EKS Blueprints for Terraform). También puede crear un clúster desde la Consola de administración de AWS. Consulte Características de Amazon EKS para obtener una lista de lo que obtiene con Amazon EKS. Las responsabilidades de Kubernetes que Amazon EKS asume por usted incluyen:

- Plano de control administrado: AWS garantiza que el clúster de Amazon EKS esté disponible y sea escalable, ya que administra el plano de control por usted y lo hace disponible en todas las zonas de disponibilidad de AWS.
- Administración de nodos: en lugar de agregar nodos manualmente, puede hacer que Amazon EKS cree nodos automáticamente según sea necesario mediante Grupos de nodos administrados o Karpenter. Los grupos de nodos administrados tienen integraciones con el Escalado automático de clústeres de Kubernetes. Con las herramientas de administración de nodos, puede aprovechar los ahorros de costos, como las instancias de spot y la consolidación de nodos, y la disponibilidad, con las funciones de programación para establecer cómo se implementan las cargas de trabajo y cómo se seleccionan los nodos.
- Redes de clústeres: mediante plantillas de CloudFormation, `eksctl` configura las redes entre los componentes del plano de control y del plano de datos (nodo) del clúster de Kubernetes. También configura puntos de conexión a través de los cuales se pueden llevar a cabo las comunicaciones internas y externas. Consulte De-mystifying cluster networking for Amazon EKS worker nodes para obtener más información. Las comunicaciones entre los pods de Amazon EKS se llevan a cabo mediante Pod Identities de EKS, lo que permite a los pods aprovechar los métodos de administración de credenciales y permisos en la nube de AWS.
- Complementos: Amazon EKS le ahorra tener que crear y agregar componentes de software que se utilizan habitualmente para admitir clústeres de Kubernetes. Por ejemplo, cuando crea un clúster de Amazon EKS desde la consola de administración de AWS, se agregan automáticamente el kube-proxy de Amazon EKS, el complemento para Kubernetes CNI de Amazon VPC y los complementos de CoredNS. Consulte Complementos de Amazon EKS para obtener más información sobre estos complementos, incluida una lista de los que están disponibles.

Para ejecutar sus clústeres en sus propias computadoras y redes en las instalaciones, Amazon ofrece Amazon EKS Anywhere. En lugar de que la nube de AWS sea el proveedor, tiene la opción de ejecutar Amazon EKS Anywhere en plataformas como VMware vSphere, bare metal (proveedor de Tinkerbell), Snow, CloudStack o Nutanix con su propio equipo.

Amazon EKS Anywhere se basa en el mismo software de Amazon EKS Distro que utiliza Amazon EKS. Sin embargo, Amazon EKS Anywhere se basa en diferentes implementaciones de la interfaz Cluster API de Kubernetes (CAPI) para administrar todo el ciclo de vida de las máquinas de un clúster de Amazon EKS Anywhere (como CAPV para vSphere y CAPC para CloudStack). Como todo el clúster se ejecuta en su equipo, usted asume la responsabilidad adicional de administrar el plano de control y hacer copias de seguridad de sus datos (consulte etcd más adelante en este documento).

Componentes del clúster

Los componentes del clúster de Kubernetes se dividen en dos áreas principales: el plano de control y los nodos de trabajo. Los componentes del plano de control administran el clúster y proporcionan acceso a sus API. Los nodos de trabajo (a veces denominados simplemente nodos) proporcionan los lugares donde se ejecutan las cargas de trabajo reales. Los componentes de los nodos consisten en servicios que se ejecutan en cada nodo para comunicarse con el plano de control y ejecutar contenedores. El conjunto de nodos de trabajo de su clúster se denomina plano de datos.

Plano de control

El plano de control consiste en un conjunto de servicios que administran el clúster. Es posible que todos estos servicios se ejecuten en un solo equipo o que estén repartidos en varios equipos. Internamente, se denominan instancias del plano de control (CPI). La forma en que se ejecutan las CPI depende del tamaño del clúster y de los requisitos de alta disponibilidad. A medida que aumenta la demanda en el clúster, un servicio de plano de control puede escalarse para ofrecer más instancias de ese servicio y equilibrar la carga de las solicitudes entre las instancias.

Entre las tareas que llevan a cabo los componentes del plano de control de Kubernetes se incluyen las siguientes:

- Comunicación con los componentes del clúster (servidor de API): el servidor de API (kube-apiserver) expone la API de Kubernetes para que las solicitudes al clúster se puedan efectuar tanto desde dentro como desde fuera del clúster. En otras palabras, las solicitudes para agregar o cambiar los objetos de un clúster (pods, servicios, nodos, etc.) pueden provenir de comandos externos, como las solicitudes de `kubectl` para ejecutar un pod. Del mismo modo, se pueden hacer solicitudes desde el servidor de API a los componentes del clúster, por ejemplo, una consulta al servicio `kubelet` para conocer el estado de un pod.
- Almacene datos sobre el clúster (almacén de valores clave etcd): el servicio etcd desempeña la función fundamental de hacer un seguimiento del estado actual del clúster. Si el servicio etcd

dejara de ser accesible, no podría actualizar ni consultar el estado del clúster, aunque las cargas de trabajo seguirían ejecutándose durante un tiempo. Por ese motivo, los clústeres críticos suelen tener varias instancias del servicio etcd con equilibrio de carga que se ejecutan a la vez y hacen copias de seguridad periódicas del almacén de valores clave etcd en caso de pérdida o corrupción de los datos. Tenga en cuenta que, en Amazon EKS, todo esto se gestiona automáticamente de forma predeterminada. Amazon EKS Anywhere proporciona instrucciones para la copia de seguridad y restauración de etcd. Consulte el modelo de datos de etcd para saber cómo administra etcd los datos.

- Programar los pods por nodos (Programador): las solicitudes para iniciar o detener un pod en Kubernetes se dirigen al Programador de Kubernetes (kube-scheduler). Como un clúster puede tener varios nodos capaces de ejecutar el pod, es el programador quien decide en qué nodo (o nodos, en el caso de las réplicas) debe ejecutarse el pod. Si no hay suficiente capacidad disponible para ejecutar el pod solicitado en un nodo existente, la solicitud fallará, a menos que haya tomado otras medidas. Estas disposiciones podrían incluir la habilitación de servicios como los grupos de nodos administrados o Karpenter, que puedan iniciar automáticamente nuevos nodos para gestionar las cargas de trabajo.
- Mantener los componentes en el estado deseado (Controller Manager): Kubernetes Controller Manager se ejecuta como un proceso daemon (kube-controller-manager) para observar el estado del clúster y hacer cambios en él para restablecer los estados esperados. En concreto, hay varios controladores que vigilan diferentes objetos de Kubernetes, entre los que se incluyen un controlador del ciclo de vida de los nodos, un controlador de conjunto de estados, un controlador de punto de conexión, un controlador de cronjobs, etc.
- Administrar los recursos de la nube (Cloud Controller Manager): Cloud Controller Manager (cloud-controller-manager) gestiona las interacciones entre Kubernetes y el proveedor de la nube que hace las solicitudes de los recursos del centro de datos subyacente. Los controladores administrados por Cloud Controller Manager pueden incluir un controlador de rutas (para configurar las rutas de la red en la nube), un controlador de servicios (para usar los servicios de equilibrio de carga en la nube) y un controlador de nodos (para usar las API de la nube para mantener los nodos de Kubernetes sincronizados con los nodos de la nube).

Nodos de trabajo (plano de datos)

En el caso de un clúster de Kubernetes de un solo nodo, las cargas de trabajo se ejecutan en la misma máquina que el plano de control. Sin embargo, una configuración más normal consiste en tener uno o más sistemas de computación independientes (nodos) dedicados a ejecutar cargas de trabajo de Kubernetes.

Al crear un clúster de Kubernetes por primera vez, algunas herramientas de creación de clústeres permiten configurar un número determinado de nodos para agregarlos al clúster (ya sea al identificar sistemas de computación existentes o hacer que el proveedor cree otros nuevos). Antes de agregar cargas de trabajo a esos sistemas, se agregan servicios a cada nodo para implementar estas características:

- Administrar cada nodo (kubelet): el servidor de API se comunica con el servicio de kubelet que se ejecuta en cada nodo para asegurarse de que el nodo esté registrado correctamente y de que los pods solicitados por el Programador estén funcionando. El kubelet puede leer los manifiestos de los pods y configurar los volúmenes de almacenamiento u otras características que necesiten los pods del sistema local. También puede comprobar el estado de los contenedores que se ejecutan localmente.
- Ejecutar contenedores en un nodo (tiempo de ejecución de contenedores): el tiempo de ejecución de contenedores de cada nodo administra los contenedores solicitados para cada pod asignado al nodo. Esto significa que puede extraer imágenes de contenedores del registro correspondiente, ejecutar el contenedor, detenerlo y responder a las consultas sobre el contenedor. El tiempo de ejecución del contenedor predeterminado es containerd. A partir de la versión 1.24 de Kubernetes, se eliminó la integración especial de Docker (Dockershim), que podía usarse como tiempo de ejecución del contenedor de Kubernetes. Si bien puede seguir usando Docker para probar y ejecutar contenedores en su sistema local, para usar Docker con Kubernetes tendrá que instalar el motor de Docker en cada nodo para usarlo con Kubernetes.
- Administrar las redes entre contenedores (kube-proxy): para poder facilitar la comunicación entre los pods mediante los servicios, Kubernetes necesitaba configurar las redes de pods para rastrear las direcciones IP y los puertos asociados a esos pods. El servicio kube-proxy se ejecuta en todos los nodos para permitir que se produzca la comunicación entre los pods.

Extensión de clústeres

Hay algunos servicios que puede agregar a Kubernetes para que sean compatibles con el clúster, pero no se ejecutan en el plano de control. Estos servicios suelen ejecutarse directamente en los nodos del espacio de nombres kube-system o en su propio espacio de nombres (como suele ocurrir con los proveedores de servicios de terceros). Un ejemplo común es el servicio CoreDNS, que proporciona servicios DNS al clúster. Consulte Discovering built in services para obtener información sobre cómo ver qué servicios de clúster se están ejecutando en kube-system en su clúster.

Hay diferentes tipos de complementos que puede considerar agregar a sus clústeres. Para mantener sus clústeres en buen estado, puede agregar características de observabilidad que le permitan

efectuar tareas como el registro, la auditoría y las métricas. Con esta información, puede solucionar los problemas que se producen, a menudo a través de las mismas interfaces de observabilidad. Algunos ejemplos de estos tipos de servicios son Amazon GuardDuty, CloudWatch,, AWS Distro paraOpenTelemetry, el complemento Amazon VPC CNI para Kubernetes y Grafana Kubernetes Monitoring. En cuanto al almacenamiento, los complementos de Amazon EKS incluyen el controlador CSI de Amazon Elastic Block Store (para agregar dispositivos de almacenamiento en bloques), el controlador CSI de Amazon Elastic File System (para agregar almacenamiento al sistema de archivos) y varios complementos de almacenamiento de terceros (como el controlador CSI de Amazon FSx para NetApp ONTAP).

Para obtener una lista más completa de los complementos de Amazon EKS disponibles, consulte Complementos de Amazon EKS.

Cargas de trabajo

Kubernetes define una carga de trabajo como “una aplicación que se ejecuta en Kubernetes”. Esa aplicación puede consistir en un conjunto de microservicios que se ejecutan como contenedores en pods, o puede ejecutarse como un trabajo por lotes u otro tipo de aplicaciones. El trabajo de Kubernetes consiste en asegurarse de que las solicitudes que haga para configurar o implementar esos objetos se lleven a cabo. Si se dedica a implementar aplicaciones, debería aprender cómo se crean los contenedores, cómo se definen los pods y qué métodos puede usar para implementarlos.

Contenedores

El elemento más básico de la carga de trabajo de una aplicación que se implementa y administra Kubernetes es un pod. Un pod representa una forma de almacenar los componentes de una aplicación, así como de definir las especificaciones que describen los atributos del pod. Compárelo con algo parecido a un paquete RPM o Deb, que agrupa software para un sistema Linux, pero no se ejecuta en sí mismo como una entidad.

Como el pod es la unidad implementable más pequeña, normalmente contiene un contenedor. Sin embargo, en el caso de que los contenedores estén bien acoplados, puede haber varios contenedores en un mismo pod. Por ejemplo, un contenedor de servidor web puede estar empaquetado en un pod con un contenedor tipo sidecar que puede proporcionar servicios de registro, supervisión u otro servicio que esté estrechamente relacionado con el contenedor del servidor web. En este caso, al estar en el mismo pod, se garantiza que, para cada instancia del pod en ejecución, ambos contenedores se ejecuten siempre en el mismo nodo. Del mismo modo, todos los contenedores de un pod comparten el mismo entorno, y los contenedores de un pod se ejecutan

como si estuvieran en el mismo host aislado. El efecto de esto es que los contenedores comparten una única dirección IP que proporciona acceso al pod y los contenedores pueden comunicarse entre sí como si se ejecutaran en su propio host local.

Las especificaciones del pod (PodSpec) definen el estado deseado del pod. Puede implementar un pod individual o varios pods a través de los recursos de carga de trabajo para administrar las plantillas de pod. Los recursos de carga de trabajo incluyen implementaciones (para administrar múltiples réplicas de pods), StatefulSets (para implementar pods que deben ser únicos, como los pods de bases de datos) y DaemonSets (donde un pod debe ejecutarse de forma continua en todos los nodos). Puede obtener más información sobre estos temas más adelante.

Mientras que un pod es la unidad más pequeña que se implemente, un contenedor es la unidad más pequeña que se crea y administra.

Creación de contenedores

En realidad, el pod no es más que una estructura alrededor de uno o más contenedores, en la que cada contenedor contiene el sistema de archivos, los ejecutables, los archivos de configuración, las bibliotecas y otros componentes necesarios para ejecutar realmente la aplicación. Debido a que una empresa llamada Docker Inc. fue la primera en popularizar los contenedores, algunas personas se refieren a los contenedores como contenedores Docker. Sin embargo, desde entonces, la Open Container Initiative ha definido los tiempos de ejecución, las imágenes y los métodos de distribución de los contenedores para la industria. Si a esto le sumamos el hecho de que los contenedores se crearon a partir de muchas características de Linux existentes, otros suelen denominar contenedores OCI, contenedores Linux o simplemente contenedores.

Cuando crea un contenedor, normalmente comienza con un archivo Docker (literalmente llamado así). Dentro de ese Dockerfile, puede identificar lo siguiente:

- Una imagen base: una imagen de contenedor base es un contenedor que normalmente se crea a partir de una versión mínima del sistema de archivos de un sistema operativo (como Red Hat Enterprise Linux o Ubuntu) o de un sistema mínimo que se mejora para proporcionar software que ejecute tipos específicos de aplicaciones (como aplicaciones nodejs o python).
- Software de aplicaciones: puede agregar el software de su aplicación a su contenedor de la misma manera que lo agregaría a un sistema Linux. Por ejemplo, en su Dockerfile puede ejecutar `npm` y `yarn` para instalar una aplicación Java o `yum` y `dnf` para instalar paquetes RPM. En otras palabras, si utiliza el comando RUN en un Dockerfile, puede ejecutar cualquier comando que esté disponible en el sistema de archivos de la imagen base para instalar software o configurar software dentro de la imagen contenedora resultante.

- Instrucciones: la referencia del Dockerfile describe las instrucciones que puede agregar a un Dockerfile al configurarlo. Estas incluyen instrucciones que se utilizan para crear lo que hay en el propio contenedor (archivos `ADD` o `COPY` del sistema local), identificar los comandos que se van a ejecutar cuando se ejecuta el contenedor (`CMD` o `ENTRYPOINT`) y conectar el contenedor al sistema en el que se ejecuta (identificando el `USER` que se va a ejecutar, el `VOLUME` local que se va a montar o los puertos a `EXPOSE`).

Si bien el comando `docker` y el servicio se han utilizado tradicionalmente para crear contenedores (`docker build`), otras herramientas disponibles para crear imágenes de contenedores incluyen podman y nerdctl. Consulte Building Better Container Images o Build with Docker para obtener más información sobre la creación de contenedores.

Almacenamiento de contenedores

Una vez que haya creado la imagen del contenedor, puede almacenarla en un registro de distribución de contenedores de su estación de trabajo o en un registro de contenedores público. La ejecución de un registro de contenedores privado en su estación de trabajo le permite almacenar las imágenes de los contenedores de forma local y ponerlas a su disposición de forma inmediata.

Para almacenar las imágenes de los contenedores de una forma más pública, puede enviarlas a un registro de contenedores público. Los registros públicos de contenedores proporcionan una ubicación central para almacenar y distribuir las imágenes de los contenedores. Algunos ejemplos de registros de contenedores públicos son Amazon Elastic Container Registry, el registro Red Hat Quay y el registro Docker Hub.

Al ejecutar cargas de trabajo en contenedores en Amazon Elastic Kubernetes Service (Amazon EKS), le recomendamos que extraiga copias de las imágenes oficiales de Docker que se almacenan en Amazon Elastic Container Registry. AWS Amazon ECR ha estado almacenando estas imágenes desde 2021. Puede buscar imágenes de contenedores populares en la Galería pública de Amazon ECR y, específicamente, para las imágenes del Docker Hub, puede buscar en la Galería de Docker de Amazon ECR.

Ejecución de contenedores

Como los contenedores se crean en un formato estándar, un contenedor puede ejecutarse en cualquier máquina que pueda ejecutar un contenedor en tiempo de ejecución (por ejemplo, Docker) y cuyo contenido coincida con la arquitectura de la máquina local (por ejemplo, `x86_64` o `arm`). Para probar un contenedor o simplemente ejecutarlo en el escritorio local, puede usar los comandos

`docker run` o `podman run` para iniciar un contenedor en el servidor local. Sin embargo, en Kubernetes, cada nodo de trabajo tiene implementado un tiempo de ejecución del contenedor y depende de Kubernetes que se solicite que un nodo ejecute un contenedor.

Una vez que se ha asignado un contenedor para que se ejecute en un nodo, este comprueba si la versión solicitada de la imagen del contenedor ya existe en el nodo. Si no es así, Kubernetes indica al tiempo de ejecución del contenedor que extraiga ese contenedor del registro de contenedores correspondiente y, a continuación, lo ejecute localmente. Tenga en cuenta que la imagen de un contenedor hace referencia al paquete de software que se mueve entre su portátil, el registro del contenedor y los nodos de Kubernetes. Un contenedor hace referencia a una instancia en ejecución de esa imagen.

Pods

Una vez que los contenedores estén listos, trabajar con los pods incluye configurar, implementar y hacer que estos sean accesibles.

Configuración de pods

Cuando define un pod, le asigna un conjunto de atributos. Esos atributos deben incluir al menos el nombre del pod y la imagen del contenedor para que se ejecuten. Sin embargo, hay muchas otras cosas que también querrá configurar con las definiciones de su pod (consulte la página PodSpec para obtener más información sobre lo que puede incluir un pod). Entre ellos se incluyen:

- Almacenamiento: cuando un contenedor en ejecución se detiene y se elimina, el almacenamiento de datos en ese contenedor desaparecerá, a menos que configure un almacenamiento más permanente. Kubernetes admite muchos tipos de almacenamiento diferentes y los abstrae bajo el nombre de Volúmenes. Los tipos de almacenamiento incluyen CephFS, NFS, iSCSI, etc. Incluso puede usar un dispositivo de bloques local desde la computadora local. Con uno de esos tipos de almacenamiento disponibles en su clúster, puede montar el volumen de almacenamiento en un punto de montaje seleccionado del sistema de archivos del contenedor. Un volumen persistente es aquel que sigue existiendo después de eliminar el pod, mientras que un volumen efímero se elimina cuando se elimina el pod. Si el administrador del clúster creó diferentes StorageClasses para el clúster, es posible que tenga la opción de elegir los atributos del almacenamiento que va a utilizar, por ejemplo, si el volumen se elimina o se recupera después de su uso, si se ampliará si se necesita más espacio e incluso si cumple con ciertos requisitos de rendimiento.
- Secretos: al poner los secretos a disposición de los contenedores en las especificaciones del pod, puede proporcionar los permisos que esos contenedores necesitan para acceder a los sistemas

de archivos, las bases de datos u otros activos protegidos. Las claves, las contraseñas y los tokens son algunos de los elementos que se pueden almacenar como secretos. El uso de secretos permite no tener que almacenar esta información en imágenes de contenedores, sino que solo es necesario que los secretos estén disponibles para los contenedores en ejecución. Los ConfigMaps son similares a los secretos. Un ConfigMap tiende a contener información menos crítica, como pares clave-valor para configurar un servicio.

- Recursos de contenedores: los objetos para seguir configurando los contenedores pueden adoptar la forma de configuración de recursos. Para cada contenedor, puede solicitar la cantidad de memoria y CPU que puede usar, así como establecer límites a la cantidad total de esos recursos que puede usar el contenedor. Consulte Resource Management for Pods and Containers para ver ejemplos.
- Interrupciones: los pods se pueden interrumpir de forma involuntaria (un nodo deja de funcionar) o de forma voluntaria (se desea una mejora). Al configurar un presupuesto de interrupciones para los pods, puede controlar en cierta medida la disponibilidad de su aplicación en caso de que se produzcan interrupciones. Para ver ejemplos, consulte Specifying a Disruption Budget for your application.
- Espacios de nombres: Kubernetes proporcionan diferentes formas de aislar los componentes de Kubernetes y las cargas de trabajo entre sí. Ejecutar todos los pods de una aplicación concreta en el mismo espacio de nombres es una forma habitual de proteger y administrar esos pods juntos. Puede crear sus propios espacios de nombres para usarlos o elegir no indicar ningúno (lo que hace que Kubernetes utilice el espacio de nombres `default`). Los componentes del plano de control de Kubernetes normalmente se ejecutan en el espacio de nombres de kube-system.

La configuración que acabamos de describir normalmente se recopila en un archivo YAML para aplicarla al clúster de Kubernetes. En el caso de los clústeres de Kubernetes personales, puede almacenar estos archivos YAML en el sistema local. Sin embargo, dado que los clústeres y las cargas de trabajo son más importantes, GitOps es una forma popular de automatizar el almacenamiento y las actualizaciones de los recursos de carga de trabajo e infraestructura de Kubernetes.

Los objetos que se utilizan para recopilar e implementar la información del pod se definen mediante uno de los siguientes métodos de implementación.

Implementación de pods

El método que elija para implementar los pods depende del tipo de aplicación que planee ejecutar con ellos. Aquí tiene algunas opciones:

- Aplicaciones sin estado: una aplicación sin estado no guarda los datos de la sesión de un cliente, por lo que no es necesario volver a hacer referencia a lo que ocurrió en una sesión anterior. Esto hace que sea más fácil sustituir los pods por otros nuevos en caso de que no funcionen correctamente, o bien moverlos de un sitio a otro sin guardar su estado. Si ejecuta una aplicación sin estado (como un servidor web), puede usar una Implementación para implementar Pods y Replicasets. Un ReplicaSet define cuántas instancias de un pod desea que se ejecuten simultáneamente. Aunque puede ejecutar un ReplicaSet directamente, es habitual ejecutar réplicas directamente dentro de una implementación, para definir cuántas réplicas de un pod deben ejecutarse a la vez.
- Aplicaciones con estado: una aplicación con estado es aquella en la que la identidad del pod y el orden en que se lanzan los pods son importantes. Estas aplicaciones necesitan un almacenamiento persistente que sea estable y deben implementarse y escalarse de manera coherente. Para implementar una aplicación con estado Kubernetes, puede usar StatefulSets. Un ejemplo de una aplicación que normalmente se ejecuta como StatefulSet es una base de datos. Dentro de un StatefulSet, puede definir las réplicas, el pod y sus contenedores, los volúmenes de almacenamiento que se van a montar y las ubicaciones del contenedor donde se almacenan los datos. Consulte Run a Replicated Stateful Application para ver un ejemplo de una base de datos que se implementa como ReplicaSet.
- Aplicaciones por nodo: hay ocasiones en las que desea ejecutar una aplicación en cada nodo del clúster de Kubernetes. Por ejemplo, su centro de datos puede requerir que todos los equipos ejecuten una aplicación de monitoreo o un servicio de acceso remoto concreto. Para Kubernetes, puede utilizar un DaemonSet para garantizar que la aplicación seleccionada se ejecute en todos los nodos del clúster.
- Las aplicaciones se ejecutan hasta completarse: hay algunas aplicaciones que desea ejecutar para completar una tarea concreta. Esto podría incluir uno que publique informes de estado mensuales o que elimine datos antiguos. Se puede usar un objeto Job para configurar una aplicación para que se inicie y ejecute y, a continuación, se cierre cuando finalice la tarea. Un objeto CronJob permite configurar una aplicación para que se ejecute a una hora, minuto, día del mes, mes o día de la semana específicos, mediante una estructura definida por el formato crontab de Linux.

Hacer que las aplicaciones sean accesibles desde la red

Dado que las aplicaciones solían implementarse como un conjunto de microservicios que se desplazaban a diferentes lugares, Kubernetes necesitaba una forma de que esos microservicios pudieran encontrarse entre sí. Además, para que otras personas pudieran acceder a una aplicación fuera del clúster de Kubernetes, Kubernetes necesitaba una forma de exponer esa aplicación en

direcciones y puertos externos. Estas características relacionadas con las redes se llevan a cabo con los objetos Servicio y Entrada, respectivamente:

- Servicios: dado que un pod puede moverse a diferentes nodos y direcciones, otro pod que necesite comunicarse con el primer pod podría tener dificultades para encontrar dónde está. Para resolver este problema, Kubernetes le permite representar una aplicación como un servicio. Con un servicio, puede identificar un pod o un conjunto de pods con un nombre concreto y, a continuación, indicar qué puerto expone el servicio de esa aplicación desde el pod y qué puertos podría utilizar otra aplicación para contactar con ese servicio. Otro pod de un clúster puede simplemente solicitar un servicio por su nombre y Kubernetes dirigirá esa solicitud al puerto adecuado de una instancia del pod que ejecute ese servicio.
- Entrada: entrada es lo que permite que las aplicaciones representadas por los servicios de Kubernetes estén disponibles para los clientes que se encuentran fuera del clúster. Las características básicas de entrada incluyen un equilibrador de carga (administrado por la entrada), el controlador de entrada y reglas para dirigir las solicitudes desde el controlador al servicio. Hay varios controladores de entrada que puede elegir con Kubernetes.

Siguientes pasos

Comprender los conceptos básicos de Kubernetes y su relación con Amazon EKS le ayudará a navegar por la documentación de Amazon EKS y la documentación de Kubernetes para encontrar la información que necesita para administrar los clústeres de Amazon EKS e implementar cargas de trabajo en esos clústeres. Para comenzar a usar Amazon EKS, elija una de las siguientes opciones:

- Crear un clúster sencillo
- Crear un clúster más complejo
- Implementar una aplicación de ejemplo
- Explorar las formas de administrar el clúster

Opciones de implementación

Puede implementar Amazon EKS con cualquiera de las opciones siguientes:

Amazon EKS en la nube

> Puede ejecutar Kubernetes en la nube de AWS sin necesidad de instalar, operar ni mantener sus propios nodos o plano de control de Kubernetes. Esta opción es la que se describe en esta guía.

Amazon EKS en Outposts

AWS Outposts habilita los Servicios de AWS, la infraestructura y los modelos operativos nativos en las instalaciones. Con Amazon EKS en Outposts, puede elegir ejecutar clústeres extendidos o locales. Con clústeres extendidos, el plano de control de Kubernetes se ejecuta en una Región de AWS y los nodos se ejecutan en Outposts. Con los clústeres locales, todo el clúster de Kubernetes se ejecuta localmente en Outposts, incluidos el plano de control de Kubernetes y los nodos. Para obtener más información, consulte Amazon EKS en AWS Outposts.

Amazon EKS Anywhere

Amazon EKS Anywhere es una opción de implementación para Amazon EKS que le permite crear y operar fácilmente clústeres de Kubernetes en las instalaciones. Tanto Amazon EKS como Amazon EKS Anywhere se basan en Amazon EKS Distro. Para obtener más información sobre Amazon EKS Anywhere y sus diferencias con Amazon EKS, consulte Información general y Comparación de Amazon EKS Anywhere con Amazon EKS en la documentación de Amazon EKS Anywhere. Para obtener respuestas a algunas preguntas frecuentes, consulte Preguntas frecuentes sobre Amazon EKS Anywhere.

Amazon EKS Distro

Amazon EKS Distro es una distribución del mismo software y dependencias de código abierto de Kubernetes implementados por Amazon EKS en la nube. Amazon EKS Distro sigue el mismo ciclo de lanzamiento de la versión de Kubernetes que Amazon EKS y se proporciona como un proyecto de código abierto. Para obtener más información, consulte Amazon EKS Distro. También puede ver y descargar el código fuente de Amazon EKS Distro en GitHub.

Al elegir qué opciones de implementación usar para el clúster de Kubernetes, tenga en cuenta lo siguiente:

Funcionalidad	Amazon EKS	Amazon EKS en Outposts	Amazon EKS Anywhere	Amazon EKS Distro
Hardware	Suministrado por AWS	Suministrado por AWS	Suministrado por usted	Suministrado por usted
Ubicación de la implementación	Nube de AWS	Su centro de datos	Su centro de datos	Su centro de datos

Funcionalidad	Amazon EKS	Amazon EKS en Outposts	Amazon EKS Anywhere	Amazon EKS Distro
Ubicación del plano de control de Kubernetes	Nube de AWS	Nube de AWS o su centro de datos	Su centro de datos	Su centro de datos
Ubicación del plano de datos de Kubernetes	Nube de AWS	Su centro de datos	Su centro de datos	Su centro de datos
Soporte	AWS Support	AWS Support	AWS Support	Apoyo de la comunidad de OSS

Configuración para usar Amazon EKS

Los recursos de AWS suelen tener restricciones de acceso que limitan el acceso a la entidad de AWS que los creó. Por lo tanto, es fundamental establecer una configuración de usuario adecuada en la AWS Command Line Interface desde el principio. Además, debe equipar su máquina local con las herramientas esenciales para una administración eficiente de la línea de comandos de su clúster de Amazon EKS. Este tema le será de ayuda para empezar a administrar su clúster desde la línea de comandos.

Paso 1: configurar la AWS CLI

La AWS CLI es una herramienta de línea de comandos para trabajar con los servicios de AWS, incluido Amazon EKS. También se usa para autenticar a los roles o usuarios de IAM para acceder al clúster de Amazon EKS y a otros recursos de AWS desde su máquina local. Para aprovisionar recursos en AWS desde la línea de comandos, debe obtener un ID de clave de acceso de AWS y una clave secreta para utilizarlos en la línea de comandos. A continuación, debe configurar estas credenciales en la AWS CLI. Si aún no ha instalado la AWS CLI, consulte Instalar o actualizar la última versión de la AWS CLI en la Guía del usuario de AWS Command Line Interface.

Para crear una clave de acceso

1. Inicie sesión en el AWS Management Console.
2. En la esquina superior derecha, elija el nombre de usuario de su cuenta de AWS para abrir el menú de navegación. En este ejemplo, elija **`webadmin`**. A continuación, elija Credenciales de seguridad.
3. En Clave de acceso, elija Crear clave de acceso.
4. Elija Interfaz de la línea de comandos (CLI) y, a continuación, haga clic en Siguiente.
5. Elija Create access key (Crear clave de acceso).
6. Elija Descargar archivo .csv.

Para configurar la AWS CLI

Después de instalar la AWS CLI, siga estos pasos para configurarla: Para obtener más información, consulte Configuración de AWS CLI en la Guía del usuario de AWS Command Line Interface.

1. En una ventana de terminal, ingrese el siguiente comando:

   ```
   aws configure
   ```

   Si lo desea, puede configurar un perfil con nombre, como **`--profile cluster-admin`**. Si configura un perfil con nombre en la AWS CLI, deberá transferir siempre este indicador en los siguientes comandos.

2. Introduzca las credenciales de AWS. Por ejemplo:

   ```
   AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
   AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   Default region name [None]: region-code
   Default output format [None]: json
   ```

Cómo obtener un token de seguridad

Si es necesario, ejecute el siguiente comando para obtener un token de seguridad nuevo para la AWS CLI. Para obtener más información, consulte `get-session-token` en la Referencia de los comandos de AWS CLI.

De forma predeterminada, el token será válido durante 15 minutos. Para cambiar el tiempo de espera predeterminado de la sesión, transfiera el indicador **`--duration-seconds`**. Por ejemplo:

```
aws sts get-session-token --duration-seconds 3600
```

Este comando devuelve las credenciales de seguridad temporales de una sesión de la AWS CLI. Debería ver la siguiente respuesta de salida:

```
{
    "Credentials": {
        "AccessKeyId": "ASIA5FTRU3LOEXAMPLE",
        "SecretAccessKey": "JnKgvwfqUD9mNsPoi9IbxAYEXAMPLE",
        "SessionToken": "VERYLONGSESSIONTOKENSTRING",
        "Expiration": "2023-02-17T03:14:24+00:00"
    }
}
```

Cómo verificar la identidad del usuario

Si es necesario, ejecute el siguiente comando para verificar las credenciales de AWS para su identidad de usuario de IAM (por ejemplo, *ClusterAdmin*) para la sesión de terminal.

```
aws sts get-caller-identity
```

Este comando devuelve el nombre de recurso de Amazon (ARN) de la entidad de IAM que está configurada para la AWS CLI. Debería ver la siguiente respuesta de ejemplo de salida:

```
{
    "UserId": "AKIAIOSFODNN7EXAMPLE",
    "Account": "01234567890",
    "Arn": "arn:aws:iam::01234567890:user/ClusterAdmin"
}
```

Paso 2: instalar las herramientas de Kubernetes

Para comunicarse con un clúster de Kubernetes, necesitará una herramienta para interactuar con la API de Kubernetes. Además, necesitará algunas otras herramientas, como una para administrar los entornos de Kubernetes en su máquina local.

Para crear los recursos de AWS

- Recursos del clúster de Amazon EKS: si es la primera vez que utiliza AWS, le recomendamos que instale `eksctl`. `eksctl` es una utilidad de infraestructura como código (IaC) que utiliza AWS CloudFormation para crear fácilmente su clúster de Amazon EKS. También crea recursos adicionales de Kubernetes, como cuentas de servicio. Para obtener instrucciones sobre cómo instalar `eksctl`, consulte Instalación en la documentación de `eksctl`.
- Recursos de AWS: si el usuario está acostumbrado a automatizar el aprovisionamiento y la implementación de la infraestructura de AWS, recomendamos instalar Terraform. Terraform es una herramienta de infraestructura como código (IaC) de código abierto desarrollada por HashiCorp. Le permite definir y aprovisionar la infraestructura mediante un lenguaje de configuración de alto nivel, como el lenguaje de configuración de HashiCorp (HCL) o JSON. Para obtener instrucciones sobre cómo instalar Terraform, consulte Instalar Terraform en la documentación de Terraform.

Para instalar **kubectl**

`kubectl` es una herramienta de línea de comandos de código abierto que se utiliza para comunicarse con el servidor de la API de Kubernetes en el clúster de Amazon EKS. Si aún no lo tiene instalado en su máquina local, elija una de las siguientes opciones.

- Versiones de AWS: para instalar una versión de `kubectl` compatible con Amazon EKS, consulte Instalación o actualización del `kubectl`.
- Versiones comunitarias: para instalar la última versión comunitaria de `kubectl`, consulte la página Instalar herramientas en la documentación de Kubernetes.

Cómo configurar un entorno de desarrollo

- Herramienta de implementación local: si es la primera vez que usa una herramienta de Kubernetes, considere la posibilidad de instalar una herramienta de implementación local como `minikube` o `kind`. Estas herramientas le permiten administrar un clúster de Amazon EKS en su máquina local.
- Administrador de paquetes: Helm es un administrador de paquetes de Kubernetes muy popular que simplifica la instalación y administración de paquetes complejos. Con Helm, es más fácil instalar y administrar paquetes como el controlador del equilibrador de carga de AWS en su clúster de Amazon EKS.

Siguientes pasos

- Introducción a Amazon EKS

Instalación o actualización del **kubectl**

`Kubectl` es una herramienta de línea de comandos que se utiliza para comunicarse con el servidor de API de Kubernetes. El dato binario `kubectl` está disponible en muchos administradores de paquetes de sistemas operativos. Utilizar un administrador de paquetes para la instalación es normalmente más sencillo que el proceso manual de descarga e instalación.

Este tema le ayuda a descargar e instalar, o actualizar, el `kubectl` binario en su dispositivo. El dato binario es idéntico a las versiones de comunidad ascendente. El dato binario no es exclusivo de Amazon EKS o de AWS.

> **Note**
>
> Debe utilizar una versión de `kubectl` con una diferencia de versión de menos de un número que el plano del control del clúster de Amazon EKS. Por ejemplo, un cliente de `kubectl` `1.29` trabaja con los clústeres Kubernetes, `1.28`, `1.29` y `1.30`.

Instalar o actualizar **`kubectl`**

1. Determine si ya tiene `kubectl` instalado en su dispositivo.

   ```
   kubectl version --client
   ```

   Si tiene `kubectl` instalado en la ruta de su dispositivo, el resultado de ejemplo incluye información similar a la siguiente. Si desea actualizar la versión que tiene instalada actualmente con una versión posterior, complete el siguiente paso y asegúrese de instalar la nueva versión en la misma ubicación en la que se encuentra la versión actual.

   ```
   Client Version: v1.30.X-eks-1234567
   ```

   Si no recibe un resultado, entonces no tiene `kubectl` instalado o no está instalado en una ubicación que esté en la ruta de acceso del dispositivo.
2. Instalar o actualizar `kubectl` en los sistemas operativos `macOS`, Linux y Windows.

   macOS

   Para instalar o actualizar **`kubectl`** en **`macOS`**

   1. Descargue el dato binario de la versión de Kubernetes de su clúster de Amazon S3.
      - Kubernetes `1.30`

        ```
        curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/darwin/amd64/kubectl
        ```

      - Kubernetes `1.29`

        ```
        curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/darwin/amd64/kubectl
        ```

- Kubernetes 1.28

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.27

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.26

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.25

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.24

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.23

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.22

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/darwin/amd64/kubectl
```

- Kubernetes 1.21

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/darwin/amd64/kubectl
```

2. (Opcional) Compruebe el dato binario descargado con la suma de comprobación de SHA-256 de su dato binario.

   a. Descargue la suma de comprobación de SHA-256 de la versión de Kubernetes de su clúster.

   - Kubernetes 1.30

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.29

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.28

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.27

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.26

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.25

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.24

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/darwin/amd64/kubectl.sha256
     ```

   - Kubernetes 1.23

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/darwin/amd64/kubectl.sha256
```

b. Verifique la suma de comprobación de `SHA-256` del dato binario descargado.

```
openssl sha1 -sha256 kubectl
```

c. Asegúrese de que la suma de comprobación generada en la salida coincida con la suma de comprobación del archivo de `kubectl.sha256` descargado.

3. Aplique permisos de ejecución al binario.

```
chmod +x ./kubectl
```

4. Copie el binario en una carpeta en `PATH`. Si ya ha instalado una versión de `kubectl`, recomendamos que cree un `$HOME/bin/kubectl` y se asegure de que `$HOME/bin` venga en primer lugar en su `$PATH`.

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

5. (Opcional) Agregue la ruta `$HOME/bin` a su archivo de inicialización del shell para que se configure cuando abra un shell.

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bash_profile
```

Linux (amd64)

Para instalar o actualizar **`kubectl`** en Linux (**`amd64`**)

1. Descargue el dato binario de `kubectl` para la versión de Kubernetes del clúster de Amazon S3.
   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl
     ```

   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl
     ```

   - Kubernetes `1.29`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/linux/amd64/kubectl
     ```

   - Kubernetes `1.28`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/linux/amd64/kubectl
     ```

   - Kubernetes `1.27`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/linux/amd64/kubectl
     ```

   - Kubernetes `1.26`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/linux/amd64/kubectl
     ```

   - Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/linux/amd64/kubectl
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/amd64/kubectl
```

2. (Opcional) Compruebe el dato binario descargado con la suma de comprobación de `SHA-256` de su dato binario.

   a. Descargue la suma de comprobación de `SHA-256` para la versión de Kubernetes del clúster desde Amazon S3 mediante el comando de la plataforma de hardware de su dispositivo. El primer enlace de cada versión es para `amd64` y el segundo enlace es para `arm64`.

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl.sha256
```

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/amd64/kubectl.sha256
```

   - Kubernetes `1.29`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.28`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.27`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.26`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/
bin/linux/amd64/kubectl.sha256
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/amd64/kubectl.sha256
```

b. Controle la suma de comprobación de SHA-256 del archivo binario descargado con uno de los siguientes comandos.

- ```
  sha256sum -c kubectl.sha256
  ```

  Al usar este comando, asegúrese de ver el resultado siguiente:

  ```
  kubectl: OK
  ```

- ```
  openssl sha1 -sha256 kubectl
  ```

  Cuando use este comando, asegúrese de que la suma de comprobación generada en la salida coincida con la suma de comprobación del archivo de `kubectl.sha256` descargado.

3. Aplique permisos de ejecución al binario.

```
chmod +x ./kubectl
```

4. Copie el binario en una carpeta en PATH. Si ya ha instalado una versión de `kubectl`, recomendamos que cree un `$HOME/bin/kubectl` y se asegure de que `$HOME/bin` venga en primer lugar en su $PATH.

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

5. (Opcional) Agregue la ruta `$HOME/bin` a su archivo de inicialización del shell para que se configure cuando abra un shell.

> **Note**
>
> En este paso, se presupone que usa el shell Bash; si utiliza otro shell, cambie el comando para utilizar su archivo de inicialización del shell específico.

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
```

Linux (arm64)

Para instalar o actualizar **`kubectl`** en Linux (**`arm64`**)

1. Descargue el dato binario de `kubectl` para la versión de Kubernetes del clúster de Amazon S3.
   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.30`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.29`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.28`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.27`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/linux/arm64/kubectl
     ```

   - Kubernetes `1.26`

     ```
     curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/linux/arm64/kubectl
     ```

- Kubernetes `1.25`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.24`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.23`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.22`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/bin/linux/arm64/kubectl
```

- Kubernetes `1.21`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/arm64/kubectl
```

2. (Opcional) Compruebe el dato binario descargado con la suma de comprobación de `SHA-256` de su dato binario.

   a. Descargue la suma de comprobación de `SHA-256` para la versión de Kubernetes del clúster desde Amazon S3 mediante el comando de la plataforma de hardware de su dispositivo. El primer enlace de cada versión es para `amd64` y el segundo enlace es para `arm64`.

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/arm64/kubectl.sha256
```

   - Kubernetes `1.30`

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.29

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.28

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.27

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.26

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.25

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.16/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.24

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.17/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.23

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.17/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.22

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.17/2024-04-19/
bin/linux/arm64/kubectl.sha256
```

- Kubernetes 1.21

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/arm64/kubectl.sha256
```

b. Controle la suma de comprobación de SHA-256 del archivo binario descargado con uno de los siguientes comandos.

- ```
  sha256sum -c kubectl.sha256
  ```

  Al usar este comando, asegúrese de ver el resultado siguiente:

  ```
  kubectl: OK
  ```

- ```
  openssl sha1 -sha256 kubectl
  ```

  Cuando use este comando, asegúrese de que la suma de comprobación generada en la salida coincida con la suma de comprobación del archivo de `kubectl.sha256` descargado.

3. Aplique permisos de ejecución al binario.

```
chmod +x ./kubectl
```

4. Copie el binario en una carpeta en PATH. Si ya ha instalado una versión de `kubectl`, recomendamos que cree un `$HOME/bin/kubectl` y se asegure de que `$HOME/bin` venga en primer lugar en su `$PATH`.

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

5. (Opcional) Agregue la ruta `$HOME/bin` a su archivo de inicialización del shell para que se configure cuando abra un shell.

> **Note**
>
> En este paso, se presupone que usa el shell Bash; si utiliza otro shell, cambie el comando para utilizar su archivo de inicialización del shell específico.

```
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
```

Windows

Para instalar o actualizar **kubectl** en Windows

1. Abra un terminal PowerShell.
2. Descargue el dato binario de `kubectl` para la versión de Kubernetes del clúster de Amazon S3.
   - Kubernetes `1.30`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.30.0/2024-05-12/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.29`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.3/2024-04-19/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.28`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.8/2024-04-19/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.27`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.12/2024-04-19/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.26`

     ```
     curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.15/2024-04-19/bin/windows/amd64/kubectl.exe
     ```

   - Kubernetes `1.25`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.25.16/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.24`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.24.17/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.23`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.23.17/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.22`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.22.17/2024-04-19/bin/windows/amd64/kubectl.exe
```

- Kubernetes `1.21`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.21.14/2024-04-19/bin/windows/amd64/kubectl.exe
```

3. (Opcional) Compruebe el dato binario descargado con la suma de comprobación de `SHA-256` de su dato binario.

   a. Descargue la suma de comprobación de `SHA-256` de la versión de Kubernetes del clúster para Windows.

   - Kubernetes `1.30`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.30.0/2024-05-12/bin/windows/amd64/kubectl.exe.sha256
```

   - Kubernetes `1.29`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.29.3/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

   - Kubernetes `1.28`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.28.8/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.27`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.27.12/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.26`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.26.15/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.25`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.25.16/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.24`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.24.17/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.23`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.23.17/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.22`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.22.17/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

- Kubernetes `1.21`

```
curl.exe -O https://s3.us-west-2.amazonaws.com/amazon-
eks/1.21.14/2024-04-19/bin/windows/amd64/kubectl.exe.sha256
```

b. Verifique la suma de comprobación de `SHA-256` del dato binario descargado.

```
Get-FileHash kubectl.exe
```

c. Asegúrese de que la suma de comprobación generada en la salida coincida con la suma de comprobación del archivo de `kubectl.sha256` descargado. La salida de PowerShell debe ser una cadena de caracteres equivalente en mayúsculas.

4. Copie el binario en una carpeta en PATH. Si tiene un directorio existente en su PATH que utiliza para utilidades de línea de comandos, copie el binario en ese directorio. De lo contrario, lleve a cabo los pasos que figuran a continuación.
   a. Cree un nuevo directorio para los binarios de línea de comandos, como `C:\bin`.
   b. Copie el binario `kubectl.exe` en el nuevo directorio.
   c. Edite la variable de entorno PATH del usuario o sistema para agregar el nuevo directorio a su PATH.
   d. Cierre el terminal de PowerShell y abra uno nuevo para obtener la nueva variable PATH.

3. Una vez que instale `kubectl`, puede comprobar la versión.

```
kubectl version --client
```

Al instalar por primera vez `kubectl`, aún no está configurado para comunicarse con ningún servidor. Trataremos esta configuración según sea necesario en otros procedimientos. Si alguna vez necesita actualizar la configuración para comunicarse con un clúster en particular, puede ejecutar el siguiente comando. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster. Reemplace *`my-cluster`* por el nombre del clúster.

```
aws eks update-kubeconfig --region region-code --name my-cluster
```

Introducción a Amazon EKS

Asegúrese de tener configurado el uso de Amazon EKS antes de seguir las guías de introducción. Para obtener más información, consulte Configuración para usar Amazon EKS.

Existen dos guías de introducción disponibles para crear un clúster de Kubernetes nuevo con nodos en Amazon EKS:

- Introducción a Amazon EKS: `eksctl`: esta guía de introducción lo ayuda a instalar todos los recursos necesarios a fin de comenzar a utilizar Amazon EKS mediante `eksctl`, una utilidad de línea de comandos sencilla para crear y administrar clústeres de Kubernetes en Amazon EKS. Al final del tutorial, contará con un clúster de Amazon EKS en ejecución en el que puede implementar aplicaciones. Esta es la forma más sencilla y rápida de comenzar a utilizar Amazon EKS.
- Introducción a Amazon EKS: AWS Management Console y AWS CLI: esta guía de introducción lo ayuda a crear todos los recursos necesarios para comenzar a utilizar Amazon EKS con la AWS Management Console y la AWS CLI. Al final del tutorial, contará con un clúster de Amazon EKS en ejecución en el que puede implementar aplicaciones. En esta guía, creará cada recurso necesario para un clúster de Amazon EKS de forma manual. Los procedimientos proporcionan una visibilidad de cómo se crea cada recurso y cómo interactúan entre sí.

También ofrecemos las siguientes referencias:

- Para ver una colección seleccionada de tutoriales prácticos, consulte Navegar por Amazon EKS en la comunidad de AWS.
- Para ver ejemplos de código, consulte Ejemplos de código para Amazon EKS con las SDK deAWS.

Introducción a Amazon EKS: **`eksctl`**

Esta guía lo ayuda a crear todos los recursos necesarios a fin de comenzar a utilizar Amazon Elastic Kubernetes Service (Amazon EKS) mediante `eksctl`, una utilidad de línea de comandos sencilla para crear y administrar clústeres de Kubernetes en Amazon EKS. Al final de este tutorial, contará con un clúster de Amazon EKS en ejecución en el que puede implementar aplicaciones.

Los procedimientos de esta guía crean varios recursos de forma automática que debe establecer manualmente al crear el clúster mediante la AWS Management Console. Si prefiere crear la mayoría

de los recursos de forma manual y comprender mejor cómo interactúan entre sí, utilice la AWS Management Console para crear el clúster y la informática. Para obtener más información, consulte Introducción a Amazon EKS: AWS Management Console y AWS CLI.

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesitará para crear y administrar un clúster de Amazon EKS.

- **`kubectl`**: una herramienta de línea de comandos para trabajar con clústeres de Kubernetes. Para obtener más información, consulte Instalación o actualización del `kubectl`.
- **`eksctl`**: una herramienta de línea de comandos para trabajar con clústeres de EKS que automatiza varias tareas individuales. Para obtener más información, consulte la sección de Instalación en la documentación de `eksctl`.
- Permisos de IAM necesarios: la entidad principal de seguridad de IAM que está utilizando debe contar con permisos para trabajar con los roles de IAM de Amazon EKS y los roles vinculados al servicio, AWS CloudFormation, y una VPC y recursos relacionados. Para obtener más información, consulte Acciones, recursos y claves de condición de Amazon Elastic Container Service for Kubernetes y Uso de roles vinculados a servicios en la guía del usuario de IAM. Debe completar todos los pasos de esta guía como el mismo usuario. Ejecute el siguiente comando para comprobar el usuario actual:

```
aws sts get-caller-identity
```

Paso 1: crear el clúster y los nodos de Amazon EKS

> **Important**
>
> Para comenzar de la manera más sencilla y rápida posible, este tema incluye pasos a fin de crear un clúster y nodos con la configuración predeterminada. Antes de crear un clúster y nodos para su uso en producción, recomendamos que conozca toda la configuración e implemente un clúster y nodos con la configuración que satisfaga sus requisitos. Para obtener más información, consulte Creación de un clúster de Amazon EKS y Nodos de Amazon EKS. Algunos ajustes de configuración solo se pueden habilitar al crear el clúster y los nodos.

Puede crear un clúster con uno de los siguientes tipos de nodos. Para obtener más información sobre cada tipo, consulte Nodos de Amazon EKS. Después de implementar el clúster, puede agregar otros tipos de nodos.

- Fargate – Linux: seleccione este tipo de nodo si desea ejecutar aplicaciones de Linux en AWS Fargate. Fargate es un motor de computación sin servidor que le permite implementar Pods de Kubernetes sin administrar instancias de Amazon EC2.
- Nodos administrados – Linux: seleccione este tipo de nodo si desea ejecutar aplicaciones de Amazon Linux en instancias de Amazon EC2. Aunque no se trata en esta guía, también puede agregar nodos autoadministrados de Windows y nodos Bottlerocket a su clúster.

Cree su clúster de Amazon EKS con el siguiente comando. Puede reemplazar *my-cluster* por un valor propio. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. Reemplace *region-code* por cualquier Región de AWS en la que se admita Amazon EKS. Para ver una lista de Regiones de AWS, consulte Puntos de conexión y cuotas de Amazon EKS en la Guía de referencia general de AWS.

Fargate – Linux

```
eksctl create cluster --name my-cluster --region region-code --fargate
```

Managed nodes – Linux

```
eksctl create cluster --name my-cluster --region region-code
```

La creación del clúster tarda varios minutos. Durante la creación, verá varias líneas de salida. La última línea de salida es similar a la siguiente línea de ejemplo.

```
[...]
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

`eksctl` creó un archivo de `config` de `kubectl` en `~/.kube` o agregó la configuración del clúster nuevo dentro de un archivo existente de `config` en `~/.kube` en su computadora.

Después de que se complete la creación del clúster, consulte la pila de AWS CloudFormation denominada `eksctl-my-cluster-cluster` en la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation para ver todos los recursos que se crearon.

Paso 2: vea recursos de Kubernetes

1. Vea los nodos del clúster.

   ```
   kubectl get nodes -o wide
   ```

   Un ejemplo de salida sería el siguiente.

   Fargate – Linux

   ```
   NAME                                                STATUS   ROLES    AGE
      VERSION              INTERNAL-IP   EXTERNAL-IP   OS-IMAGE         KERNEL-
   VERSION                   CONTAINER-RUNTIME
   fargate-ip-192-0-2-0.region-code.compute.internal   Ready    <none>
     8m3s    v1.2.3-eks-1234567   192.0.2.0     <none>        Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   fargate-ip-192-0-2-1.region-code.compute.internal   Ready    <none>
     7m30s   v1.2.3-eks-1234567   192-0-2-1     <none>        Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   ```

   Managed nodes – Linux

   ```
   NAME                                        STATUS   ROLES    AGE    VERSION
             INTERNAL-IP   EXTERNAL-IP   OS-IMAGE         KERNEL-VERSION
          CONTAINER-RUNTIME
   ip-192-0-2-0.region-code.compute.internal   Ready    <none>   6m7s
      v1.2.3-eks-1234567   192.0.2.0     192.0.2.2     Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   ip-192-0-2-1.region-code.compute.internal   Ready    <none>   6m4s
      v1.2.3-eks-1234567   192.0.2.1     192.0.2.3     Amazon Linux 2
     1.23.456-789.012.amzn2.x86_64   containerd://1.2.3
   ```

   Para obtener más información acerca de lo que ve en la salida, consulte Vea los recursos de Kubernetes.

2. Vea las cargas de trabajo que se ejecutan en el clúster.

```
kubectl get pods -A -o wide
```

Un ejemplo de salida sería el siguiente.

Fargate – Linux

```
NAMESPACE     NAME                          READY   STATUS    RESTARTS   AGE   IP
          NODE                                                NOMINATED NODE
 READINESS GATES
kube-system   coredns-1234567890-abcde   1/1     Running   0          18m
  192.0.2.0   fargate-ip-192-0-2-0.region-code.compute.internal   <none>
   <none>
kube-system   coredns-1234567890-12345   1/1     Running   0          18m
  192.0.2.1   fargate-ip-192-0-2-1.region-code.compute.internal   <none>
   <none>
```

Managed nodes – Linux

```
NAMESPACE     NAME                          READY   STATUS    RESTARTS   AGE     IP
          NODE                                        NOMINATED NODE   READINESS
 GATES
kube-system   aws-node-12345             1/1     Running   0          7m43s
   192.0.2.1   ip-192-0-2-1.region-code.compute.internal   <none>
 <none>
kube-system   aws-node-67890             1/1     Running   0          7m46s
   192.0.2.0   ip-192-0-2-0.region-code.compute.internal   <none>
 <none>
kube-system   coredns-1234567890-abcde   1/1     Running   0          14m
   192.0.2.3   ip-192-0-2-3.region-code.compute.internal   <none>
 <none>
kube-system   coredns-1234567890-12345   1/1     Running   0          14m
   192.0.2.4   ip-192-0-2-4.region-code.compute.internal   <none>
 <none>
kube-system   kube-proxy-12345           1/1     Running   0          7m46s
   192.0.2.0   ip-192-0-2-0.region-code.compute.internal   <none>
 <none>
kube-system   kube-proxy-67890           1/1     Running   0          7m43s
   192.0.2.1   ip-192-0-2-1.region-code.compute.internal   <none>
 <none>
```

Para obtener más información acerca de lo que ve en la salida, consulte Vea los recursos de Kubernetes.

Paso 3: eliminar sus clústeres y nodos

Cuando haya terminado con el clúster y los nodos que creó para este tutorial, deberá eliminarlos con el siguiente comando. Si quiere hacer más con este clúster antes de eliminarlo, consulte Siguientes pasos.

```
eksctl delete cluster --name my-cluster --region region-code
```

Siguientes pasos

Los siguientes temas de documentación lo ayudarán a ampliar la funcionalidad de su clúster.

- Implemente una aplicación de muestra en su clúster.
- La entidad principal de IAM que creó el clúster es la única entidad principal que puede realizar llamadas al servidor de la API de Kubernetes con `kubectl` o laAWS Management Console. Si desea que otras entidades principales de IAM tengan acceso al clúster, debe agregarlas. Para obtener más información, consulte Concesión de acceso a las API de Kubernetes y Permisos necesarios.
- Antes de implementar un clúster para su uso en producción, le recomendamos que se familiarice con toda la configuración de clústeres y nodos. Algunos ajustes (como habilitar el acceso SSH a los nodos de Amazon EC2) deben establecerse cuando se crea el clúster.
- Para aumentar la seguridad del clúster, configure el complemento de interfaz de red de contenedores de Amazon VPC a fin de utilizar roles de IAM para cuentas de servicio.

Introducción a Amazon EKS: AWS Management Console y AWS CLI

Esta guía lo ayuda a crear todos los recursos necesarios para comenzar a utilizar Amazon Elastic Kubernetes Service (Amazon EKS) mediante la AWS Management Console y la AWS CLI. En esta guía, creará cada recurso de forma manual. Al final de este tutorial, contará con un clúster de Amazon EKS en ejecución en el que puede implementar aplicaciones.

Los procedimientos de esta guía le dan una visibilidad completa sobre cómo se crea cada recurso y cómo interactúan los recursos entre sí. Si prefiere que la mayoría de los recursos se creen de forma automática, utilice la CLI de `eksctl` para crear el clúster y los nodos. Para obtener más información, consulte Introducción a Amazon EKS: `eksctl`.

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesitará para crear y administrar un clúster de Amazon EKS.

- AWS CLI: una herramienta de línea de comandos para trabajar con servicios de AWS, incluido Amazon EKS. Para obtener más información, consulte Instalar, actualizar y desinstalar la AWS CLI en la guía del usuario de AWS Command Line Interface. Después de instalar la AWS CLI, recomendamos que también la configure. Para obtener más información, consulte Configuración rápida con `aws configure` en la guía del usuario de AWS Command Line Interface.
- **`kubectl`**: una herramienta de línea de comandos para trabajar con clústeres de Kubernetes. Para obtener más información, consulte Instalación o actualización del `kubectl`.
- Permisos de IAM necesarios: la entidad principal de seguridad de IAM que está utilizando debe contar con permisos para trabajar con los roles de IAM de Amazon EKS y los roles vinculados al servicio, AWS CloudFormation, una VPC y recursos relacionados. Para obtener más información, consulte Acciones, recursos y claves de condición de Amazon Elastic Kubernetes Service y Uso de roles vinculados a servicios en la Guía del usuario de IAM. Debe completar todos los pasos de esta guía como el mismo usuario. Ejecute el siguiente comando para comprobar el usuario actual:

  ```
  aws sts get-caller-identity
  ```

- Le recomendamos que siga los pasos de este tema en un intérprete de comandos Bash. Si no está utilizando un intérprete de comandos Bash, algunos comandos de script, como los caracteres de continuación de línea y la forma en que se establecen y utilizan las variables, requieren ajustes para su intérprete de comandos. Además, las reglas de entrecomillado y escape de su intérprete de comandos pueden ser diferentes. Para obtener más información, consulte Uso de entrecomillado de cadenas en la AWS CLI de la Guía del usuario de la AWS Command Line Interface.

Paso 1: crear el clúster de Amazon EKS

> **Important**
>
> Para comenzar de la manera más sencilla y rápida posible, en este tema se incluye pasos a fin de crear un clúster con la configuración predeterminada. Antes de crear un clúster para su uso en producción, recomendamos que conozca toda la configuración e implemente un clúster con la configuración que satisfaga sus requisitos. Para obtener más información, consulte Creación de un clúster de Amazon EKS. Algunos ajustes de configuración solo se pueden habilitar al crear el clúster.

Para crear el clúster

1. Cree una Amazon VPC con subredes privadas y públicas que cumplan con los requisitos de Amazon EKS. Reemplace *region-code* por cualquier Región de AWS en la que se admita Amazon EKS. Para ver una lista de Regiones de AWS, consulte Puntos de conexión y cuotas de Amazon EKS en la Guía de referencia general de AWS.Puede reemplazar *my-eks-vpc-stack* con cualquier nombre que elija.

   ```
   aws cloudformation create-stack \
     --region region-code \
     --stack-name my-eks-vpc-stack \
     --template-url https://s3.us-west-2.amazonaws.com/amazon-
   eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml
   ```

   > **Tip**
   >
   > Para obtener una lista de todos los recursos que crea el comando anterior, abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation. Elija la pila *my-eks-vpc-stack* y, a continuación, elija la pestaña Resources (Recursos).

2. Cree un rol de IAM de clúster y adjúntelo a la política administrada de IAM de Amazon EKS. Los clústeres de Kubernetes administrados por Amazon EKS realizan llamadas a otros servicios de AWS en su nombre para administrar los recursos que utiliza con el servicio.

a. Copie el siguiente contenido en un archivo denominado *eks-cluster-role-trust-policy.json*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Cree el rol.

```
aws iam create-role \
  --role-name myAmazonEKSClusterRole \
  --assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
```

c. Adjunte la política administrada de IAM por Amazon EKS requerida al rol.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
  --role-name myAmazonEKSClusterRole
```

3. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

   Asegúrese de que la Región de AWS que se muestra en la parte superior derecha de la consola sea la Región de AWS en la que desea crear el clúster. De lo contrario, elija el menú desplegable junto al nombre de la Región de AWS y elija la Región de AWS que desea utilizar.

4. Elija Add cluster (Agregar clúster) y, a continuación, elija Create (Crear). Si no ve esta opción, elija Clusters (Clústeres) en el panel de navegación izquierdo primero.

5. En la página Configure cluster (Configurar clúster), haga lo siguiente:

   a. Ingrese un Name (Nombre) para su clúster, como **`my-cluster`**. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El

nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.

b. En Cluster Service Role (Rol de servicio de clúster), elija *myAmazonEKSClusterRole*.

c. Conserve el resto de la configuración con sus valores predeterminados y elija Next (Siguiente).

6. En la página Specify networking (Especificar redes), haga lo siguiente:

a. Elija el ID de la VPC que creó en un paso anterior en la lista desplegable de VPC. Es algo similar a *vpc-00x0000x000x0x000* | *my-eks-vpc-stack-VPC*.

b. Conserve el resto de la configuración con sus valores predeterminados y elija Siguiente.

7. En la página Configurar observabilidad, elija Siguiente.

8. En la página Seleccionar complementos, elija Siguiente.

   Para obtener más información sobre los complementos, consulte Complementos de Amazon EKS.

9. En la página Configurar las opciones de complementos seleccionados, elija Siguiente.

10. En la página Review and create (Revisar y crear), elija Create (Crear).

    A la derecha del nombre del clúster, el estado del clúster es Creating (En creación) durante varios minutos hasta que se complete el proceso de aprovisionamiento del clúster. No siga con el paso siguiente hasta que el estado sea Active (Activo).

> **Note**
>
> Es posible que reciba un error que indique que una de las zonas de disponibilidad de la solicitud no tiene capacidad suficiente para crear un clúster de Amazon EKS. Si esto ocurre, el mensaje de error indicará las zonas de disponibilidad que admiten un clúster nuevo. Intente crear el clúster de nuevo con al menos dos subredes ubicadas en las zonas de disponibilidad admitidas para su cuenta. Para obtener más información, consulte Capacidad insuficiente.

Paso 2: configurar el equipo para comunicarse con el clúster

En esta sección creará un archivo de `kubeconfig` para el clúster. La configuración de este archivo permite a la CLI de `kubectl` comunicarse con el clúster.

Para configurar el equipo a fin de comunicarse con el clúster

1. Cree o actualice un archivo de `kubeconfig` para el clúster. Reemplace *`region-code`* con la Región de AWS en la creó el clúster. Reemplace *`my-cluster`* por el nombre del clúster.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   De forma predeterminada, el archivo de `config` se crea en `~/.kube` o la configuración del clúster nuevo se agrega a un archivo de `config` existente en `~/.kube`.

2. Pruebe la configuración.

   ```
   kubectl get svc
   ```

   > **Note**
   >
   > Si recibe cualquier error de tipo de recurso o autorización, consulte Acceso denegado o no autorizado (`kubectl`) en el tema de solución de problemas.

   Un ejemplo de salida sería el siguiente.

   ```
   NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
   svc/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   1m
   ```

Paso 3: crear nodos

> **Important**
>
> Para comenzar de la manera más sencilla y rápida posible, en este tema se incluyen pasos a fin de crear nodos con la configuración predeterminada. Antes de crear nodos para su uso en producción, recomendamos que conozca toda la configuración e implemente nodos con la configuración que satisfaga sus requisitos. Para obtener más información, consulte Nodos de Amazon EKS. Algunos ajustes de configuración solo se pueden habilitar al crear los nodos.

Puede crear un clúster con uno de los siguientes tipos de nodos. Para obtener más información sobre cada tipo, consulte Nodos de Amazon EKS. Después de implementar el clúster, puede agregar otros tipos de nodos.

- Fargate – Linux: elija este tipo de nodo si desea ejecutar aplicaciones de Linux en AWS Fargate. Fargate es un motor de computación sin servidor que le permite implementar Pods de Kubernetes sin administrar instancias de Amazon EC2.
- Nodos administrados – Linux: elija este tipo de nodo si desea ejecutar aplicaciones de Amazon Linux en instancias de Amazon EC2. Aunque no se trata en esta guía, también puede agregar nodos autoadministrados de Windows y nodos Bottlerocket su clúster.

Fargate – Linux

Crear un perfil de Fargate. Cuando los Pods de Kubernetes se implementan con criterios que coinciden con los criterios definidos en el perfil, los Pods se implementan en Fargate.

Para crear un perfil de Fargate

1. Cree un rol de IAM y adjúntelo a la política administrada de IAM de Amazon EKS. Cuando su clúster crea Pods en infraestructura de Fargate, los componentes que se ejecutan en la infraestructura de Fargate deben hacer llamadas a las API de AWS en su nombre. Es así para que puedan realizar acciones, como extraer imágenes de contenedores de Amazon ECR o enrutar registros a otros servicios de AWS. El rol de ejecución de Pod de Amazon EKS proporciona los permisos de IAM para esta tarea.

   a. Copie el siguiente contenido en un archivo denominado *pod-execution-role-trust-policy*.json. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Si desea utilizar la misma función en todas las Regiones de AWS en su cuenta, reemplace *region-code* con *. Reemplace *111122223333* por el nombre del clúster y *my-cluster* por el ID de la cuenta. Si quiere utilizar el mismo rol para todos los clústeres de su cuenta, reemplace *my-cluster* con *.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
```

```
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Cree un rol de IAM de ejecución de Pod.

```
aws iam create-role \
  --role-name AmazonEKSFargatePodExecutionRole \
  --assume-role-policy-document file://"pod-execution-role-trust-
policy.json"
```

c. Adjunte la política administrada de IAM por Amazon EKS requerida al rol.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/
AmazonEKSFargatePodExecutionRolePolicy \
  --role-name AmazonEKSFargatePodExecutionRole
```

2. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
3. En la página Clusters (Clústeres), elija el clúster *my-cluster*.
4. En la página ***my-cluster***, haga lo siguiente:

   a. Elija la pestaña Computación.

   b. En Fargate profiles (Perfiles de Fargate), elija Add Fargate profile (Agregar perfil de Fargate).

5. En la página Configure Fargate Profile (Configurar perfil de Fargate), haga lo siguiente:

   a. En Name (Nombre), ingrese un nombre único para su perfil de Fargate; por ejemplo, ***my-profile***.

   b. En Pod execution role (Rol de ejecución de pods), elija el rol AmazonEKSFargatePodExecutionRole que ha creado en el paso anterior.

c. Elija el menú desplegable Subnets (Subredes) y anule la selección de cualquier subred con `Public` en su nombre. Solo las subredes privadas son compatibles con los Pods que se ejecutan en Fargate.

d. Elija Siguiente.

6. En la página Configure Pod selection (Configurar la selección de), haga lo siguiente:

a. En Namespace (Espacio de nombres), escriba **`default`**.

b. Elija Siguiente.

7. En la página Revisar y crear, revise la información de su perfil de Fargate y elija Crear.

8. Al cabo de unos minutos, el Status (Estado) en Fargate Profile configuration (Configuración de perfil de Fargate) cambiará de Creating (En creación) a Active (Activo). No siga con el paso siguiente hasta que el estado sea Active (Activo).

9. Si planea implementar todos los Pods en Fargate (ninguno en los nodos de Amazon EC2), haga lo siguiente para crear otro perfil de Fargate y ejecutar el solucionador de nombres predeterminado (CoreDNS) en Fargate.

> **Note**
>
> Si no hace esto, no tendrá ningún nodo en este momento.

a. En la página Fargate Profile (Perfil de Fargate), elija *`my-profile`*.

b. En Fargate profiles (Perfiles de Fargate), elija Add Fargate profile (Agregar perfil de Fargate).

c. En Nombre, escriba ***`CoreDNS`***.

d. En Pod execution role (Rol de ejecución de pods), elija el rol AmazonEKSFargatePodExecutionRole que ha creado en el paso anterior.

e. Elija el menú desplegable Subnets (Subredes) y anule la selección de cualquier subred con `Public` en su nombre. Solo las subredes privadas son compatibles con los Pods que se ejecutan en Fargate.

f. Elija Siguiente.

g. En Namespace (Espacio de nombres), escriba **`kube-system`**.

h. Elija Match labels (Etiquetas de coincidencia) y, luego, elija Add label (Agregar etiqueta).

i. Ingrese **k8s-app** en Key (Clave) y **kube-dns** en Value (Valor). Esto es necesario para que el solucionador de nombres predeterminado (CoreDNS) se implemente en Fargate.

j. Elija Siguiente.

k. En la página Revisar y crear, revise la información de su perfil de Fargate y elija Crear.

l. Ejecute el siguiente comando para eliminar la anotación `eks.amazonaws.com/compute-type : ec2`predeterminada de los Pods de CoreDNS.

```
kubectl patch deployment coredns \
    -n kube-system \
    --type json \
    -p='[{"op": "remove", "path": "/spec/template/metadata/annotations/
eks.amazonaws.com~1compute-type"}]'
```

> **Note**
>
> El sistema crea e implementa dos nodos según la etiqueta de perfil de Fargate que agregó. No verá nada en la lista Node Groups (Grupos de nodos) porque no se aplican a los nodos de Fargate, pero observará los nodos nuevos en la pestaña Overview (Información general).

Managed nodes – Linux

Cree un grupo de nodos administrados, al especificar las subredes y el rol de IAM de nodo que creó en los pasos anteriores.

Para crear el grupo de nodos administrados por Linux de Amazon EC2

1. Cree un rol de IAM de nodo y adjúntelo a la política administrada de IAM de Amazon EKS. El daemon de `kubelet` del nodo de Amazon EKS realiza llamadas a las API de AWS en su nombre. Los nodos reciben permisos de dichas llamadas de API a través de políticas asociadas y de un perfil de instancias de IAM.

   a. Copie el siguiente contenido en un archivo denominado *`node-role-trust-policy.json`*.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Cree el rol de IAM de nodo.

```
aws iam create-role \
  --role-name myAmazonEKSNodeRole \
  --assume-role-policy-document file://"node-role-trust-policy.json"
```

c. Adjunte las políticas de IAM administradas requeridas al rol.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name myAmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name myAmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
  --role-name myAmazonEKSNodeRole
```

2. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
3. Elija el nombre del clúster que creó en Paso 1: crear el clúster de Amazon EKS, por ejemplo, *my-cluster*.
4. En la página ***my-cluster***, haga lo siguiente:
   a. Elija la pestaña Compute (Computación).
   b. Elija Add Node Group (Agregar grupo de nodos).
5. En la página Configure Node Group (Configurar grupo de nodos), haga lo siguiente:
   a. En Name (Nombre), ingrese un nombre único para el grupo de nodos administrados, por ejemplo, ***my-nodegroup***. El nombre del grupo de nodos no puede tener más de

63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.

b. En Node IAM role name (Nombre de rol de IAM de nodo), elija el rol *myAmazonEKSNodeRole* que creó en el paso anterior. Recomendamos que cada grupo de nodos utilice su propio rol de IAM exclusivo.

c. Elija Siguiente.

6. En la página Set compute and scaling configuration (Establecer la configuración de computación y escalado), acepte los valores predeterminados y elija Next (Siguiente).
7. En la página Specify networking (Especificar redes), acepte los valores predeterminados y elija Next (Siguiente).
8. En la página Revisar y crear, revise la configuración del grupo de nodos administrados y elija Crear.
9. Al cabo de varios minutos, el Status (Estado) en la Node Group configuration (Configuración del grupo de nodos) cambiará de Creating (En creación) a Active (Activo). No siga con el paso siguiente hasta que el estado sea Active (Activo).

Paso 4: ver recursos

Puede ver sus nodos y cargas de trabajo de Kubernetes.

Para ver los nodos y las cargas de trabajo

1. En el panel de navegación izquierdo, elija Clusters (Clústeres). En la lista de Clusters (Clústeres), elija el nombre del clúster que ha creado, como *my-cluster*.
2. En la página ***my-cluster***, elija lo siguiente:

   a. En la pestaña Compute (Informática), verá la lista de Nodes (Nodos) que se implementaron para el clúster. Puede elegir el nombre de un nodo para obtener más información sobre él.

   b. En la pestaña Resources (Recursos): consulte todos los recursos de Kubernetes que se implementan de forma predeterminada en un clúster de Amazon EKS. Seleccione cualquier tipo de recurso en la consola de para obtener más información sobre él.

Paso 5: eliminar recursos

Cuando haya terminado con el clúster y los nodos que creó para este tutorial, deberá eliminar los recursos creados. Si desea hacer más con este clúster antes de eliminar los recursos, consulte Siguientes pasos.

Para eliminar los recursos que creó en esta guía

1. Elimine cualquier grupo de nodos o perfil de Fargate que haya creado.
   a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
   b. En el panel de navegación izquierdo, elija Clusters (Clústeres). En la lista de clústeres, elija *my-cluster*.
   c. Elija la pestaña Computación.
   d. Si ha creado un grupo de nodos, lija el grupo de nodos *my-nodegroup* y, luego, elija Delete (Eliminar). Introduzca ***my-nodegroup*** y luego escriba Delete (Eliminar).
   e. Para cada perfil de Fargate que haya creado, elíjalo y, luego, elija Delete (Eliminar). Ingrese el nombre del perfil y, a continuación, elija Delete (Eliminar).

      > **Note**
      >
      > Si elimina un segundo perfil de Fargate, es posible que tenga que esperar a que se termine de eliminar el primero.

   f. No continúe hasta que el grupo de nodos o los perfiles de Fargate se eliminen.
2. Eliminar el clúster.
   a. En el panel de navegación izquierdo, elija Clusters (Clústeres). En la lista de clústeres, elija *my-cluster*.
   b. Seleccione Delete cluster (Eliminar clúster).
   c. Introduzca ***my-cluster*** y luego elija Delete (Eliminar). No continúe hasta que se elimine el clúster.
3. Elimine la pila de AWS CloudFormation de la VPC que ha creado.
   a. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
   b. Elija la pila *my-eks-vpc-stack* y, a continuación, elija Delete (Eliminar).

c. En el cuadro de diálogo de confirmación Delete ***my-eks-vpc-stack*** (Eliminar my-eks-vpc-stack), elija Delete stack (Eliminar pila).

4. Elimine los roles de IAM que ha creado.

   a. Abra la consola de IAM en https://console.aws.amazon.com/iam/.

   b. En el panel de navegación izquierdo, seleccione Roles.

   c. Seleccione todos los roles que creó en la lista (***myAmazonEKSClusterRole***, así como AmazonEKSFargatePodExecutionRole o *myAmazonEKSNodeRole*). Elija Delete (Eliminar), ingrese el texto de confirmación solicitado y, luego, elija Delete (Eliminar).

Siguientes pasos

Los siguientes temas de documentación lo ayudarán a ampliar la funcionalidad de su clúster.

- La entidad principal de IAM que creó el clúster es la única entidad principal que puede realizar llamadas al servidor de la API de Kubernetes con `kubectl` o laAWS Management Console. Si desea que otras entidades principales de IAM tengan acceso al clúster, debe agregarlas. Para obtener más información, consulte Concesión de acceso a las API de Kubernetes y Permisos necesarios.
- Implemente una aplicación de muestra en su clúster.
- Antes de implementar un clúster para su uso en producción, le recomendamos que se familiarice con toda la configuración de clústeres y nodos. Algunos ajustes (como habilitar el acceso SSH a los nodos de Amazon EC2) deben establecerse cuando se crea el clúster.
- Para aumentar la seguridad del clúster, configure el complemento de interfaz de red de contenedores de Amazon VPC a fin de utilizar roles de IAM para cuentas de servicio.

Clústeres de Amazon EKS

Los clústeres de Amazon EKS constan de dos componentes principales:

- El plano de control de Amazon EKS
- Los nodos de Amazon EKS registrados en el plano de control

El plano de control de Amazon EKS consta de nodos del plano de control que ejecutan el software de Kubernetes, como `etcd` y el servidor de la API de Kubernetes. El plano de control se ejecuta con una cuenta administrada por AWS y la API de Kubernetes se expone a través del punto de conexión de Amazon EKS asociado al clúster. Cada plano de control del clúster de Amazon EKS es de inquilino único y particular, y se ejecuta en su propio conjunto de instancias de Amazon EC2.

Todos los datos almacenados por los nodos de `etcd` y los volúmenes asociados de Amazon EBS se cifran mediante AWS KMS. El plano de control del clúster se aprovisiona en varias zonas de disponibilidad y se presenta por un Network Load Balancer de Elastic Load Balancing. Amazon EKS también aprovisiona interfaces de red elásticas en sus subredes de VPC para proporcionar conectividad desde las instancias del plano de control a los nodos (por ejemplo, a fin de admitir flujos de datos `kubectl exec logs proxy`).

Important

En el entorno de Amazon EKS, el almacenamiento de `etcd` está limitado a 8 GiB según la orientación ascendente. Puede supervisar una métrica del tamaño actual de la base de datos al ejecutar el siguiente comando. Si el clúster tiene una versión de Kubernetes inferior a `1.28`, reemplace *`apiserver_storage_size_bytes`* por lo siguiente:

- Versiones `1.27` y `1.26` de Kubernetes: **`apiserver_storage_db_total_size_in_bytes`**
- Versión `1.25` y versiones anteriores de Kubernetes: **`etcd_db_total_size_in_bytes`**

```
kubectl get --raw=/metrics | grep "apiserver_storage_size_bytes"
```

Los nodos de Amazon EKS se ejecutan con su cuenta de AWS y lo conectan con el plano de control del clúster a través del punto de conexión del servidor de la API y un archivo de certificado creado para el clúster.

> **Note**
>
> - Puede descubrir cómo funcionan los diferentes componentes de Amazon EKS en Redes de Amazon EKS.
> - Para obtener información sobre clústeres conectados, consulte Amazon EKS Connector.

Temas

Creación de un clúster de Amazon EKS

En este tema, se ofrece información general de las opciones disponibles y se describe qué debe tener en cuenta al crear un clúster de Amazon EKS. Si necesita crear un clúster en un Outpost de AWS, consulte Clústeres locales para Amazon EKS en AWS Outposts. Si es la primera vez que crea un clúster de Amazon EKS, recomendamos que siga una de nuestras guías de Introducción a Amazon EKS. Estas guías le ayudan a crear un clúster simple y predeterminado sin expandirse a todas las opciones disponibles.

Requisitos previos

- Una VPC y subredes existentes que cumplen con los Requisitos de Amazon EKS. Antes de implementar un clúster para su uso en producción, le recomendamos que conozca a fondo los requisitos de VPC y subred. Si no tiene una VPC y subredes, puede crearlas utilizando una plantilla AWS CloudFormation Amazon EKS proporcionada.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- Una entidad principal de IAM con permisos para `create` y `describe` un clúster de Amazon EKS. Para obtener más información, consulte Crea un clúster local de Kubernetes en un Outpost y Enumeración o descripción de todos los clústeres.

Crear un clúster de Amazon EKS

1. Si ya tiene un rol de IAM de clúster o va a crear su clúster con `eksctl`, puede omitir este paso. Por defecto, `eksctl` crea un rol para usted.

   Para crear el rol de IAM del clúster de Amazon EKS

   1. Ejecute el siguiente comando para crear un archivo de política de confianza JSON de IAM.

      ```
      cat >eks-cluster-role-trust-policy.json <<EOF
      {
        "Version": "2012-10-17",
      ```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

2. Cree el rol de IAM del clúster de Amazon EKS. Si es necesario, introduzca *eks-cluster-role-trust-policy.json* con la ruta del equipo en la que escribió el archivo en el paso anterior. El comando asocia la política de confianza creada en el paso anterior al rol. Para crear un rol de IAM, a la entidad principal de IAM que está creando el rol se le debe asignar la acción `iam:CreateRole` (permiso).

```
aws iam create-role --role-name myAmazonEKSClusterRole --assume-role-policy-
document file://"eks-cluster-role-trust-policy.json"
```

3. Puede asignar la política administrada de Amazon EKS o bien crear su propia política personalizada. Para conocer los permisos mínimos que debe utilizar en su política personalizada, consulte Rol de IAM del clúster de Amazon EKS.

   Adjunte la política administrada de IAM por Amazon EKS denominada `AmazonEKSClusterPolicy` al rol. Para adjuntar una política de IAM a una entidad principal de IAM, se debe asignar una de las siguientes acciones de IAM (permisos) a la entidad principal que adjunta la política: `iam:AttachUserPolicy` o `iam:AttachRolePolicy`.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name myAmazonEKSClusterRole
```

2. Cree un clúster de Amazon EKS.

   Puede crear un clúster mediante la `eksctl`, la AWS Management Console o la AWS CLI.

   eksctl

   Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Cómo crear el clúster

Cree un clúster de Amazon EKS `IPv4` con la versión más reciente de Kubernetes de Amazon EKS en su Región de AWS predeterminada. Antes de ejecutar el comando, realice los siguientes reemplazos:

- Reemplace *`region-code`* por la Región de AWS en la que desea implementar sus recursos.
- Reemplace *`my-cluster`* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
- Reemplace *`1.29`* por cualquier versión compatible con Amazon EKS.

  > **Note**
  >
  > Para implementar un clúster 1.30 en este momento, debe utilizar el AWS Management Console o el AWS CLI.

- Cambie los valores de `vpc-private-subnets` para satisfacer sus necesidades. También puede agregar identificadores adicionales. Debe especificar al menos dos ID de subredes. Si prefiere especificar subredes públicas, puede cambiar `--vpc-private-subnets` a `--vpc-public-subnets`. Las subredes públicas tienen una tabla de enrutamiento asociada a una ruta a una puerta de enlace de Internet, pero las subredes privadas no tienen una tabla de enrutamiento asociada. Recomendamos utilizar subredes privadas siempre que sea posible.

  Las subredes que elija deben cumplir los Requisitos de subred de Amazon EKS. Antes de seleccionar subredes, le recomendamos que esté familiarizado con todas las consideraciones y requisitos de subred y VPC de Amazon EKS.

```
eksctl create cluster --name my-cluster --region region-code --version 1.29 --
vpc-private-subnets subnet-ExampleID1,subnet-ExampleID2 --without-nodegroup
```

El aprovisionamiento de clústeres tarda varios minutos. Mientras se crea el clúster, aparecen varias líneas de salida. La última línea de salida es similar a la siguiente línea de ejemplo.

```
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

> **Tip**
>
> Para ver la mayoría de las opciones que se pueden especificar al crear un clúster con `eksctl`, utilice el comando **`eksctl create cluster --help`**. Para ver todas las opciones disponibles, puede utilizar un archivo `config`. Para obtener más información, consulte Uso de archivos de configuración y el esquema de archivos de configuración en la documentación de `eksctl`. Puede encontrar ejemplos de archivos de configuración en GitHub.

Configuración opcional

A continuación se muestra una configuración opcional que, si es necesario, debe agregarse al comando anterior. Solo puede habilitar estas opciones al crear el clúster, no después. Si necesita especificar estas opciones, debe crear el clúster con un archivo de configuración `eksctl` y especifique la configuración, en lugar de utilizar el comando anterior.

- Si desea especificar uno o varios grupos de seguridad que Amazon EKS asigna a las interfaces de red que crea, especifique la opción `securityGroup`.

  Independientemente de que elija grupos de seguridad o no, Amazon EKS crea un grupo de seguridad que permite la comunicación entre el clúster y la VPC. Amazon EKS asocia este grupo de seguridad, y el que elija, a las interfaces de red que crea. Para obtener más información acerca del grupo de seguridad de clúster que crea Amazon EKS, consulte the section called “Requisitos del grupo de seguridad”. Puede modificar las reglas del grupo de seguridad en el clúster que crea Amazon EKS.

- Si quiere especificar desde qué bloque de enrutamiento entre dominios sin clase (CIDR) `IPv4` Kubernetes asigna direcciones IP de servicio, especifique la opción `serviceIPv4CIDR`.

  Especificar su propio intervalo puede ayudar a evitar conflictos entre servicios de Kubernetes y otras redes interconectadas o conectadas a la VPC. Escriba un rango en notación CIDR. Por ejemplo: `10.2.0.0/16`.

  El bloque de CIDR debe cumplir los siguientes requisitos:

  - Se encuentra dentro de una de las siguientes gamas: `10.0.0.0/8`, `172.16.0.0/12` o `192.168.0.0/16`.
  - Tiene un tamaño mínimo de `/24` y un tamaño máximo de `/12`.
  - No se superponen con el rango de la VPC de los recursos de Amazon EKS.

  Solo se puede especificar esta opción cuando se utiliza la familia `IPv4` de direcciones y solo en la creación de clústeres. Si no especifica esto, entonces Kubernetes asigna direcciones IP de servicio desde los bloques CIDR `10.100.0.0/16` o `172.20.0.0/16`.
- Si va a crear un clúster y desea que el clúster asigne direcciones `IPv6` a Pods y servicios en lugar de direcciones `IPv4`, especifique la opción `ipFamily`.

  Kubernetes asigna direcciones `IPv4` a los Pods y servicios de manera predeterminada. Antes de decidir utilizar la familia `IPv6`, asegúrese de estar familiarizado con todas las consideraciones y requisitos en los temas the section called “Requisitos y consideraciones de la VPC”, the section called “Requisitos y consideraciones de la subred”, the section called “Requisitos del grupo de seguridad” y the section called “IPv6”. Si elige la familia `IPv6`, no puede especificar un intervalo de direcciones para que Kubernetes asigne direcciones de servicio desde `IPv6` como puede hacer para la familia `IPv4`. Kubernetes asigna direcciones de servicio del rango de direcciones local exclusivo (`fc00::/7`).

AWS Management Console

Cómo crear el clúster

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija Agregar clúster y, a continuación, elija Crear.
3. En la página Configurar clúster rellene los siguientes campos:

- Nombre: un nombre único para el clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones bajos. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
- Versión de Kubernetes: La versión de Kubernetes que debe utilizarse para el clúster. Le recomendamos que seleccione la versión más reciente, a menos que necesite una versión anterior.
- Rol de servicio del clúster: elija el rol de IAM del clúster de Amazon EKS que creó para permitir que el plano de control de Kubernetes administre los recursos de AWS en su nombre.
- Cifrado de secretos: (opcional) elija habilitar el cifrado de secretos de los secretos de Kubernetes con una clave de KMS. También puede habilitarlo después de crear el clúster. Antes de habilitar esta capacidad, asegúrese de estar familiarizado con la información de Habilitación del cifrado de secretos en un clúster existente.
- Etiquetas: (opcional) agregue las etiquetas a su clúster. Para obtener más información, consulte Etiquetado de los recursos de Amazon EKS.

  Cuando haya terminado con esta página, seleccione Siguiente.

4. En la página Especificar red seleccione valores para los siguientes campos:
   - VPC: Elija una VPC existente que cumpla con los Requisitos de Amazon EKS VPC en el que crear el clúster. Antes de elegir una VPC, le recomendamos que esté familiarizado con todos los requisitos y consideraciones de Requisitos y consideraciones de Amazon EKS VPC y subred. No puede cambiar la VPC que desea utilizar después de crear un clúster. Si no hay ninguna VPC en la lista, debe crear una primero. Para obtener más información, consulte Creación de una VPC para su clúster de Amazon EKS.
   - Subredes: de forma predeterminada, se preseleccionan todas las subredes disponibles de la VPC especificada en el campo anterior. Debe seleccionar al menos dos.

     Las subredes que elija deben cumplir los Requisitos de subred de Amazon EKS. Antes de seleccionar subredes, le recomendamos que esté familiarizado con todas las consideraciones y requisitos de subred y VPC de Amazon EKS.

     Grupos de seguridad: (opcional) especifique uno o varios grupos de seguridad que desea que Amazon EKS asocie a las interfaces de red que crea.

Independientemente de que elija grupos de seguridad o no, Amazon EKS crea un grupo de seguridad que permite la comunicación entre el clúster y la VPC. Amazon EKS asocia este grupo de seguridad, y el que elija, a las interfaces de red que crea. Para obtener más información acerca del grupo de seguridad de clúster que crea Amazon EKS, consulte the section called “Requisitos del grupo de seguridad”. Puede modificar las reglas del grupo de seguridad en el clúster que crea Amazon EKS.

- Elija la familia de direcciones IP del clúster: Puede elegir IPv4 e IPv6.

  Kubernetes asigna direcciones `IPv4` a los Pods y servicios de manera predeterminada. Antes de decidir utilizar la familia `IPv6`, asegúrese de estar familiarizado con todas las consideraciones y requisitos en los temas the section called “Requisitos y consideraciones de la VPC”, the section called “Requisitos y consideraciones de la subred”, the section called “Requisitos del grupo de seguridad” y the section called “IPv6”. Si elige la familia `IPv6`, no puede especificar un intervalo de direcciones para que Kubernetes asigne direcciones de servicio desde `IPv6` como puede hacer para la familia `IPv4`. Kubernetes asigna direcciones de servicio del rango de direcciones local exclusivo (`fc00::/7`).

- (Opcional) Elija Configuración del rango de direcciones IP de Kubernetes Service y especifique un Rango de servicio **`IPv4`**.

  Especificar su propio intervalo puede ayudar a evitar conflictos entre servicios de Kubernetes y otras redes interconectadas o conectadas a la VPC. Escriba un rango en notación CIDR. Por ejemplo: `10.2.0.0/16`.

  El bloque de CIDR debe cumplir los siguientes requisitos:

  - Se encuentra dentro de una de las siguientes gamas: `10.0.0.0/8`, `172.16.0.0/12` o `192.168.0.0/16`.
  - Tiene un tamaño mínimo de `/24` y un tamaño máximo de `/12`.
  - No se superponen con el rango de la VPC de los recursos de Amazon EKS.

  Solo se puede especificar esta opción cuando se utiliza la familia `IPv4` de direcciones y solo en la creación de clústeres. Si no especifica esto, entonces Kubernetes asigna direcciones IP de servicio desde los bloques CIDR `10.100.0.0/16` o `172.20.0.0/16`.

- Para el Acceso al punto de conexión del clúster, seleccione una opción. Una vez que se crea el clúster, puede cambiar esta opción. Antes de seleccionar una opción no predeterminada, asegúrese de familiarizarse con las opciones y sus implicaciones. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS.

  Cuando haya terminado con esta página, seleccione Siguiente.

5. (Opcional) En la página Configurar la observabilidad, seleccione qué opciones de métricas y registro de planos de control quiere activar. De forma predeterminada, cada tipo de registro está desactivado.
   - Para obtener más información sobre la opción de las métricas de Prometheus, consulte Active las métricas de Prometheus al crear un clúster.
   - Para obtener más información sobre las opciones de registro de plano de control, consulte Registro de plano de control de Amazon EKS.

   Cuando haya terminado con esta página, seleccione Siguiente.
6. En la página Seleccionar complementos, elija los complementos que desea agregar al clúster. Puede elegir tantos complementos de Amazon EKS y complementos de AWS Marketplace como necesite. Si los complementos de AWS Marketplace que quiere instalar no aparecen en la lista, puede buscar los complementos de AWS Marketplace disponibles al introducir texto en el cuadro de búsqueda. También puede buscar por categoría, proveedor o modelo de precios y, a continuación, elegir los complementos en los resultados de la búsqueda. Cuando haya terminado con esta página, seleccione Siguiente.
7. En la página Configurar las opciones de complementos seleccionados, seleccione la versión que desee instalar. Siempre puede actualizar a una versión posterior después de crear el clúster. Puede actualizar la configuración de cada complemento después de crear el clúster. Para obtener más información acerca de la configuración de complementos, consulte Actualización de un complemento. Cuando haya terminado con esta página, seleccione Siguiente.
8. En la página Revisar y crear, revise la información que introdujo o seleccionó en las páginas anteriores. Si necesita realizar cambios, seleccione Edit (Editar). Cuando esté satisfecho, seleccione Create (Crear). El campo Status (Estado) muestra CREATING (CREANDO) mientras se aprovisiona el clúster.

> **Note**
>
> Es posible que reciba un error que indique que una de las zonas de disponibilidad de la solicitud no tiene capacidad suficiente para crear un clúster de Amazon EKS. Si esto ocurre, el mensaje de error indicará las zonas de disponibilidad que admiten un clúster nuevo. Intente crear el clúster de nuevo con al menos dos subredes ubicadas en las zonas de disponibilidad admitidas para su cuenta. Para obtener más información, consulte Capacidad insuficiente.

El aprovisionamiento de clústeres tarda varios minutos.

AWS CLI

Cómo crear el clúster

1. Cree el clúster con el siguiente comando. Antes de ejecutar el comando, realice los siguientes reemplazos:
   - Reemplace *region-code* por la Región de AWS en la que desea implementar sus recursos.
   - Reemplace *my-cluster* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones bajos. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
   - Reemplace *1.30* por cualquier versión compatible con Amazon EKS.
   - Reemplace *111122223333* por el ID de la cuenta y *myAmazonEKSClusterRole* por el nombre rol de IAM de su clúster.
   - Reemplace los valores `subnetIds` con valores propios. También puede agregar identificadores adicionales. Debe especificar al menos dos ID de subredes.

     Las subredes que elija deben cumplir los Requisitos de subred de Amazon EKS. Antes de seleccionar subredes, le recomendamos que esté familiarizado con todas las consideraciones y requisitos de subred y VPC de Amazon EKS.

- Si no desea especificar un ID de grupo de seguridad, elimine `,securityGroupIds=sg-`*`ExampleID1`* del comando. Si desea especificar uno o varios ID de grupo de seguridad, sustituya los valores de `securityGroupIds` con la suya propia. También puede agregar identificadores adicionales.

  Independientemente de que elija grupos de seguridad o no, Amazon EKS crea un grupo de seguridad que permite la comunicación entre el clúster y la VPC. Amazon EKS asocia este grupo de seguridad, y el que elija, a las interfaces de red que crea. Para obtener más información acerca del grupo de seguridad de clúster que crea Amazon EKS, consulte the section called “Requisitos del grupo de seguridad”. Puede modificar las reglas del grupo de seguridad en el clúster que crea Amazon EKS.

```
aws eks create-cluster --region region-code --name my-cluster --kubernetes-
version 1.30 \
   --role-arn arn:aws:iam::111122223333:role/myAmazonEKSClusterRole \
   --resources-vpc-config
 subnetIds=subnet-ExampleID1,subnet-ExampleID2,securityGroupIds=sg-ExampleID1
```

> **Note**
>
> Es posible que reciba un error que indique que una de las zonas de disponibilidad de la solicitud no tiene capacidad suficiente para crear un clúster de Amazon EKS. Si esto ocurre, el mensaje de error indicará las zonas de disponibilidad que admiten un clúster nuevo. Intente crear el clúster de nuevo con al menos dos subredes ubicadas en las zonas de disponibilidad admitidas para su cuenta. Para obtener más información, consulte Capacidad insuficiente.

Configuración opcional

A continuación se muestra una configuración opcional que, si es necesario, debe agregarse al comando anterior. Solo puede habilitar estas opciones al crear el clúster, no después.

- Si quiere especificar desde qué bloque de enrutamiento entre dominios sin clase (CIDR) `IPv4` Kubernetes asigna direcciones IP de servicio, debe especificarlo agregando el **`--kubernetes-network-config serviceIpv4Cidr=`***`CIDR block`* al siguiente comando.

Especificar su propio intervalo puede ayudar a evitar conflictos entre servicios de Kubernetes y otras redes interconectadas o conectadas a la VPC. Escriba un rango en notación CIDR. Por ejemplo: `10.2.0.0/16`.

El bloque de CIDR debe cumplir los siguientes requisitos:

- Se encuentra dentro de una de las siguientes gamas: `10.0.0.0/8`, `172.16.0.0/12` o `192.168.0.0/16`.
- Tiene un tamaño mínimo de `/24` y un tamaño máximo de `/12`.
- No se superponen con el rango de la VPC de los recursos de Amazon EKS.

Solo se puede especificar esta opción cuando se utiliza la familia `IPv4` de direcciones y solo en la creación de clústeres. Si no especifica esto, entonces Kubernetes asigna direcciones IP de servicio desde los bloques CIDR `10.100.0.0/16` o `172.20.0.0/16`.

- Si está creando un clúster y desea que el clúster asigne direcciones `IPv6` a Pods y servicios en lugar de direcciones `IPv4`, agregue **`--kubernetes-network-config ipFamily=ipv6`** al siguiente comando.

  Kubernetes asigna direcciones `IPv4` a los Pods y servicios de manera predeterminada. Antes de decidir utilizar la familia `IPv6`, asegúrese de estar familiarizado con todas las consideraciones y requisitos en los temas the section called “Requisitos y consideraciones de la VPC”, the section called “Requisitos y consideraciones de la subred”, the section called “Requisitos del grupo de seguridad” y the section called “IPv6”. Si elige la familia `IPv6`, no puede especificar un intervalo de direcciones para que Kubernetes asigne direcciones de servicio desde `IPv6` como puede hacer para la familia `IPv4`. Kubernetes asigna direcciones de servicio del rango de direcciones local exclusivo (`fc00::/7`).

2. La provisión del clúster puede tardar varios minutos. Puede consultar el estado del clúster con el siguiente comando.

```
aws eks describe-cluster --region region-code --name my-cluster --query
 "cluster.status"
```

No continúe con el siguiente paso hasta que la salida devuelta sea `ACTIVE`.

3. Si ha creado el clúster mediante `eksctl`, puede omitir este paso. Esto se debe a que `eksctl` ya ha completado este paso por usted. Habilite `kubectl` para comunicarse con el clúster agregando un nuevo contexto al archivo `kubectl config`. Para obtener más información acerca de cómo crear y actualizar el archivo, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   Added new context arn:aws:eks:region-code:111122223333:cluster/my-cluster to /home/username/.kube/config
   ```

4. Confirme la comunicación con el clúster ejecutando el siguiente comando.

   ```
   kubectl get svc
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
   kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   28h
   ```

5. (Recomendado) Para utilizar algunos complementos de Amazon EKS, o para permitir que las cargas de trabajo individuales de Kubernetes tengan permisos específicos de AWS Identity and Access Management (IAM), cree un proveedor de OpenID Connect (OIDC) de IAM para su clúster. Solo tiene que crear un proveedor de OIDC de IAM para su clúster una vez. Para obtener más información sobre los complementos de Amazon EKS, consulte Complementos de Amazon EKS. Para obtener más información sobre la asignación de permisos de IAM específicos a sus cargas de trabajo, consulte Roles de IAM para cuentas de servicio.

6. (Recomendado) Configure el clúster para el complemento Amazon VPC CNI plugin for Kubernetes antes de implementar nodos Amazon EC2 en su clúster. De forma predeterminada, el complemento se instaló con el clúster. Cuando agrega nodos de Amazon EC2 a su clúster, el complemento se implementa automáticamente en cada nodo de Amazon EC2 que agregue. El complemento requiere que adjunte una de las siguientes políticas de IAM a un rol de IAM:

   Política de IAM administrada de **`AmazonEKS_CNI_Policy`**

   Si el clúster usa la familia `IPv4`

Una política de IAM que usted cree

Si el clúster usa la familia `IPv6`

El rol de IAM al que adjunta la política puede ser el rol de IAM de nodo o un rol dedicado que se usa solo para el complemento. Recomendamos adjuntar la política a este rol. Para obtener más información sobre la creación del rol, consulte la Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA) o la Rol de IAM de nodo de Amazon EKS.

7. Si ha implementado el clúster mediante el AWS Management Console, puede omitir este paso. La AWS Management Console implementa los complementos Amazon VPC CNI plugin for Kubernetes, CoreDNS y `kube-proxy` de Amazon EKS, de forma predeterminada.

   Si implementa el clúster mediante `eksctl` o la AWS CLI, entonces se implementan los complementos Amazon VPC CNI plugin for Kubernetes, CoreDNS y `kube-proxy` autoadministrados. Puede migrar los complementos autoadministrados Amazon VPC CNI plugin for Kubernetes, CoreDNS y `kube-proxy` que se implementan con su clúster en complementos de Amazon EKS. Para obtener más información, consulte Complementos de Amazon EKS.
8. (Opcional) Si aún no lo ha hecho, puede habilitar las métricas de Prometheus para su clúster. Para obtener más información, consulte Crear un raspador en la Guía del usuario de Amazon Managed Service for Prometheus.
9. Si ha activado las métricas de Prometheus, debe configurar las `aws-auth ConfigMap` para conceder al raspador permisos dentro del clúster. Para obtener más información, consulte Configuración del clúster de Amazon EKS en la Guía del usuario de Amazon Managed Service for Prometheus.
10. Si tiene previsto implementar cargas de trabajo en su clúster que utilicen volúmenes de Amazon EBS y creó un clúster de la versión `1.23` o posterior, deberá instalar el Controlador CSI de Amazon EBS en su clúster antes de implementar las cargas de trabajo.

Siguientes pasos recomendados:

- La entidad principal de IAM que creó el clúster es la única entidad principal de IAM que tiene acceso al clúster. Conceda permisos a otras entidades principales de IAM para que puedan acceder al clúster.
- Si la entidad principal de IAM que creó el clúster solo tiene los permisos de IAM mínimos a los que se hace referencia en los requisitos previos, es posible que desee agregar permisos de Amazon

EKS adicionales para esa entidad principal. Para obtener más información sobre cómo conceder permisos de Amazon EKS a entidades principales de IAM, consulte Administración de identidades y accesos para Amazon EKS.

- Si desea que la entidad principal de IAM que creó el clúster o cualquier otra entidad principal vea los recursos de Kubernetes en la consola de Amazon EKS, conceda los Permisos necesarios a las entidades.
- Si desea que los nodos y las entidades principales de IAM accedan al clúster desde su VPC, habilite el punto de conexión privado para el clúster. El punto de conexión público está habilitado de forma predeterminada. Si lo desea, puede desactivar el punto de conexión público una vez que haya activado el punto de conexión privado. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS.
- Habilite el cifrado de secretos para su clúster.
- Configure el registro para el clúster.
- Agregue nodos al clúster.

Información sobre clústeres

La información sobre los clústeres de Amazon EKS ofrece recomendaciones que lo ayudarán a seguir las prácticas recomendadas de Amazon EKS y Kubernetes. Todos los clústeres de Amazon EKS se someten a comprobaciones automáticas y periódicas con una lista de información seleccionada por Amazon EKS. Amazon EKS administra en su totalidad estas comprobaciones de información y ofrece recomendaciones sobre cómo abordar cualquier resultado.

Uso recomendado de la información sobre clústeres:

- Antes de actualizar la versión de Kubernetes del clúster, consulte la información sobre el clúster en la consola de EKS.
- Si su clúster ha identificado problemas, revíselos y aplique las correcciones adecuadas. Los problemas incluyen enlaces a Amazon EKS y Kubernetes.
- Tras solucionar los problemas, espere a que se actualicen las estadísticas del clúster. Si se han resuelto todos los problemas, actualice el clúster.

Actualmente, Amazon EKS solo devuelve información relacionada con la preparación para la actualización de la versión de Kubernetes.

La información sobre las actualizaciones identifica los posibles problemas que podrían afectar a las actualizaciones del clúster de Kubernetes. Esto minimiza el esfuerzo que los administradores dedican a preparar las actualizaciones y aumenta la fiabilidad de las aplicaciones en las versiones más recientes de Kubernetes. Amazon EKS analiza automáticamente los clústeres para compararlos con una lista de posibles problemas que podrían afectar las actualizaciones de la versión de Kubernetes. Amazon EKS actualiza con frecuencia la lista de comprobaciones de información en función de las revisiones de los cambios realizados en cada lanzamiento de versión de Kubernetes.

La información sobre las actualizaciones de Amazon EKS acelera el proceso de prueba y verificación de las nuevas versiones. También permiten a los administradores de clústeres y a los desarrolladores de aplicaciones aprovechar las capacidades más recientes de Kubernetes, ya que destacan las inquietudes y ofrecen consejos para solucionarlas. Para ver la lista de comprobaciones de información realizadas y cualquier problema relevante que Amazon EKS haya identificado, puede llamar a la operación de la API `ListInsights` de Amazon EKS o buscar en la consola de Amazon EKS.

La información sobre los clústeres se actualiza periódicamente. No puede actualizar manualmente la información del clúster. Si soluciona un problema con el clúster, la información del clúster tardará algún tiempo en actualizarse. Para determinar si una solución se ha aplicado correctamente, compare la hora en que se implementó el cambio con la “hora de la última actualización” de la información del clúster.

Consulta de la información del clúster (consola)

Para ver información sobre un clúster de Amazon EKS:

a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
b. En la lista de clústeres, elija el nombre del clúster de Amazon EKS del que desea ver la información.
c. Seleccione la pestaña Información sobre actualizaciones.
d. En la página Información sobre actualizaciones, verá los siguientes campos:
   - Nombre: la comprobación realizada por Amazon EKS en relación con el clúster.
   - Estado de la información: una información con un estado de “Error” normalmente significa que la versión de Kubernetes afectada es N+1 de la versión actual del clúster, mientras que un estado de “Advertencia” significa que la información se aplica a una versión futura de Kubernetes N +2 o superior. Una información con el estado “Aprobado” significa que Amazon EKS no ha encontrado ningún problema relacionado con esta comprobación de información en su clúster.

Un estado de información “Desconocido” significa que Amazon EKS no puede determinar si su clúster se ve afectado por esta comprobación de información.

- Versión: la versión de Kubernetes que la información comprobó para detectar posibles problemas.
- Hora de la última actualización (UTC-5:00): la hora en que se actualizó por última vez el estado de la información para este clúster.
- Hora de la última transición (UTC-5:00): la hora en que se modificó por última vez el estado de esta información.
- Descripción: información de la comprobación de información, que incluye la alerta y las acciones recomendadas para su corrección.

Consulta de la información del clúster (AWS CLI)

Para ver información sobre un clúster de Amazon EKS:

a. Determine qué clúster desea comprobar para obtener información. El siguiente comando enumera toda la información para el clúster especificado. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado:
   - Reemplace *region-code* por el código de su Región de AWS.
   - Reemplace *my-cluster* por el nombre del clúster.

```
aws eks list-insights --region region-code --cluster-name my-cluster
```

Un ejemplo de salida sería el siguiente.

```
{
    "insights": [
        {
            "category": "UPGRADE_READINESS",
            "name": "Deprecated APIs removed in Kubernetes v1.29",
            "insightStatus": {
                "status": "PASSING",
                "reason": "No deprecated API usage detected within the last 30 days."
            },
            "kubernetesVersion": "1.29",
            "lastTransitionTime": 1698774710.0,
            "lastRefreshTime": 1700157422.0,
            "id": "123e4567-e89b-42d3-a456-579642341238",
```

```
            "description": "Checks for usage of deprecated APIs that are scheduled
 for removal in Kubernetes v1.29. Upgrading your cluster before migrating to the
 updated APIs supported by v1.29 could cause application impact."
        }
    ]
}
```

b. Ejecute el siguiente comando para obtener información descriptiva. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado:

- Reemplace *region-code* por el código de su Región de AWS.
- Reemplace *123e4567-e89b-42d3-a456-579642341238* por el ID de información obtenido de la lista de información del clúster.
- Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-insight --region region-code --id 123e4567-e89b-42d3-
a456-579642341238 --cluster-name my-cluster
```

Un ejemplo de salida sería el siguiente.

```
{
    "insight": {
        "category": "UPGRADE_READINESS",
        "additionalInfo": {
            "EKS update cluster documentation": "https://docs.aws.amazon.com/eks/
latest/userguide/update-cluster.html",
            "Kubernetes v1.29 deprecation guide": "https://kubernetes.io/docs/
reference/using-api/deprecation-guide/#v1-29"
        },
        "name": "Deprecated APIs removed in Kubernetes v1.29",
        "insightStatus": {
            "status": "PASSING",
            "reason": "No deprecated API usage detected within the last 30 days."
        },
        "kubernetesVersion": "1.29",
        "recommendation": "Update manifests and API clients to use newer Kubernetes
 APIs if applicable before upgrading to Kubernetes v1.29.",
        "lastTransitionTime": 1698774710.0,
        "lastRefreshTime": 1700157422.0,
        "categorySpecificSummary": {
            "deprecationDetails": [
```

```
                {
                    "usage": "/apis/flowcontrol.apiserver.k8s.io/v1beta2/
flowschemas",
                    "replacedWith": "/apis/flowcontrol.apiserver.k8s.io/v1beta3/
flowschemas",
                    "stopServingVersion": "1.29",
                    "clientStats": [],
                    "startServingReplacementVersion": "1.26"
                },
                {
                    "usage": "/apis/flowcontrol.apiserver.k8s.io/v1beta2/
prioritylevelconfigurations",
                    "replacedWith": "/apis/flowcontrol.apiserver.k8s.io/v1beta3/
prioritylevelconfigurations",
                    "stopServingVersion": "1.29",
                    "clientStats": [],
                    "startServingReplacementVersion": "1.26"
                }
            ]
        },
        "id": "f6a11fe4-77f7-48c6-8326-9a13f022ecb3",
        "resources": [],
        "description": "Checks for usage of deprecated APIs that are scheduled for
 removal in Kubernetes v1.29. Upgrading your cluster before migrating to the updated
 APIs supported by v1.29 could cause application impact."
    }
}
```

Actualización de una versión de Kubernetes de clúster de Amazon EKS

Cuando haya una nueva versión de Kubernetes disponible en Amazon EKS, puede actualizar el clúster de Amazon EKS a la versión más reciente.

> **Important**
>
> Una vez que actualice un clúster, no podrá cambiarlo a una versión anterior. Recomendamos que antes de actualizar a una nueva versión de Kubernetes revise la información en Versiones de Amazon EKS de Kubernetes y los pasos de actualización de este tema.

Las nuevas versiones de Kubernetes suelen presentar cambios significativos. Por ende, recomendamos que pruebe el comportamiento de las aplicaciones en la nueva versión de Kubernetes antes de realizar la actualización en los clústeres de producción. Para ello, puede crear un flujo de trabajo de integración continua con el fin de probar el comportamiento de la aplicación antes de pasar a una nueva versión de Kubernetes.

El proceso de actualización consiste en que Amazon EKS lance nodos de servidor de API nuevos con la versión actualizada de Kubernetes para sustituir a los existentes. Amazon EKS lleva a cabo una infraestructura estándar y una comprobación de estado de la disponibilidad del tráfico de red en estos nodos nuevos para verificar que funcionan según lo esperado. Sin embargo, una vez que haya iniciado la actualización del clúster, no podrá pausarla ni detenerla. Si cualquiera de estas comprobaciones falla, Amazon EKS revierte la implementación de la infraestructura y su clúster se mantiene en la versión anterior de Kubernetes. Las aplicaciones en ejecución no se ven afectadas y su clúster nunca queda en un estado no determinista o irrecuperable. Si fuese necesario, Amazon EKS realiza copias de seguridad de forma habitual a todos los clústeres administrados y existe un mecanismo de recuperación de clústeres. Evaluamos y mejoramos de forma constante nuestros procesos de administración de la infraestructura de Kubernetes.

Para actualizar el clúster, Amazon EKS requiere hasta cinco direcciones IP disponibles de las subredes que se especificaron cuando creó el clúster. Amazon EKS crea nuevas interfaces de red elástica de clúster (interfaces de red) en cualquiera de las subredes especificadas. Las interfaces de red se pueden crear en subredes diferentes a las que están las interfaces de red existentes, así que asegúrese de que las reglas del grupo de seguridad permitan la comunicación de clúster necesaria para cualquiera de las subredes que especificó al crear su clúster. Si alguna de las subredes especificadas al crear el clúster no existe, no tiene suficientes direcciones IP disponibles o no tiene reglas de grupo de seguridad que permitan la comunicación del clúster necesaria, la actualización puede tener errores.

> **Note**
>
> Para garantizar que el punto de conexión del servidor de API de su clúster esté siempre accesible, Amazon EKS ofrece un plano de control de Kubernetes y realiza actualizaciones sucesivas de las instancias del servidor API durante las operaciones de actualización. Para tener en cuenta los cambios en las direcciones IP de las instancias del servidor de API que admiten su punto de conexión del servidor API de Kubernetes, debe asegurarse de que los clientes de su servidor API gestionen las reconexiones de manera eficaz. Versiones recientes

de `kubectl` y las bibliotecas del cliente de Kubernetes que cuentan con soporte oficial, realizan este proceso de reconexión de forma transparente.

Actualice la versión de Kubernetes de un clúster de Amazon EKS

Para actualizar la versión de Kubernetes del clúster

1. Compare la versión de Kubernetes de su plano de control de clúster con la versión de Kubernetes de sus nodos.

   - Obtenga la versión de Kubernetes del plano de control de clúster.

     ```
     kubectl version
     ```

   - Obtenga la versión de Kubernetes de sus nodos. Este comando devuelve todos los nodos autoadministrados y administrados de Amazon EC2 y Fargate. Cada Pod de Fargate aparece como su propio nodo.

     ```
     kubectl get nodes
     ```

   Antes de actualizar un plano de control a una nueva versión de Kubernetes, asegúrese de que la versión secundaria de Kubernetes de ambos nodos administrados y de Fargate en el clúster debe ser la misma que la de la versión actual del plano de control. Por ejemplo, si el plano de control se ejecuta con la versión `1.29` y uno de los nodos con la versión `1.28`, debe actualizar los nodos a la versión `1.29` antes de actualizar el plano de control a la 1.30. También recomendamos que actualice los nodos autoadministrados a la misma versión que el plano de control antes de actualizar el plano de control. Para obtener más información, consulte Actualización de un grupo de nodos administrados y Actualizaciones de nodos autoadministrados. Si tiene nodos de Fargate con una versión secundaria inferior a la versión del plano de control, elimine primero el Pod que representa el nodo. Luego, actualice su plano de control. Los Pods restantes se actualizarán a la nueva versión después de volver a implementarlos.

2. Si la versión de Kubernetes con la que implementó originalmente el clúster era Kubernetes `1.25` o posterior, omita este paso.

De manera predeterminada, el controlador de admisión de la política de seguridad del Pod se encuentra habilitado en clústeres de Amazon EKS. Antes de actualizar el clúster, asegúrese de que las políticas de seguridad del Pod adecuadas estén implementadas. Esto ocurre para evitar posibles problemas de seguridad. Puede consultar la política predeterminada con el comando **`kubectl get psp eks.privileged`**.

```
kubectl get psp eks.privileged
```

Si recibe el siguiente error, consulte Política de seguridad predeterminada del Pod de Amazon EKS antes de continuar.

```
Error from server (NotFound): podsecuritypolicies.extensions "eks.privileged" not
 found
```

3. Si la versión de Kubernetes con la que implementó originalmente el clúster era Kubernetes `1.18` o posterior, omita este paso.

   Es posible que deba eliminar un término interrumpido de su manifiesto de CoreDNS.

   a. Verifique si su manifiesto CoreDNS cuenta con una línea que solo tiene la palabra `upstream`.

      ```
      kubectl get configmap coredns -n kube-system -o jsonpath='{$.data.Corefile}' |
       grep upstream
      ```

      Si no se devuelve un resultado, significa que el manifiesto no cuenta con la línea. En tal caso, continúe en el paso siguiente Si se devuelve la palabra `upstream`, elimine la línea.

   b. Elimine la línea que está cerca de la parte superior del archivo que solo tiene la palabra `upstream` en el archivo de configmap. No cambie nada más en el archivo. Después de eliminar la línea, guarde los cambios.

      ```
      kubectl edit configmap coredns -n kube-system -o yaml
      ```

4. Actualice el clúster mediante `eksctl`, la AWS Management Console o la AWS CLI.

Important

- Si va a actualizar a la versión `1.23` y usar volúmenes de Amazon EBS en el clúster, debe instalar el controlador CSI de Amazon EBS en el clúster antes de actualizar el clúster a la versión `1.23` para evitar interrupciones en la carga de trabajo. Para obtener más información, consulte Kubernetes 1.23 y Controlador CSI de Amazon EBS.
- Kubernetes `1.24` y versiones posteriores utilizan `containerd` como el tiempo de ejecución predeterminado del contenedor. Si va a cambiar al tiempo de ejecución de `containerd` y ya ha configurado Fluentd para Container Insights, debe migrar Fluentd a Fluent Bit antes de actualizar el clúster. Los analizadores de Fluentd están configurados para analizar únicamente los mensajes de registro en formato JSON. A diferencia de `dockerd`, el tiempo de ejecución del contenedor `containerd`contiene mensajes de registro que no están en formato JSON. Si no migra a Fluent Bit, algunos de los analizadores de Fluentd's configurados generarán una enorme cantidad de errores dentro del contenedor de Fluentd. por el número de versión compatible con Amazon EKS al que desea actualizar su clústerPara enviar registros a CloudWatch Logs, consulte Configurar Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs.
- Puesto que Amazon EKS ejecuta un plano de control de alta disponibilidad, puede actualizar solo una versión secundaria a la vez. Para obtener más información acerca de este requisito, consulte Política de compatibilidad de versiones y diferencia de versiones de Kubernetes. Supongamos que la versión del clúster actual es la `1.28` y quiere actualizarla a la `1.30`. Primero debe actualizar su clúster de versión `1.28` a la versión `1.29` y, a continuación, actualizar su clúster de versión `1.29` a la versión `1.30`.
- Revise la compatibilidad de versiones entre `kube-apiserver` de Kubernetes y `kubelet` en sus nodos.
  - A partir de la versión de Kubernetes `1.28`, en `kubelet` puede haber hasta tres versiones secundarias anteriores a `kube-apiserver`. Consulte Política de compatibilidad de escalado entre versiones de Kubernetes.
  - Si el `kubelet` de sus nodos administrados y de Fargate corresponde a la versión de Kubernetes `1.25` o una más reciente, puede actualizar su clúster hasta tres versiones más avanzadas sin necesidad de actualizar la versión de `kubelet`.

Por ejemplo, si `kubelet` está en la versión `1.25`, puede actualizar la versión del clúster de Amazon EKS de `1.25` a `1.26` a `1.27` y a `1.28`, mientras que `kubelet` permanezca en la versión `1.25`.

- Si el `kubelet` de sus nodos administrados y de Fargate está en la versión de Kubernetes `1.24` o anterior, es posible que solo haya hasta dos versiones secundarias anteriores a `kube-apiserver`. En otras palabras, si `kubelet` es versión `1.24` o anterior, solo puede actualizar el clúster hasta dos versiones más avanzadas. Por ejemplo, si `kubelet` está en la versión `1.21`, puede actualizar la versión del clúster de Amazon EKS de `1.21` a `1.22` y a `1.23`, pero no podrá actualizar el clúster a `1.24` mientras `kubelet` permanezca en `1.21`.

- Como práctica recomendada antes de iniciar una actualización, asegúrese de que el `kubelet` de sus nodos esté en la misma versión de Kubernetes que la de su plano de control.
- Si el clúster está configurado con una versión de Amazon VPC CNI plugin for Kubernetes anterior a `1.8.0`, le recomendamos actualizar el complemento a la versión más reciente antes de actualizar el clúster. Para actualizar el complemento, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.
- Si está actualizando el clúster a una versión `1.25` o posterior y ha implementado AWS Load Balancer Controller en el clúster, actualice el controlador a la versión `2.4.7` o posterior antes de actualizar la versión del clúster a `1.25`. Para obtener más información, consulte las notas de la versión Kubernetes 1.25.

eksctl

En este procedimiento, se requiere la versión `0.183.0` o posterior de `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar y actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

Actualice la versión de Kubernetes de su plano de control de Amazon EKS. Reemplace *my-cluster* por el nombre del clúster. Reemplace *1.30* por el número de versión compatible

con Amazon EKS al que desea actualizar su clúster. Para ver una lista de los números de versiones compatibles, consulte Versiones de Amazon EKS de Kubernetes.

```
eksctl upgrade cluster --name my-cluster --version 1.30 --approve
```

La actualización puede tardar varios minutos en completarse.

AWS Management Console

a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
b. Elija el nombre del clúster de Amazon EKS que desea actualizar y elija Actualizar versión del clúster.
c. En Versión de Kubernetes, seleccione la versión a la que desea actualizar el clúster y elija Actualizar.
d. En Nombre del clúster, escriba el nombre del clúster y seleccione Confirmar.

   La actualización puede tardar varios minutos en completarse.

AWS CLI

a. Actualice el clúster de Amazon EKS con el siguiente comando de la AWS CLI. Reemplace los *example values* por los de su propiedad. Reemplace *1.30* por el número de versión compatible con Amazon EKS al que desea actualizar su clúster. Para ver una lista de los números de versiones compatibles, consulte Versiones de Amazon EKS de Kubernetes.

   ```
   aws eks update-cluster-version --region region-code --name my-cluster --
   kubernetes-version 1.30
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   {
       "update": {
           "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
           "status": "InProgress",
           "type": "VersionUpdate",
           "params": [
               {
                   "type": "Version",
                   "value": "1.30"
   ```

```
            },
            {
                "type": "PlatformVersion",
                "value": "eks.1"
            }
        ],
[...]
        "errors": []
    }
}
```

b. Monitoree el estado de la actualización del clúster con el siguiente comando. Utilice el nombre del clúster e ID de actualización devueltos por el comando anterior. Cuando se muestra el estado `Successful`, la actualización se ha completado. La actualización puede tardar varios minutos en completarse.

```
aws eks describe-update --region region-code --name my-cluster --update-
id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

Un ejemplo de salida sería el siguiente.

```
{
    "update": {
        "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
        "status": "Successful",
        "type": "VersionUpdate",
        "params": [
            {
                "type": "Version",
                "value": "1.30"
            },
            {
                "type": "PlatformVersion",
                "value": "eks.1"
            }
        ],
[...]
        "errors": []
    }
}
```

5. Una vez que se complete la actualización del clúster, actualice los nodos a la misma versión secundaria de Kubernetes de su clúster actualizado. Para obtener más información, consulte Actualizaciones de nodos autoadministrados y Actualización de un grupo de nodos administrados. Los Pods nuevos que se lancen en Fargate tienen una versión de `kubelet` que coinciden con la versión del clúster. Los Pods de Fargate existentes no cambian.
6. (Opcional) Si implementó el Cluster Autoscaler de Kubernetes en el clúster antes de actualizar este último, actualice dicho Cluster Autoscaler a la versión más reciente que coincida con la versión principal y secundaria de Kubernetes a las que actualizó.

   a. Abra la página de versiones del escalador automático del clúster en un navegador web y busque la versión más reciente del escalador automático del clúster que coincida con la versión principal y secundaria de Kubernetes de su clúster. Por ejemplo, si la versión de Kubernetes del clúster es `1.30`, busque la última versión del escalador automático del clúster que comience por `1.30`. Registre el número de versión semántica (`1.30.n`, por ejemplo) de esa versión para usarlo en el siguiente paso.

   b. Establezca la etiqueta de la imagen del escalador automático del clúster en la versión que ha registrado en el paso anterior con el siguiente comando. Si es necesario, reemplace *`1.30.n`* por su propio valor.

   ```
   kubectl -n kube-system set image deployment.apps/cluster-autoscaler cluster-autoscaler=registry.k8s.io/autoscaling/cluster-autoscaler:v1.30.n
   ```

7. (Solo para clústeres con nodos de GPU) Si el clúster tiene grupos de nodos compatibles con GPU (por ejemplo, `p3.2xlarge`), debe actualizar el complemento del dispositivo NVIDIA para Kubernetes DaemonSet de su clúster. Reemplace *`vX.X.X`* con la versión Plugin de dispositivo NVidia/K8S deseada antes de ejecutar el siguiente comando.

   ```
   kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
   ```

8. Actualice los complementos Amazon VPC CNI plugin for Kubernetes, CoreDNS y `kube-proxy`. Recomendamos actualizar los complementos a las versiones mínimas que figuran en los tokens de las cuentas de servicio.

   - Si está usando complementos de Amazon EKS, seleccione Clústeres en la consola de Amazon EKS y, a continuación, seleccione el nombre del clúster que actualizó en el panel de navegación izquierdo. Las notificaciones aparecen en la consola. Le informan que hay una versión nueva disponible para cada complemento que tenga una actualización disponible.

Para actualizar un complemento, seleccione la pestaña Complementos. En uno de los cuadros de un complemento que tenga una actualización disponible, seleccione Actualizar ahora, seleccione una versión disponible y, a continuación, seleccione Actualizar.

- Como alternativa, puede utilizar la AWS CLI o `eksctl` para actualizar los complementos. Para obtener más información, consulte Actualización de un complemento.

9. De ser necesario, actualice su versión de `kubectl`. Debe utilizar una versión de `kubectl` con una diferencia de versión de menos de un número que el plano del control del clúster de Amazon EKS. Por ejemplo, un cliente de `kubectl` `1.29` trabaja con los clústeres Kubernetes, `1.28`, `1.29` y `1.30`. Puede comprobar su versión instalada actualmente con el siguiente comando.

```
kubectl version --client
```

Eliminación de un clúster de Amazon EKS

Cuando termine de utilizar un clúster de Amazon EKS, debe eliminar los recursos asociados para no incurrir en costos innecesarios.

Para eliminar un clúster conectado, consulte Anulación del registro de un clúster.

Important

- Si tiene en el clúster servicios activos asociados a un equilibrador de carga, deberá eliminar los servicios antes de eliminar el clúster para que los equilibradores de carga se eliminen correctamente. De lo contrario, pueden quedar recursos huérfanos en la VPC que le impidan eliminarla.
- Si recibe un error porque se ha eliminado el creador del clúster, consulte este artículo para resolver el problema.
- Los recursos de Amazon Managed Service para Prometheus están fuera del ciclo de vida del clúster y deben mantenerse por fuera del clúster. Al eliminar el clúster, asegúrese de eliminar, también, cualquier raspador para reducir los costes aplicables. Para más información, consulte Búsqueda y eliminación de rapsadores en la Guía de usuario de Amazon Managed Service para Prometheus.

Puede eliminar un clúster mediante `eksctl`, la AWS Management Console o la AWS CLI.

eksctl

Para eliminar un clúster de Amazon EKS y los nodos con **`eksctl`**

En este procedimiento, se requiere la versión `eksctl` o posterior de la `0.183.0`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

1. Enumere todos los servicios que se ejecutan en el clúster.

   ```
   kubectl get svc --all-namespaces
   ```

2. Elimine los servicios que tengan asociado un valor `EXTERNAL-IP`. Estos servicios se presentan por medio de un equilibrador de carga de Elastic Load Balancing y debe eliminarlos en Kubernetes para que el equilibrador de carga y los recursos asociados se lancen correctamente.

   ```
   kubectl delete svc service-name
   ```

3. Elimine el clúster y sus nodos asociados con el siguiente comando, al reemplazar *`prod`* por el nombre de su clúster.

   ```
   eksctl delete cluster --name prod
   ```

   Salida:

   ```
   [#]  using region region-code
   [#]  deleting EKS cluster "prod"
   [#]  will delete stack "eksctl-prod-nodegroup-standard-nodes"
   [#]  waiting for stack "eksctl-prod-nodegroup-standard-nodes" to get deleted
   [#]  will delete stack "eksctl-prod-cluster"
   [#]  the following EKS cluster resource(s) for "prod" will be deleted: cluster.
    If in doubt, check CloudFormation console
   ```

AWS Management Console

Cómo eliminar un clúster de Amazon EKS con la AWS Management Console

1. Enumere todos los servicios que se ejecutan en el clúster.

```
kubectl get svc --all-namespaces
```

2. Elimine los servicios que tengan asociado un valor `EXTERNAL-IP`. Estos servicios se presentan por medio de un equilibrador de carga de Elastic Load Balancing y debe eliminarlos en Kubernetes para que el equilibrador de carga y los recursos asociados se lancen correctamente.

```
kubectl delete svc service-name
```

3. Elimine todos los grupos de nodos y perfiles de Fargate.

   a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

   b. En el panel de navegación izquierdo, seleccione Clústeres de Amazon EKS y, a continuación, en la lista de clústeres con pestañas, seleccione el nombre del clúster que desea eliminar.

   c. Elija la pestaña Compute (Informática) y elija un grupo de nodos para eliminar. Elija Delete (Eliminar), introduzca el nombre del grupo de nodos y, a continuación, elija Delete (Eliminar). Elimine todos los grupos de nodos del clúster.

   > **Note**
   >
   > Los grupos de nodos enumerados solo son los grupos de nodos administrados.

   d. Seleccione un Fargate Profile (Perfil de Fargate) para eliminar, seleccione Delete (Eliminar), ingrese el nombre del perfil y, a continuación, seleccione Delete (Eliminar). Elimine todos los perfiles de Fargate en el clúster.

4. Elimine todas las pilas de AWS CloudFormation de nodos autoadministrados.

   a. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.

   b. Seleccione la pila de nodos que desea eliminar y, luego, elija Delete (Eliminar).

c. En el cuadro de diálogo de confirmación Delete stack (Eliminar pila), elija Delete stack (Eliminar pila). Elimine todas las pilas de nodos autoadministradas del clúster.

5. Eliminar el clúster.

   a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

   b. Seleccione el clúster que desea eliminar y elija Delete (Eliminar).

   c. En la pantalla de confirmación de eliminación del clúster, elija Delete (Eliminar).

6. (Opcional) Elimine la pila de AWS CloudFormation de la VPC.

   a. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.

   b. Seleccione la pila de VPC que desea eliminar y, luego, elija Delete (Eliminar).

   c. En el cuadro de diálogo de confirmación Eliminar pila, elija Eliminar pila.

AWS CLI

Cómo eliminar un clúster de Amazon EKS con la AWS CLI

1. Enumere todos los servicios que se ejecutan en el clúster.

```
kubectl get svc --all-namespaces
```

2. Elimine los servicios que tengan asociado un valor `EXTERNAL-IP`. Estos servicios se presentan por medio de un equilibrador de carga de Elastic Load Balancing y debe eliminarlos en Kubernetes para que el equilibrador de carga y los recursos asociados se lancen correctamente.

```
kubectl delete svc service-name
```

3. Elimine todos los grupos de nodos y perfiles de Fargate.

   a. Enumere los grupos de nodos del clúster con el siguiente comando.

```
aws eks list-nodegroups --cluster-name my-cluster
```

> **Note**
>
> Los grupos de nodos enumerados son solo los grupos de nodos administrados.

b. Elimine cada grupo de nodos con el siguiente comando. Elimine todos los grupos de nodos del clúster.

```
aws eks delete-nodegroup --nodegroup-name my-nodegroup --cluster-name my-cluster
```

c. Enumere los perfiles de Fargate del clúster con el siguiente comando.

```
aws eks list-fargate-profiles --cluster-name my-cluster
```

d. Elimine cada perfil de Fargate con el siguiente comando. Elimine todos los perfiles de Fargate en el clúster.

```
aws eks delete-fargate-profile --fargate-profile-name my-fargate-profile --cluster-name my-cluster
```

4. Elimine todas las pilas de AWS CloudFormation de nodos autoadministrados.

a. Muestre las pilas de AWS CloudFormation disponibles con el siguiente comando. Busque el nombre de la plantilla de nodos en la salida resultante.

```
aws cloudformation list-stacks --query "StackSummaries[].StackName"
```

b. Elimine cada pila de nodos con el siguiente comando y reemplace *node-stack* por el nombre de su pila de nodos. Elimine todas las pilas de nodos autoadministradas del clúster.

```
aws cloudformation delete-stack --stack-name node-stack
```

5. Elimine el clúster con el siguiente comando, sustituyendo *my-cluster* por el nombre de su clúster.

```
aws eks delete-cluster --name my-cluster
```

6. (Opcional) Elimine la pila de AWS CloudFormation de la VPC.

a. Muestre las pilas de AWS CloudFormation disponibles con el siguiente comando. Busque el nombre de la plantilla de VPC en la salida resultante.

```
aws cloudformation list-stacks --query "StackSummaries[].StackName"
```

b. Elimine la pila de VPC con el siguiente comando, sustituyendo *my-vpc-stack* por el nombre de la pila de VPC.

```
aws cloudformation delete-stack --stack-name my-vpc-stack
```

Control de acceso al punto de conexión del clúster de Amazon EKS

Esto lo ayudará a habilitar el acceso privado al punto de conexión del servidor de API de Kubernetes de su clúster de Amazon EKS y a limitar, o a desactivar por completo, el acceso público desde Internet.

Cuando se crea un clúster nuevo, Amazon EKS crea un punto de conexión para el servidor de la API de Kubernetes administrado que utiliza a fin de comunicarse con su clúster (mediante herramientas de administración de Kubernetes como, por ejemplo, `kubectl`). De forma predeterminada, este punto de conexión del servidor de la API es público en Internet y el acceso al servidor de la API está protegido mediante una combinación de AWS Identity and Access Management (IAM) y el Control de acceso basado en rol (RBAC) nativo de Kubernetes.

Puede habilitar el acceso privado al servidor de la API de Kubernetes para que toda la comunicación entre los nodos y el servidor de la API permanezcan dentro de su VPC. Puede limitar las direcciones IP que pueden acceder a su servidor de API desde Internet o desactivar por completo el acceso a Internet al servidor de API.

> **Note**
>
> Dado que este punto de conexión es para el servidor de la API de Kubernetes y no un punto de conexión de AWS PrivateLink tradicional que sirve para comunicarse con una API de AWS, no aparece como un punto de conexión en la consola de Amazon VPC.

Al habilitar el acceso privado al punto de conexión para el clúster, Amazon EKS crea una zona alojada privada de Route 53 en su nombre y la asocia a la VPC de su clúster. Esta zona alojada

privada se administra mediante Amazon EKS y no aparece en los recursos de Route 53 de su cuenta. Para que la zona privada alojada dirija el tráfico adecuadamente a su servidor de API, su VPC debe tener `enableDnsHostnames` y `enableDnsSupport` establecidos en `true` y las opciones de DHCP establecidas para su VPC deben incluir `AmazonProvidedDNS` en su lista de servidores de nombres de dominios. A fin de obtener más información, consulte Actualización de soporte de DNS para su VPC en la guía del usuario de Amazon VPC.

Puede definir los requisitos de acceso al punto de conexión del servidor de la API al crear un nuevo clúster; puede actualizar el acceso al punto de conexión del servidor de la API para un clúster en cualquier momento.

Modificar el acceso al punto de conexión del clúster

Utilice los procedimientos de esta sección para modificar el acceso al punto de conexión para un clúster existente. En la siguiente tabla se muestran las combinaciones de acceso al punto de conexión del servidor de la API y sus comportamientos asociados.

Opciones de acceso al punto de conexión del servidor de la API

Acceso público al punto de conexión	Acceso privado al punto de conexión	Comportamiento
Habilitado	Deshabilitad	• Este es el comportamiento predeterminado para los clústeres de Amazon EKS nuevos. • Las solicitudes de la API de Kubernetes que provienen de dentro de la VPC de su clúster (como comunicación desde el nodo al plano de control) dejan la VPC, pero no la red de Amazon. • Se puede acceder al servidor de la API del clúster desde Internet. De forma opcional, puede limitar los bloques de

Acceso público al punto de conexión	Acceso privado al punto de conexión	Comportamiento
		CIDR que pueden acceder al punto de conexión público. Si limita el acceso a bloques de CIDR específicos, se recomiend a habilitar también el punto de conexión privado o asegurarse de que los bloques de CIDR que especifique incluyan las direcciones desde las que los nodos y los Pods de Fargate (si los utiliza) acceden al punto de conexión público.
Habilitado	Habilitado	• Las solicitudes de la API de Kubernetes de dentro de la VPC de su clúster (como comunicación desde el nodo al plano de control) utilizan el punto de conexión privado de VPC. • Se puede acceder al servidor de la API del clúster desde Internet. De forma opcional, puede limitar los bloques de CIDR que pueden acceder al punto de conexión público.

Acceso público al punto de conexión	Acceso privado al punto de conexión	Comportamiento
Deshabilitad	Habilitado	• Todo el tráfico al servidor de la API del clúster debe proceder de dentro de la VPC de su clúster o de una red conectada. • No hay acceso público al servidor de la API desde Internet. Cualquier comando `kubectl` debe provenir de dentro de la VPC o de una red conectada. Para ver las opciones de conectividad, consulte Acceso a un servidor de API solo privado. • Los servidores DNS públicos resuelven el punto de conexión del servidor de API del clúster en una dirección IP privada desde la VPC. En el pasado, el punto de conexión se podía resolver desde dentro de la VPC. Si el punto de conexión no se resuelve en una dirección IP privada dentro de la VPC para un clúster existente, puede: • Habilitar el acceso público y, a continuación, volver a deshabilitarlo. Solo tiene

Acceso público al punto de conexión	Acceso privado al punto de conexión	Comportamiento
		que hacerlo una vez para un clúster y el punto de conexión se resolverá en una dirección IP privada a partir de ese momento. • Actualice el clúster.

Puede modificar el acceso al punto de conexión del servidor de API del clúster mediante la AWS Management Console o la AWS CLI.

AWS Management Console

Para modificar el acceso al punto de conexión del servidor de la API del clúster mediante la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster para mostrar la información del clúster.
3. En la pestaña Networking (Redes), elija Update (Actualizar).
4. Para el acceso privado, decida si desea habilitar o deshabilitar el acceso privado al punto de conexión del servidor de API de Kubernetes del clúster. Si habilita el acceso privado, las solicitudes de la API de Kubernetes que provengan de dentro de la VPC del clúster utilizan el punto de conexión de VPC privada. Debe habilitar el acceso privado para deshabilitar el acceso público.
5. Para el acceso público, decida si desea habilitar o deshabilitar el acceso público al punto de conexión del servidor de API de Kubernetes del clúster. Si deshabilita el acceso público, el servidor de la API de Kubernetes del clúster solo puede recibir solicitudes que provengan de dentro de la VPC del clúster.
6. (Opcional) Si ha habilitado el acceso público, puede especificar qué direcciones de Internet pueden comunicarse con el punto de conexión público. Seleccione Advanced Settings (Configuración avanzada). Introduzca un bloque CIDR, como `203.0.113.5/32`. El bloque no puede incluir direcciones reservadas. Puede introducir bloques adicionales seleccionando Add Source (Agregar origen). Hay un número máximo de bloques de CIDR que puede especificar. Para obtener más información, consulte Cuotas de servicio de Amazon EKS.

Si no especifica ningún bloque, el punto de conexión del servidor de API público recibe solicitudes de todas las direcciones IP (`0.0.0.0/0`). Si restringe el acceso a su punto de conexión público mediante bloques de CIDR, se recomienda habilitar también el acceso al punto de conexión privado para que los nodos y los Pods de Fargate (si los utiliza) puedan comunicarse con el clúster. Si el punto de conexión privado no está habilitado, los orígenes de CIDR del punto de conexión de acceso público deben incluir los orígenes de salida de la VPC. Por ejemplo, si tiene un nodo en una subred privada que se comunica con Internet a través de una puerta de enlace NAT, deberá agregar la dirección IP saliente de la puerta de enlace NAT como parte de un bloque de CIDR permitido en su punto de conexión público.

7. Elija Update (Actualizar) para finalizar.

AWS CLI

Para modificar el acceso al punto de conexión del servidor de la API del clúster mediante la AWS CLI

Complete los siguientes pasos con la versión `1.27.160` o posterior de la AWS CLI. Puede comprobar su versión actual con `aws --version`. Para realizar la instalación o actualización de la AWS CLI, consulte Instalación de la AWS CLI.

1. Actualice el acceso al punto de conexión del servidor de la API del clúster con el siguiente comando de la AWS CLI Sustituya el nombre de su clúster y los valores de acceso de punto de conexión deseados. Si configura el `endpointPublicAccess=true`, podrá introducir un solo bloque de CIDR o una lista separada por comas de bloques de CIDR para `publicAccessCidrs`. Los bloques no pueden incluir direcciones reservadas. Si especifica bloques de CIDR, el punto de conexión del servidor de API público solo recibirá solicitudes de los bloques enumerados. Hay un número máximo de bloques de CIDR que puede especificar. Para obtener más información, consulte Cuotas de servicio de Amazon EKS. Si restringe el acceso a su punto de conexión público mediante bloques de CIDR, se recomienda habilitar también el acceso al punto de conexión privado para que los nodos y los Pods de Fargate (si los utiliza) puedan comunicarse con el clúster. Si el punto de conexión privado no está habilitado, los orígenes de CIDR del punto de conexión de acceso público deben incluir los orígenes de salida de la VPC. Por ejemplo, si tiene un nodo en una subred privada que se comunica con Internet a través de una puerta de enlace NAT, deberá agregar la dirección IP saliente de la puerta de enlace NAT como parte de un bloque de CIDR permitido en su punto de conexión público. Si no especifica ningún bloque de CIDR, el

punto de conexión del servidor de API público recibe solicitudes de todas las direcciones IP (0.0.0.0/0).

> **Note**
>
> El siguiente comando habilita el acceso privado y público desde una única dirección IP al punto de conexión del servidor de API. Reemplace *203.0.113.5/32* por un único bloque de CIDR o una lista separada por comas de bloques de CIDR a los que desea restringir el acceso a la red.

```
aws eks update-cluster-config \
    --region region-code \
    --name my-cluster \
    --resources-vpc-config
 endpointPublicAccess=true,publicAccessCidrs="203.0.113.5/32",endpointPrivateAccess=true
```

Un ejemplo de salida sería el siguiente.

```
{
    "update": {
        "id": "e6f0905f-a5d4-4a2a-8c49-EXAMPLE00000",
        "status": "InProgress",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "true"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            },
            {
                "type": "publicAccessCidrs",
                "value": "[\203.0.113.5/32\"]"
            }
        ],
        "createdAt": 1576874258.137,
        "errors": []
    }
```

```
}
```

2. Monitoree el estado de la actualización del acceso al punto de conexión con el siguiente comando, utilizando el nombre del clúster y el ID de actualización devueltos por el comando anterior. Su actualización se habrá completado cuando el estado mostrado sea `Successful`.

```
aws eks describe-update \
    --region region-code \
    --name my-cluster \
    --update-id e6f0905f-a5d4-4a2a-8c49-EXAMPLE00000
```

Un ejemplo de salida sería el siguiente.

```
{
    "update": {
        "id": "e6f0905f-a5d4-4a2a-8c49-EXAMPLE00000",
        "status": "Successful",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "true"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            },
            {
                "type": "publicAccessCidrs",
                "value": "[\203.0.113.5/32\"]"
            }
        ],
        "createdAt": 1576874258.137,
        "errors": []
    }
}
```

Acceso a un servidor de API solo privado

Si ha deshabilitado el acceso público al punto de conexión del servidor de la API de Kubernetes del clúster, solo puede obtener acceso al servidor de API desde dentro de la VPC o desde una red conectada. Hay varias formas de obtener acceso al punto de conexión del servidor de API de Kubernetes:

Red conectada

Conecte su red a la VPC con una puerta de enlace de tránsito de AWS u otra opción de conectividad y, a continuación, utilice un equipo en la red conectada. Debe asegurarse de que el grupo de seguridad del plano de control de Amazon EKS tiene reglas para permitir el tráfico de entrada en el puerto 443 desde la red conectada.

Host bastión de Amazon EC2

Puede lanzar una instancia de Amazon EC2 en una subred pública de la VPC del clúster y, a continuación, iniciar sesión mediante SSH en esa instancia para ejecutar comandos de `kubectl`. Para obtener más información, consulte hosts bastión de Linux en AWS. Debe asegurarse de que el grupo de seguridad del plano de control de Amazon EKS tiene reglas para permitir el tráfico de entrada en el puerto 443 desde su host bastión. Para obtener más información, consulte Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS.

Cuando configure `kubectl` para el host bastión, asegúrese de utilizar las credenciales de AWS que ya están asignadas a su configuración de RBAC del clúster o agregue la entidad principal de IAM que utilizará el bastión a la configuración de RBAC antes de eliminar el acceso público al punto de conexión. Para obtener más información, consulte the section called “Concesión de acceso a las API de Kubernetes” y Acceso denegado o no autorizado (`kubectl`).

IDE AWS Cloud9

AWS Cloud9 es un entorno de desarrollo integrado (IDE) basado en la nube que permite escribir, ejecutar y depurar su código con solo un navegador. Puede crear un IDE de AWS Cloud9 en la VPC de su clúster y utilizar el IDE para comunicarse con el clúster. Para obtener más información, consulte Creación de un entorno en AWS Cloud9. Debe asegurarse de que el grupo de seguridad del plano de control de Amazon EKS tiene reglas para permitir el tráfico de entrada en el puerto 443 desde el grupo de seguridad de IDE. Para obtener más información, consulte Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS.

Cuando configure `kubectl` para el IDE de AWS Cloud9, asegúrese de utilizar las credenciales de AWS que ya están asignadas a su configuración de RBAC del clúster o agregue la entidad

principal de IAM que utilizará el IDE a la configuración de RBAC antes de eliminar el acceso público al punto de conexión. Para obtener más información, consulte Concesión de acceso a las API de Kubernetes y Acceso denegado o no autorizado (`kubectl`).

Habilitación del cifrado de secretos en un clúster existente

Si habilita el cifrado de secretos, los secretos de Kubernetes se cifran con la clave de AWS KMS key que seleccione. La clave KMS debe cumplir las siguientes condiciones:

- Simétrica
- Debe poder cifrar y descifrar datos
- Debe estar creada en la misma Región de AWS que el clúster
- Si la clave de KMS se creó en una cuenta diferente, la entidad principal de IAM debe tener acceso a la clave de KMS.

Para obtener más información, consulte Permitir que las entidades principales de IAM de otras cuentas utilicen una clave de KMS en la Guía para desarrolladores de AWS Key Management Service.

> **Warning**
>
> No puede desactivar el cifrado de secretos después de habilitarlo. Esta acción es irreversible.

eksctl

Puede habilitar el cifrado de dos formas:

- Agregue cifrado a su clúster con un solo comando.

  Para volver a cifrar los secretos de forma automática, ejecute el siguiente comando.

  ```
  eksctl utils enable-secrets-encryption \
      --cluster my-cluster \
      --key-arn arn:aws:kms:region-code:account:key/key
  ```

  Para optar por no volver a cifrar los secretos de forma automática, ejecute el siguiente comando.

```
eksctl utils enable-secrets-encryption
    --cluster my-cluster \
    --key-arn arn:aws:kms:region-code:account:key/key \
    --encrypt-existing-secrets=false
```

- Agregue cifrado al clúster con un archivo `kms-cluster.yaml`.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code

secretsEncryption:
  keyARN: arn:aws:kms:region-code:account:key/key
```

Para que los secretos se vuelvan a cifrar automáticamente, ejecute el siguiente comando.

```
eksctl utils enable-secrets-encryption -f kms-cluster.yaml
```

Para optar por no volver a cifrar los secretos de forma automática, ejecute el siguiente comando.

```
eksctl utils enable-secrets-encryption -f kms-cluster.yaml --encrypt-existing-
secrets=false
```

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el clúster al que desea agregar el cifrado de KMS.
3. Elija la pestaña Overview (Resumen) (está seleccionada de manera predeterminada).
4. Desplácese hasta la sección Secrets encryption (Cifrado de secretos) y elija Enable (Habilitar).
5. Seleccione una clave en el menú desplegable y elija el botón Enable (Habilitar). Si no aparece ninguna clave, primero debe crear una. Para obtener más información, consulte Creación de claves.

6. Elija el botón Confirm (Confirmar) para utilizar la clave elegida.

AWS CLI

1. Asocie la configuración del cifrado de secretos con el clúster mediante el siguiente comando de la AWS CLI. Reemplace los *example values* por los de su propiedad.

```
aws eks associate-encryption-config \
    --cluster-name my-cluster \
    --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

Un ejemplo de salida sería el siguiente.

```
{
  "update": {
    "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
    "status": "InProgress",
    "type": "AssociateEncryptionConfig",
    "params": [
      {
        "type": "EncryptionConfig",
        "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
      }
    ],
    "createdAt": 1613754188.734,
    "errors": []
  }
}
```

2. Puede monitorear el estado de la actualización de cifrado con el siguiente comando. Utilice la especificación `cluster name` y `update ID` que se devolvió en la salida anterior. Cuando se muestra el estado `Successful`, la actualización se ha completado.

```
aws eks describe-update \
    --region region-code \
    --name my-cluster \
    --update-id 3141b835-8103-423a-8e68-12c2521ffa4d
```

Un ejemplo de salida sería el siguiente.

```
{
  "update": {
    "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
    "status": "Successful",
    "type": "AssociateEncryptionConfig",
    "params": [
      {
        "type": "EncryptionConfig",
        "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\"arn:aws:kms:region-code:account:key/key\"}}]"
      }
    ],
    "createdAt": 1613754188.734>,
    "errors": []
  }
}
```

3. Para verificar que el cifrado se encuentra habilitado en el clúster, ejecute el comando `describe-cluster`. La respuesta contiene una cadena de `EncryptionConfig`.

```
aws eks describe-cluster --region region-code --name my-cluster
```

Después de habilitar el cifrado en el clúster, deberá cifrar todos los secretos existentes con la clave nueva:

> **Note**
>
> Si usa `eksctl`, solo es necesario ejecutar el siguiente comando si opta por no volver a cifrar sus secretos automáticamente.

```
kubectl get secrets --all-namespaces -o json | kubectl annotate --overwrite -f - kms-
encryption-timestamp="time value"
```

> **Warning**
>
> Si habilita el cifrado de secretos para un clúster existente y alguna vez la clave de KMS que utiliza se elimina, no hay una ruta de recuperación para el clúster. Si elimina la clave de KMS, coloca el clúster permanentemente en un estado degradado. Para obtener más información, consulte Eliminación de claves de AWS KMS.

> **Note**
>
> De forma predeterminada, el comando `create-key` crea una clave KMS de cifrado simétrico con una política de clave que da al administrador raíz de la cuenta acceso en acciones y recursos de AWS KMS. Si desea reducir los permisos, asegúrese de que las acciones `kms:DescribeKey` y `kms:CreateGrant` estén permitidas en la política para la entidad principal que llama a la API `create-cluster`.
> Para clústeres que utilizan cifrado de sobres KMS, se requieren permisos `kms:CreateGrant`. La condición `kms:GrantIsForAWSResource` no es compatible con la acción CreateCluster y no debe utilizarse en las políticas de KMS para controlar permisos `kms:CreateGrant` para los usuarios que realizan CreateCluster.

Activación de la compatibilidad con Windows para su clúster de Amazon EKS

Antes de implementar nodos de Windows, tenga en cuenta lo siguiente.

Consideraciones

- Puede utilizar redes de host en nodos de Windows mediante Pods `HostProcess`. Para obtener más información, consulte Crear un `HostProcess`Pod de Windows en la documentación de Kubernetes.
- Los clústeres de Amazon EKS deben contener uno o varios nodos de Linux o Fargate para ejecutar Pods del sistema principal que solo se ejecutan en Linux, como CoreDNS.
- Los registros de eventos `kubelet` y `kube-proxy` se redirigen al registro de eventos de Windows de EKS y se establecen en un límite de 200 MB.
- No puede utilizar Grupos de seguridad de Pods con Pods en ejecución en nodos de Windows.

- No puede utilizar redes personalizadas con Windows.
- No puede usar`IPv6` con los nodos de Windows.
- Los nodos de Windows admiten una interfaz de red elástica por nodo. De forma predeterminada, la cantidad de Pods que puede ejecutar por nodo de Windows es igual a la cantidad de direcciones IP disponibles por interfaz de red elástica para el tipo de instancia del nodo, menos uno. Para obtener más información, consulte Direcciones IP por interfaz de red por tipo de instancia en la Guía del usuario de Amazon EC2.
- En un clúster de Amazon EKS, un único servicio con un equilibrador de carga puede admitir hasta 1024 Pods de backend. Cada Pod tiene su propia dirección IP única. El límite anterior de 64 Pods dejará de aplicarse después de una actualización de Windows Server a partir de la compilación del SO 17763.2746.
- No se admiten contenedores de Windows para Pods de Amazon EKS en Fargate.
- No puede recuperar registros del pod de `vpc-resource-controller`. Anteriormente podía hacerlo cuando se implementaba el controlador en el plano de datos.
- Hay un periodo de enfriamiento antes de que se asigne una dirección `IPv4` a un nuevo pod. Esto evita que el tráfico fluya hacia un pod anterior con la misma dirección `IPv4` debido a reglas caducadas de `kube-proxy`.
- El origen del controlador se administra en GitHub. Para registrar problemas con el controlador u ofrecer ayuda con estos, visite el proyecto en GitHub.
- Al especificar un ID de AMI personalizado para los grupos de nodos administrados de Windows, agregue `eks:kube-proxy-windows` al mapa de configuración de AWS IAM Authenticator. Para obtener más información, consulte Límites y condiciones al especificar un ID de AMI.

Requisitos previos

- Un clúster existente. El clúster debe estar ejecutando una de las versiones de Kubernetes y versiones de la plataforma que se enumeran en la siguiente tabla. Se admite cualquier versión de Kubernetes y plataformas posteriores a las enumeradas en la tabla. Si la versión de su clúster o plataforma es anterior a una de las siguientes versiones, debe habilitar la compatibilidad con Windows heredado en el plano de datos del clúster. Una vez que el clúster se encuentre en una de las siguientes versiones de Kubernetes y la plataforma, o posteriores, puede eliminar la compatibilidad con Windows heredado y habilitar la compatibilidad con Windows en el plano de control.

Versión de Kubernetes	Versión de la plataforma
1.30	eks.2
1.29	eks.1
1.28	eks.1
1.27	eks.1
1.26	eks.1
1.25	eks.1
1.24	eks.2

- El clúster debe tener al menos un nodo de Linux o Pod de Fargate (recomendamos al menos dos) para ejecutar CoreDNS. Si habilita la compatibilidad con Windows heredado, debe utilizar un nodo de Linux (no puede usar un Pod de Fargate) para ejecutar CoreDNS.
- Un Rol de IAM del clúster de Amazon EKS existente.

Habilitación de la compatibilidad con Windows

Si el clúster no se encuentra en una de las versiones de Kubernetes y la plataforma enumeradas en los Requisitos previos o es posterior a ellas, debe habilitar en su lugar la compatibilidad con Windows heredado. Para obtener más información, consulte Habilitación de la compatibilidad con Windows heredado.

Si nunca ha habilitado la compatibilidad con Windows en el clúster, vaya al siguiente paso.

Si ha habilitado la compatibilidad con Windows en un clúster anterior a una de las versiones de Kubernetes o la plataforma que aparecen en los Requisitos previos, debe primero eliminar el `vpc-resource-controller` y el `vpc-admission-webhook` del plano de datos. Estos están obsoletos y ya no se necesitan.

Para habilitar la compatibilidad con Windows en su clúster

1. Si no tiene nodos de Amazon Linux en su clúster y utiliza grupos de seguridad para Pods, avance al siguiente paso. De lo contrario, confirme que la política administrada

`AmazonEKSVPCResourceController` esté adjunta al rol del clúster. Reemplace *`eksClusterRole`* por el nombre de rol del clúster.

```
aws iam list-attached-role-policies --role-name eksClusterRole
```

Un ejemplo de salida sería el siguiente.

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEKSClusterPolicy",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
        },
        {
            "PolicyName": "AmazonEKSVPCResourceController",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEKSVPCResourceController"
        }
    ]
}
```

Si la política está adjunta, como en la salida anterior, omita el siguiente paso.

2. Adjunte la política administrada AmazonEKSVPCResourceController al Rol de IAM del clúster de Amazon EKS. Reemplace *`eksClusterRole`* por el nombre de rol del clúster.

```
aws iam attach-role-policy \
  --role-name eksClusterRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSVPCResourceController
```

3. Cree un archivo denominado *`vpc-resource-controller-configmap.yaml`* con el siguiente contenido.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: amazon-vpc-cni
  namespace: kube-system
data:
  enable-windows-ipam: "true"
```

4. Aplique el `ConfigMap` en su clúster.

```
kubectl apply -f vpc-resource-controller-configmap.yaml
```

5. Verifique que el `ConfigMap` de `aws-auth` contenga una asignación para que el rol de instancia del nodo Windows incluya el grupo de permisos RBAC de `eks:kube-proxy-windows`. Puede verificar ejecutando el siguiente comando:

   ```
   kubectl get configmap aws-auth -n kube-system -o yaml
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: aws-auth
     namespace: kube-system
   data:
     mapRoles: |
       - groups:
         - system:bootstrappers
         - system:nodes
         - eks:kube-proxy-windows # This group is required for Windows DNS resolution
    to work
         rolearn: arn:aws:iam::111122223333:role/eksNodeRole
         username: system:node:{{EC2PrivateDNSName}}
   [...]
   ```

   `eks:kube-proxy-windows` debería aparecer en la lista de grupos. Si el grupo no está especificado, debe actualizar `ConfigMap` o crearlo para incluir el grupo requerido. Para obtener más información sobre `ConfigMap` de `aws-auth`, consulte Aplique el `ConfigMap` de `aws-auth` en su clúster.

Eliminación de la compatibilidad con Windows heredado del plano de datos

Si ha habilitado la compatibilidad con Windows en un clúster anterior a una de las versiones de Kubernetes o la plataforma que aparecen en los Requisitos previos, debe primero eliminar el `vpc-resource-controller` y el `vpc-admission-webhook` del plano de datos. Estos están obsoletos y ya no se necesitan porque la funcionalidad que proporcionaban ahora está habilitada en el plano de control.

1. Desinstale el `vpc-resource-controller` con el siguiente comando. Utilice este comando independientemente de la herramienta con la que lo instaló inicialmente. Reemplace *region-code* (solo la instancia de ese texto después de `/manifests/`) por la Región de AWS en la que se encuentra su clúster.

   ```
   kubectl delete -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-resource-controller/latest/vpc-resource-controller.yaml
   ```

2. Desinstale el `vpc-admission-webhook` siguiendo las instrucciones de la herramienta con la que lo instaló.

   eksctl

   Ejecute los siguientes comandos.

   ```
   kubectl delete deployment -n kube-system vpc-admission-webhook
   kubectl delete service -n kube-system vpc-admission-webhook
   kubectl delete mutatingwebhookconfigurations.admissionregistration.k8s.io vpc-admission-webhook-cfg
   ```

   kubectl on macOS or Windows

   Ejecute el siguiente comando de la . Reemplace *region-code* (solo la instancia de ese texto después de `/manifests/`) por la Región de AWS en la que se encuentra su clúster.

   ```
   kubectl delete -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/vpc-admission-webhook-deployment.yaml
   ```

3. Habilite la compatibilidad con Windows para el clúster en el plano de control.

Deshabilitación de la compatibilidad de Windows

Para deshabilitar la compatibilidad con Windows en el clúster

1. Si el clúster contiene nodos de Amazon Linux y utiliza grupos de seguridad para Pods, omita este paso.

Elimine la política administrada de IAM `AmazonVPCResourceController` de su rol de clúster. Reemplace *`eksClusterRole`* por el nombre del rol de su clúster y *`111122223333`* por el ID de la cuenta.

```
aws iam detach-role-policy \
    --role-name eksClusterRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonEKSVPCResourceController
```

2. Deshabilite IPAM de Windows en el ConfigMap de `amazon-vpc-cni`.

```
kubectl patch configmap/amazon-vpc-cni \
                    -n kube-system \
                    --type merge \
                    -p '{"data":{"enable-windows-ipam":"false"}}'
```

Implementación de pods

Cuando implementa pods en el clúster, debe especificar el sistema operativo que utilizan si ejecuta una combinación de tipos de nodos.

Para Pods de Linux, use el siguiente texto del selector de nodos en sus manifiestos.

```
nodeSelector:
        kubernetes.io/os: linux
        kubernetes.io/arch: amd64
```

Para Pods de Windows, use el siguiente texto del selector de nodos en sus manifiestos.

```
nodeSelector:
        kubernetes.io/os: windows
        kubernetes.io/arch: amd64
```

Puede implementar una aplicación de muestra para ver los selectores de nodos en uso.

Habilitación de la compatibilidad con Windows heredado

Si el clúster se encuentra en una de las versiones de Kubernetes y la plataforma enumeradas en los Requisitos previos o es posterior a ellas, recomendamos que habilite en su lugar la compatibilidad

de Windows en el plano de control. Para obtener más información, consulte Habilitación de la compatibilidad con Windows.

Los siguientes pasos lo ayudan a habilitar la compatibilidad de Windows heredado para el plano de datos de su clúster de Amazon EKS si la versión del clúster o la plataforma es anterior a las versiones enumeradas en los Requisitos previos. Una vez que su versión del clúster y la plataforma están en una versión indicada en los Requisitos previos o en una posterior, recomendamos que elimine la compatibilidad con Windows heredado y la habilite en el plano de control.

Puede utilizar `eksctl`, o un cliente de Windows, macOS o Linux, a fin de habilitar la compatibilidad con Windows heredado para el clúster.

eksctl

Para habilitar la compatibilidad con Windows heredado en su clúster con **`eksctl`**

Requisito previo

En este procedimiento, se requiere la versión `0.183.0` de `eksctl` o posterior. Puede verificar la versión con el siguiente comando.

```
eksctl version
```

Para obtener más información sobre la instalación o la actualización de `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

1. Habilite la compatibilidad con Windows para su clúster de Amazon EKS con el siguiente comando de `eksctl`. Reemplace *`my-cluster`* por el nombre del clúster. Este comando implementa el controlador de recursos de VPC y el webhook de controlador de admisión de VPC que son necesarios en los clústeres de Amazon EKS para ejecutar cargas de trabajo de Windows.

   ```
   eksctl utils install-vpc-controllers --cluster my-cluster --approve
   ```

   > **Important**
   >
   > El webhook del controlador de admisión de VPC está firmado con un certificado que vence un año después de la fecha de emisión. Para evitar el tiempo de inactividad, asegúrese de renovar el certificado antes de que venza. Para obtener

más información, consulte Renovación del certificado de webhook de admisión de VPC.

2. Una vez que haya habilitado la compatibilidad con Windows, puede lanzar un grupo de nodos de Windows en su clúster. Para obtener más información, consulte Lanzamiento de nodos de Windows autoadministrados.

Windows

Para habilitar la compatibilidad con Windows heredado en su clúster con un cliente de Windows

En los siguientes pasos, reemplace *region-code* por la Región de AWS en la que reside el clúster.

1. Implemente el controlador de recursos de VPC en su clúster.

```
kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-resource-controller/latest/vpc-resource-controller.yaml
```

2. Implemente el webhook de controlador de admisión de VPC en su clúster.

   a. Descargue los archivos de implementación y los scripts necesarios.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/vpc-admission-webhook-deployment.yaml;
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/Setup-VPCAdmissionWebhook.ps1;
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-create-signed-cert.ps1;
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-patch-ca-bundle.ps1;
```

   b. Instale OpenSSL y jq.

   c. Configure e implemente el webhook de admisión de VPC.

```
./Setup-VPCAdmissionWebhook.ps1 -DeploymentTemplate ".\vpc-admission-webhook-deployment.yaml"
```

> **Important**
>
> El webhook del controlador de admisión de VPC está firmado con un certificado que vence un año después de la fecha de emisión. Para evitar el tiempo de inactividad, asegúrese de renovar el certificado antes de que venza. Para obtener más información, consulte Renovación del certificado de webhook de admisión de VPC.

3. Determine si su clúster tiene el enlace de rol de clúster necesario.

```
kubectl get clusterrolebinding eks:kube-proxy-windows
```

Si se devuelve una salida similar a la siguiente salida de ejemplo, el clúster tendrá el enlace de rol necesario.

```
NAME                       AGE
eks:kube-proxy-windows     10d
```

Si la salida incluye `Error from server (NotFound)`, el clúster no tiene el enlace de rol de clúster necesario. Agregue el enlace creando un archivo llamado *`eks-kube-proxy-windows-crb.yaml`* con el siguiente contenido.

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: eks:kube-proxy-windows
  labels:
    k8s-app: kube-proxy
    eks.amazonaws.com/component: kube-proxy
subjects:
  - kind: Group
    name: "eks:kube-proxy-windows"
roleRef:
  kind: ClusterRole
  name: system:node-proxier
  apiGroup: rbac.authorization.k8s.io
```

Aplique la configuración al clúster.

```
kubectl apply -f eks-kube-proxy-windows-crb.yaml
```

4. Una vez que haya habilitado la compatibilidad con Windows, puede lanzar un grupo de nodos de Windows en su clúster. Para obtener más información, consulte Lanzamiento de nodos de Windows autoadministrados.

macOS and Linux

Para habilitar la compatibilidad con Windows heredado en su clúster con un cliente de macOS o Linux.

Este procedimiento requiere que la biblioteca `openssl` y el procesador JSON `jq` estén instalados en su sistema cliente.

En los siguientes pasos, reemplace *`region-code`* por la Región de AWS en la que reside el clúster.

1. Implemente el controlador de recursos de VPC en su clúster.

```
kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-resource-controller/latest/vpc-resource-controller.yaml
```

2. Cree el manifiesto del webhook de controlador de admisión de VPC para su clúster.

   a. Descargue los archivos de implementación y los scripts necesarios.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-create-signed-cert.sh
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-patch-ca-bundle.sh
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/vpc-admission-webhook-deployment.yaml
```

   b. Agregue permisos a los scripts del intérprete de comandos para que puedan ejecutarse.

```
chmod +x webhook-create-signed-cert.sh webhook-patch-ca-bundle.sh
```

   c. Cree un secreto para una comunicación segura.

```
./webhook-create-signed-cert.sh
```

d. Verifique el secreto.

```
kubectl get secret -n kube-system vpc-admission-webhook-certs
```

e. Configure el webhook y cree un archivo de implementación.

```
cat ./vpc-admission-webhook-deployment.yaml | ./webhook-patch-ca-bundle.sh >
 vpc-admission-webhook.yaml
```

3. Implemente el webhook de admisión de VPC.

```
kubectl apply -f vpc-admission-webhook.yaml
```

> **Important**
>
> El webhook del controlador de admisión de VPC está firmado con un certificado que vence un año después de la fecha de emisión. Para evitar el tiempo de inactividad, asegúrese de renovar el certificado antes de que venza. Para obtener más información, consulte Renovación del certificado de webhook de admisión de VPC.

4. Determine si su clúster tiene el enlace de rol de clúster necesario.

```
kubectl get clusterrolebinding eks:kube-proxy-windows
```

Si se devuelve una salida similar a la siguiente salida de ejemplo, el clúster tendrá el enlace de rol necesario.

```
NAME                     ROLE                                AGE
eks:kube-proxy-windows   ClusterRole/system:node-proxier   19h
```

Si la salida incluye `Error from server (NotFound)`, el clúster no tiene el enlace de rol de clúster necesario. Agregue el enlace creando un archivo llamado *`eks-kube-proxy-windows-crb.yaml`* con el siguiente contenido.

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
```

```
  name: eks:kube-proxy-windows
  labels:
    k8s-app: kube-proxy
    eks.amazonaws.com/component: kube-proxy
subjects:
  - kind: Group
    name: "eks:kube-proxy-windows"
roleRef:
  kind: ClusterRole
  name: system:node-proxier
  apiGroup: rbac.authorization.k8s.io
```

Aplique la configuración al clúster.

```
kubectl apply -f eks-kube-proxy-windows-crb.yaml
```

5. Una vez que haya habilitado la compatibilidad con Windows, puede lanzar un grupo de nodos de Windows en su clúster. Para obtener más información, consulte Lanzamiento de nodos de Windows autoadministrados.

Renovación del certificado de webhook de admisión de VPC

El certificado que utiliza el webhook de admisión de VPC vence un año después de su emisión. Para evitar el tiempo de inactividad, es importante que renueve el certificado antes de que venza. Puede verificar la fecha de vencimiento de su certificado actual con el siguiente comando.

```
kubectl get secret \
    -n kube-system \
    vpc-admission-webhook-certs -o json | \
    jq -r '.data."cert.pem"' | \
    base64 -decode | \
    openssl x509 \
    -noout \
    -enddate | \
    cut -d= -f2
```

Un ejemplo de salida sería el siguiente.

```
May 28 14:23:00 2022 GMT
```

Puede renovar el certificado con `eksctl` o un equipo con Windows o Linux/macOS. Siga las instrucciones de la herramienta que utilizó inicialmente para instalar el webhook de admisión de VPC. Por ejemplo, si instaló inicialmente el webhook de admisión de VPC con `eksctl`, debe renovar el certificado con las instrucciones de la pestaña `eksctl`.

eksctl

1. Vuelva a instalar el certificado. Reemplace *my-cluster* por el nombre del clúster.

```
eksctl utils install-vpc-controllers -cluster my-cluster -approve
```

2. Verifique que recibe el siguiente resultado.

```
2021/05/28 05:24:59 [INFO] generate received request
2021/05/28 05:24:59 [INFO] received CSR
2021/05/28 05:24:59 [INFO] generating key: rsa-2048
2021/05/28 05:24:59 [INFO] encoded CSR
```

3. Reinicie la implementación del webhook.

```
kubectl rollout restart deployment -n kube-system vpc-admission-webhook
```

4. Si el certificado que renovó ha vencido y tiene Pods de Windows estancados en el estado `Container creating`, debe eliminar y volver a implementar esos Pods.

Windows

1. Obtenga el script para generar un certificado nuevo.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-create-signed-cert.ps1;
```

2. Prepare el parámetro para el script.

```
./webhook-create-signed-cert.ps1 -ServiceName vpc-admission-webhook-svc -SecretName vpc-admission-webhook-certs -Namespace kube-system
```

3. Reinicie la implementación del webhook.

```
kubectl rollout restart deployment -n kube-system vpc-admission-webhook-deployment
```

4. Si el certificado que renovó se venció y tiene Pods de Windows estancados en el estado `Container creating`, debe eliminar y volver a implementar esos Pods.

Linux and macOS

Requisito previo

Debe tener OpenSSL y `jq` instalados en su equipo.

1. Obtenga el script para generar un certificado nuevo.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/manifests/region-code/vpc-admission-webhook/latest/webhook-create-signed-cert.sh
```

2. Cambie los permisos.

```
chmod +x webhook-create-signed-cert.sh
```

3. Ejecute el script.

```
./webhook-create-signed-cert.sh
```

4. Restablezca el webhook.

```
kubectl rollout restart deployment -n kube-system vpc-admission-webhook-deployment
```

5. Si el certificado que renovó se venció y tiene Pods de Windows estancados en el estado `Container creating`, debe eliminar y volver a implementar esos Pods.

Admite una mayor densidad de Pod en los nodos de Windows

En Amazon EKS, a cada Pod se le asigna una dirección de `IPv4` desde su VPC. Debido a esto, la cantidad de Pods que puede implementar en un nodo está limitada por las direcciones IP disponibles, incluso si hay recursos suficientes para ejecutar más Pods en el nodo. Dado que un nodo de Windows solo admite una interfaz de red elástica, de forma predeterminada, la cantidad máxima de direcciones IP disponibles en un nodo de Windows es igual a:

```
Number of private IPv4 addresses for each interface on the node - 1
```

Se utiliza una dirección IP como dirección IP principal de la interfaz de red, por lo que no se puede asignar a Pods.

Puede habilitar una mayor densidad de Pod en los nodos de Windows si habilita la delegación de prefijos IP. Esta característica le permite asignar un prefijo `/28` `IPv4` a la interfaz de red principal, en lugar de asignar direcciones `IPv4` secundarias. La asignación de un prefijo IP aumenta el número máximo de direcciones `IPv4` disponibles en el nodo a:

```
(Number of private IPv4 addresses assigned to the interface attached to the node - 1) *
 16
```

Con esta cantidad significativamente mayor de direcciones IP disponibles, las direcciones IP disponibles no deberían limitar su capacidad para escalar la cantidad de Pods en sus nodos. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.

Requisitos del clúster privado

En este tema, se describe cómo implementar un clúster de Amazon EKS que se implementa en la Nube de AWS sin acceso a Internet saliente. Si tiene un clúster local en AWS Outposts, consulte Lanzamiento de nodos autoadministrados de Amazon Linux en un Outpost en lugar de este tema.

Si no está familiarizado con las redes de Amazon EKS, consulte Desmitificación de las redes de clústeres para nodos de trabajo de Amazon EKS. Si su clúster no tiene acceso a Internet saliente, debe cumplir con los siguientes requisitos:

- El clúster debe extraer imágenes de un registro de contenedores que esté en su VPC. Puede crear un Amazon Elastic Container Registry en su VPC y copiar las imágenes del contenedor para que sus nodos puedan extraerlas. Para obtener más información, consulte Copiar una imagen de contenedor de un repositorio en otro repositorio.
- Su clúster debe tener habilitado el acceso privado a los puntos de conexión. Esto es necesario para que los nodos se registren en el punto de conexión del clúster. El acceso público del punto de conexión es opcional. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS.
- Los nodos autoadministrados de Linux y Windows deben incluir los siguientes argumentos de arranque antes de lanzarlos. Estos argumentos omiten la introspección de Amazon EKS y no requieren acceso a la API de Amazon EKS desde dentro de la VPC.

1. Determine el valor del punto de conexión del clúster con el siguiente comando. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-cluster --name my-cluster --query cluster.endpoint --output text
```

   Un ejemplo de salida sería el siguiente.

```
https://EXAMPLE108C897D9B2F1B21D5EXAMPLE.sk1.region-code.eks.amazonaws.com
```

2. Determine el valor de la autoridad de certificación del clúster con el siguiente comando. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-cluster --name my-cluster --query cluster.certificateAuthority --output text
```

   El resultado devuelto es una cadena larga.

3. Reemplace *cluster-endpoint* y *certificate-authority* en los siguientes comandos por los valores obtenidos en el resultado de los comandos anteriores. Para obtener más información acerca de cómo especificar argumentos de arranque al lanzar nodos autoadministrados, consulte Lanzar nodos autoadministrados de Amazon Linux y Lanzamiento de nodos de Windows autoadministrados.

- Para nodos de Linux:

```
--apiserver-endpoint cluster-endpoint --b64-cluster-ca certificate-authority
```

  Para obtener argumentos adicionales, consulte el script de arranque en GitHub.

- Para nodos de Windows:

> **Note**
>
> Si utiliza un CIDR de servicio personalizado, debe especificarlo con el parámetro `-ServiceCIDR`. De lo contrario, se producirá un error en la resolución de DNS del clúster para Pods.

```
-APIServerEndpoint cluster-endpoint -Base64ClusterCA certificate-authority
```

Para obtener argumentos adicionales, consulte Parámetros de configuración del script de arranque.

- El `aws-auth` `ConfigMap` del clúster debe crearse desde su VPC. Para obtener más información sobre cómo crear y añadir entradas al `ConfigMap` de `aws-auth`, ingrese **`eksctl create iamidentitymapping --help`** en su terminal. Si el `ConfigMap` no existe en su servidor, `eksctl` lo creará cuando utilice el comando para añadir una asignación de identidad.
- Los Pods configurados con roles de IAM para cuentas de servicio adquieren credenciales de una llamada a la API de AWS Security Token Service (AWS STS). Si no hay acceso a Internet saliente, debe crear y utilizar un punto de conexión de VPC de AWS STS en su VPC. La mayoría de los SDK de AWS `v1` utilizan el punto de conexión global de AWS STS de forma predeterminada (`sts.amazonaws.com`), que no utiliza el punto de conexión de VPC de AWS STS. Para utilizar el punto de conexión de VPC de AWS STS, es posible que tenga que configurar el SDK de modo que utilice el punto de conexión de AWS STS regional (`sts.`*`region-code`*`.amazonaws.com`). Para obtener más información, consulte Configure el punto de conexión AWS Security Token Service de una cuenta de servicio.

- Las subredes de VPC de su clúster deben tener un punto de conexión de interfaz de VPC para todas las subredes de VPC de los Servicios de AWS a los que los Pods necesiten acceder. Para obtener más información, consulte Acceso a un servicio de AWS a través de un punto de conexión de VPC de interfaz. Algunos servicios y puntos de conexión de uso común se enumeran en la siguiente tabla. Para obtener la lista completa de los puntos de conexión, consulte Servicios de AWS que se integran con AWS PrivateLink en la Guía de AWS PrivateLink.

Servicio	Punto de conexión
Amazon EC2	com.amazonaws.*region-code* .ec2
Amazon Elastic Container Registry (para extraer imágenes de contenedores)	com.amazonaws.*region-code* .ecr.api, com.amazonaws.*region-code* .ecr.dkr y com.amazonaws.*region-code* .s3
Application Load Balancers y Network Load Balancers	com.amazonaws.*region-code* .elasticloadbalancing
AWS X-Ray	com.amazonaws.*region-code* .xray

Servicio	Punto de conexión
Registros de Amazon CloudWatch	com.amazonaws.*region-code* .logs
AWS Security Token Service (requerido al usar roles de IAM para cuentas de servicio)	com.amazonaws.*region-code* .sts

Consideraciones

- Todos los nodos autoadministrados deben implementarse en subredes que tengan los puntos de conexión de la interfaz de VPC que necesita. Si crea un grupo de nodos administrados, el grupo de seguridad del punto de conexión de la interfaz de VPC debe permitir el CIDR para las subredes. También puede agregar el grupo de seguridad del nodo creado al grupo de seguridad del punto de conexión de la interfaz de VPC.
- Si sus Pods utilizan volúmenes de Amazon EFS, antes de implementar Controlador CSI de Amazon EFS, se debe cambiar el archivo kustomization.yaml del controlador para establecer las imágenes de contenedor de manera que utilicen la misma Región de AWS que el clúster de Amazon EKS.
- Puede utilizar AWS Load Balancer Controller para implementar los equilibradores de carga de aplicación (ALB) y los equilibradores de carga de red de AWS en su clúster privado. Al implementarlo, debe usar los indicadores de línea de comandos para establecer `enable-shield`, `enable-waf` y `enable-wafv2` como falsos. No se admite la detección de certificados con nombres de host de los objetos de entrada. Esto se debe a que el controlador necesita llegar a AWS Certificate Manager, el cual no tiene un punto de conexión de la interfaz de VPC.

  El controlador es compatible con equilibradores de carga de red con destinos IP, que son necesarios para su uso con Fargate. Para obtener más información, consulte Equilibrio de carga de aplicaciones en Amazon EKS y Crear un equilibrador de carga de red.
- Se admite el escalador automático de clústeres. Al implementar Pods de Cluster Autoscaler de clúster, asegúrese de que la línea de comandos incluya `--aws-use-static-instance-list=true`. Para obtener más información, consulte Uso de lista de instancias estáticas en GitHub. La VPC del nodo de trabajo también debe incluir el punto de conexión de VPC de AWS STS y el punto de conexión de VPC de escalado automático.

- Algunos productos de software de contenedores utilizan llamadas de API que acceden a AWS Marketplace Metering Service para supervisar el uso. Los clústeres privados no permiten estas llamadas, por lo que estos tipos de contenedores no se pueden utilizar en clústeres privados.

Versiones de Amazon EKS de Kubernetes

Kubernetes evoluciona rápidamente con nuevas características, actualizaciones de diseño y correcciones de errores. La comunidad publica nuevas versiones secundarias de Kubernetes (tales como `1.30`) en promedio una vez cada cuatro meses. Amazon EKS sigue el ciclo de publicación y obsolescencia de las versiones anteriores para las versiones secundarias. Cuando haya nuevas versiones de Kubernetes disponibles en Amazon EKS, le recomendamos que actualice proactivamente los clústeres para que utilicen la versión más reciente disponible.

Una versión secundaria dispone de soporte estándar de Amazon EKS durante los primeros 14 meses después de su publicación. Cuando una versión supera la fecha de finalización del soporte estándar, pasa automáticamente a recibir soporte extendido durante los 12 meses siguientes. El soporte extendido permite permanecer en una versión específica de Kubernetes durante más tiempo a cambio de un costo adicional por hora de clúster. Si no ha actualizado su clúster antes de que concluya el período de soporte extendido, el clúster se actualiza automáticamente a la versión extendida más antigua que tenga soporte actualmente.

Se recomienda crear el clúster con la última versión de Kubernetes disponible compatible con Amazon EKS. Si su aplicación requiere una versión específica de Kubernetes, puede seleccionar versiones anteriores. Puede crear nuevos clústeres de Amazon EKS en cualquier versión para la que se ofrezca soporte estándar o extendido.

Versiones disponibles con soporte estándar

Las siguientes versiones de Kubernetes están disponibles actualmente con soporte estándar de Amazon EKS:

- `1.30`
- `1.29`
- `1.28`
- `1.27`
- `1.26`

Para ver cambios importantes que debe conocer sobre cada versión con soporte estándar, consulte Notas de la versión para las versiones con soporte estándar.

Versiones disponibles con soporte extendido

Las siguientes versiones de Kubernetes están disponibles actualmente con soporte extendido de Amazon EKS:

- `1.25`
- `1.24`
- `1.23`

Para ver cambios importantes que debe conocer sobre cada versión con soporte extendido, consulte Notas de la versión para las versiones con soporte extendido.

Las siguientes versiones de Kubernetes están disponibles actualmente en el soporte extendido de Amazon EKS, con el requisito adicional de que no pueden crearse nuevos clústeres con estas versiones:

- `1.22`
- `1.21`

  Para obtener información acerca de las versiones, consulte Notas de la versión para las versiones 1.21 y 1.22.

Calendario de lanzamientos de Amazon EKS de Kubernetes

En la siguiente tabla aparecen las fechas importantes de publicación y soporte que deben tenerse en cuenta para cada versión de Kubernetes.

> **Note**
>
> Las fechas con solo un mes y un año son aproximadas y se actualizan con una fecha exacta cuando se conoce.

Versión de Kubernetes	Versión anterior	Versión de Amazon EKS	Fecha de finalización del soporte estándar	Fecha de finalización del soporte extendido
1.30	17 de abril de 2024	23 de mayo de 2024	23 de julio de 2025	23 de julio de 2026
1.29	13 de diciembre de 2023	23 de enero de 2024	23 de marzo de 2025	23 de marzo de 2026
1.28	15 de agosto de 2023	26 de septiembre de 2023	26 de noviembre de 2024	26 de noviembre de 2025
1.27	11 de abril de 2023	24 de mayo de 2023	24 de julio de 2024	24 de julio de 2025
1.26	9 de diciembre de 2022	11 de abril de 2023	11 de junio de 2024	11 de junio de 2025
1.25	23 de agosto de 2022	22 de febrero de 2023	1 de mayo de 2024	1 de mayo de 2025
1.24	3 de mayo de 2022	15 de noviembre de 2022	31 de enero de 2024	31 de enero de 2025
1.23	7 de diciembre de 2021	11 de agosto de 2022	11 de octubre de 2023	11 de octubre de 2024
1.22	4 de agosto de 2021	4 de abril de 2022	4 de junio de 2023	1 de septiembre de 2024
1.21	8 de abril de 2021	19 de julio de 2021	16 de febrero de 2023	15 de julio de 2024

Preguntas frecuentes sobre las versiones de Amazon EKS

¿Cuántas versiones de Kubernetes con soporte estándar hay disponibles?

En línea con el soporte que ofrece la comunidad de Kubernetes para las versiones de Kubernetes, Amazon EKS se compromete a proporcionar soporte estándar como mínimo a cuatro versiones de Kubernetes listas para producción en todo momento. Se anunciará la fecha de finalización del soporte estándar de una determinada versión secundaria de Kubernetes con una antelación mínima de 60 días. Debido al proceso de cualificación y publicación de Amazon EKS para nuevas versiones de Kubernetes, la fecha de finalización del soporte estándar de una versión de Kubernetes en Amazon EKS será la misma o posterior a aquella en que el proyecto Kubernetes deje de proporcionar soporte a la versión anterior.

¿Durante cuánto tiempo recibe soporte estándar una versión de Kubernetes por parte de Amazon EKS?

Una versión de Kubernetes recibe soporte estándar durante 14 meses después de encontrarse disponible por primera vez en Amazon EKS. Esto es cierto incluso si la versión anterior de Kubernetes ya no admite una versión disponible en Amazon EKS. Creamos parches de seguridad que se pueden aplicar a las versiones de Kubernetes compatibles con Amazon EKS.

¿Se me avisa cuando va a concluir el soporte estándar para una versión de Kubernetes en Amazon EKS?

Sí. Si alguno de los clústeres de su cuenta ejecuta una versión que está cerca del final del soporte, Amazon EKS envía un aviso a través de AWS Health Dashboard aproximadamente 12 meses después del lanzamiento de la versión de Kubernetes en Amazon EKS. El aviso incluye la fecha de finalización del soporte. Será como mínimo 60 días a partir de la fecha del aviso.

¿Qué características de Kubernetes son compatibles con Amazon EKS?

Amazon EKS admite todas las características disponibles con carácter general de la API de Kubernetes. A partir de la versión `1.24` de Kubernetes, las nuevas API beta no están habilitadas en los clústeres de forma predeterminada. Sin embargo, las API beta ya existentes y las nuevas versiones de las API beta existentes siguen habilitadas de forma predeterminada. Las características alfa no son compatibles.

¿Los grupos de nodos administrados de Amazon EKS se actualizan automáticamente junto con la versión del plano de control del clúster?

No. Un grupo de nodos administrados crea instancias de Amazon EC2 en su cuenta. Estas instancias no se actualizan de forma automática cuando usted o Amazon EKS actualizan su

plano de control. Para obtener más información, consulte Actualización de un grupo de nodos administrados. Recomendamos mantener la misma versión de Kubernetes en el plano de control y los nodos.

¿Los grupos de nodos autoadministrados se actualizan automáticamente junto con la versión del plano de control del clúster?

No. Un grupo de nodos autoadministrados incluye instancias de Amazon EC2 en su cuenta. Estas instancias no se actualizan de forma automática cuando usted o Amazon EKS actualizan la versión del plano de control en su nombre. Un grupo de nodos autoadministrados no tiene indicaciones en la consola de que necesita actualizarse. Puede ver la versión de `kubelet` instalada en un nodo al seleccionar el nodo en la lista de Nodos en la pestaña Overview (Información general) del clúster para determinar qué nodos deben actualizarse. Debe actualizar los nodos de forma manual. Para obtener más información, consulte Actualizaciones de nodos autoadministrados.

El proyecto Kubernetes comprueba la compatibilidad entre el plano de control y los nodos para un máximo de tres versiones secundarias. Por ejemplo, los nodos `1.27` continúan funcionando cuando se organicen mediante un plano de control `1.30`. No obstante, no se recomienda ejecutar un clúster con nodos que estén tres versiones secundarias por detrás del plano de control de forma constante. Para obtener más información, consulte la sección sobre la política de compatibilidad de versiones y diferencia de versiones de Kubernetes en la documentación de Kubernetes. Recomendamos mantener la misma versión de Kubernetes en el plano de control y los nodos.

¿Los Pods que se ejecutan en Fargate se actualizan automáticamente con una actualización automática de la versión del plano de control del clúster?

No. Se recomienda encarecidamente ejecutar los Pods de Fargate como parte de un controlador de replicación, tal como una implementación de Kubernetes. A continuación, realice un reinicio continuo de todos los Pods de Fargate. La versión nueva del Pod de Fargate se implementa con una versión de `kubelet` que es la misma que la versión actualizada del plano de control de clúster. Para obtener más información, consulte Implementaciones en la documentación de Kubernetes.

> **Important**
>
> Si actualiza el plano de control, aún debe actualizar los nodos de Fargate por su cuenta. Para actualizar los nodos de Fargate, elimine el Pod de Fargate representado por el

nodo y vuelva a implementar ese Pod. El Pod nuevo se implementa con una versión de `kubelet` que es la misma versión del clúster.

Preguntas frecuentes sobre el soporte extendido de Amazon EKS

Los términos “soporte estándar” y “soporte extendido” son nuevos para mí. ¿Qué significan esos términos?

El soporte estándar de una versión de Kubernetes en Amazon EKS comienza cuando se publica una versión de Kubernetes en Amazon EKS, y concluirá 14 meses después de la fecha de publicación. El soporte extendido de una versión de Kubernetes comenzará inmediatamente después de que finalice el soporte estándar, y concluirá al cabo de 12 meses a partir de ese momento. Por ejemplo, el soporte estándar de la versión `1.23` en Amazon EKS concluye el 11 de octubre de 2023. El soporte extendido de la versión `1.23` comenzó el 12 de octubre de 2023 y concluirá el 11 de octubre de 2024.

¿Qué debo hacer para conseguir soporte extendido para los clústeres de Amazon EKS?

No tiene que hacer nada para conseguir soporte extendido para sus clústeres de Amazon EKS. El soporte estándar comenzará cuando se publique una versión de Kubernetes en Amazon EKS, y concluirá 14 meses después de la fecha de publicación. El soporte extendido de una versión de Kubernetes comenzará inmediatamente después de que finalice el soporte estándar, y concluirá al cabo de 12 meses a partir de ese momento. Los clústeres que se ejecuten en una versión de Kubernetes que haya superado la fecha de finalización del soporte estándar se incorporarán automáticamente al soporte extendido.

¿Para qué versiones de Kubernetes se puede obtener soporte extendido?

El soporte extendido está disponible para las versiones de Kubernetes `1.23` y superiores. Puede ejecutar clústeres en cualquier versión durante un máximo de 12 meses después de que concluya el soporte estándar para esa versión. Esto significa que cada versión recibirá soporte durante 26 meses en Amazon EKS (14 meses de soporte estándar más 12 meses de soporte extendido).

¿Qué sucede si no quiero usar el soporte extendido?

Si no desea recibir automáticamente soporte extendido, puede actualizar su clúster a una versión de Kubernetes que tenga soporte estándar de Amazon EKS. Los clústeres que no se actualicen a una versión de Kubernetes con soporte estándar pasarán automáticamente a recibir soporte extendido.

¿Qué sucederá cuando terminen los 12 meses de soporte extendido?

Los clústeres que se ejecuten en una versión de Kubernetes que haya completado su ciclo de vida de 26 meses (14 meses de soporte estándar más 12 meses de soporte extendido) se actualizarán automáticamente a la siguiente versión.

Cuando llegue la fecha de finalización del soporte extendido, ya no podrá crear nuevos clústeres de Amazon EKS con la versión no soportada. Amazon EKS actualiza los planos de control existentes a la primera versión admitida de forma automática mediante un proceso de implementación gradual tras la fecha de finalización del soporte. Después de la actualización automática del plano de control, asegúrese de actualizar los complementos del clúster y los nodos de Amazon EC2 de forma manual. Para obtener más información, consulte Actualice la versión de Kubernetes de un clúster de Amazon EKS.

¿Cuándo se actualiza exactamente mi plano de control de manera automática después de la fecha de finalización del soporte extendido?

Amazon EKS no puede facilitar plazos concretos. Las actualizaciones automáticas pueden producirse en cualquier momento después de la fecha de finalización del soporte extendido. No recibirá ninguna notificación antes de la actualización. Recomendamos que actualice de manera proactiva su plano de control sin depender del proceso de actualización automática de Amazon EKS. Para obtener más información, consulte Actualización de una versión de Kubernetes de clúster de Amazon EKS.

¿Puedo mantener mi plano de control en una versión de Kubernetes de manera indefinida?

No. La seguridad en la nube de AWS es la mayor prioridad. Pasado cierto punto (normalmente un año), la comunidad de Kubernetes deja de publicar parches de exposiciones y vulnerabilidades comunes (CVE) y desalienta el envío de CVE para versiones obsoletas. Esto significa que es posible que ni siquiera se denuncien las vulnerabilidades específicas de una versión anterior de Kubernetes. Esto deja expuestos los clústeres sin aviso y sin opciones de corrección en caso de vulnerabilidad. Debido a esto, Amazon EKS no permite que los planos de control permanezcan en una versión que haya llegado al final del soporte extendido.

¿El soporte extendido conlleva un costo adicional?

Sí, los clústeres de Amazon EKS que se ejecuten con soporte extendido conllevan un costo adicional. Para obtener más información sobre los precios, consulte el soporte ampliado de Amazon EKS para conocer los precios de las versiones de Kubernetes en el blog AWS.

¿Qué incluye el soporte extendido?

Los clústeres de Amazon EKS con soporte extendido reciben revisiones de seguridad continuas para el plano de control de Kubernetes. Además, Amazon EKS publicará revisiones para el CNI de Amazon VPC, `kube-proxy`, y complementos de CoreDNS para las versiones con soporte extendido. Amazon EKS también lanzará revisiones para las AMI optimizadas de Amazon EKS publicadas por AWS para Amazon Linux, Bottlerocket y Windows, además de nodos de Fargate de Amazon EKS para esas versiones. Todos los clústeres con soporte extendido seguirán teniendo acceso a soporte técnico de AWS.

> **Note**
>
> El soporte extendido para las AMI de Windows optimizadas para Amazon EKS que sean publicadas por AWS no está disponible para la versión `1.23` de Kubernetes; sin embargo, no está disponible para la versión `1.24` o una posterior de Kubernetes.

¿Existen limitaciones en cuanto a revisiones para componentes ajenos a Kubernetes en el soporte extendido?

Si bien el soporte extendido cubre todos los componentes específicos de Kubernetes de AWS, solo brindará soporte a las AMI optimizadas de Amazon EKS publicadas por AWS para Amazon Linux, Bottlerocket y Windows en todo momento. Esto significa que, mientras utilice el soporte extendido, es posible que tenga componentes más recientes (tales como sistema operativo o kernel) en una AMI optimizada de Amazon EKS. Por ejemplo, cuando Amazon Linux 2 llegue al final de su ciclo de vida en 2025, las AMI de Amazon Linux optimizadas de Amazon EKS se crearán con un sistema operativo Amazon Linux más reciente. Amazon EKS anunciará y documentará discrepancias importantes en el ciclo de vida del soporte tales como esta para cada versión de Kubernetes.

¿Puedo crear nuevos clústeres con una versión con soporte extendido?

Sí, con la exclusión de `1.22` y `1.21`. Por ejemplo, puede crear un clúster de la versión `1.23`, pero no uno de la `1.22`.

Notas de la versión para las versiones con soporte estándar

En este tema se detallan cambios importantes que debe conocer sobre cada versión de Kubernetes con soporte estándar. Al actualizar, revise detenidamente los cambios que haya habido entre la versión antigua y la nueva de su clúster.

> **Note**
>
> En el caso de `1.24` y clústeres sucesivos, las AMI de Amazon EKS publicadas oficialmente incluyen `containerd` como único tiempo de ejecución. Las versiones de Kubernetes inferiores a la `1.24` usan Docker como tiempo de ejecución predeterminado. Estas versiones tienen una opción de marca de arranque que puede utilizar para probar sus cargas de trabajo en cualquier clúster compatible con `containerd`. Para obtener más información, consulte Amazon EKS dejó de ser compatible con `Dockershim`.

Kubernetes 1.30

Kubernetes `1.30` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.30`, consulte el anuncio del lanzamiento oficial.

> **Important**
>
> - A partir de la versión `1.30` o posterior de Amazon EKS, todos los grupos de nodos administrados recién creados utilizarán automáticamente Amazon Linux 2023 (AL2023) como sistema operativo de nodos de forma predeterminada. Anteriormente, los nuevos grupos de nodos utilizaban Amazon Linux 2 (AL2) de forma predeterminada. Puede seguir utilizando AL2 si lo elige como tipo de AMI cuando crea un nuevo grupo de nodos.
>   - Para obtener más información sobre Amazon Linux, consulte Comparación de Amazon Linux 2 y Amazon Linux 2023 en la Guía del Usuario de Amazon Linux.
>   - Para obtener más información sobre cómo especificar el sistema operativo de un grupo de nodos gestionado, consulte. Creación de un grupo de nodos administrados

- Con Amazon EKS `1.30`, la etiqueta `topology.k8s.aws/zone-id` se añade a los nodos de trabajo. Puede usar IDs de zona de disponibilidad (AZ IDs) para determinar la ubicación de los

recursos de una cuenta respecto de los recursos de otra. Para obtener más información, consulte ID de zona de disponibilidad para los recursosAWS en la Guía del usuario de AWS RAM.

- A partir de la versión `1.30`, Amazon EKS ya no incluye la anotación `default` en el recurso `gp2` `StorageClass` aplicada a los clústeres recién creados. Esto no tiene ningún impacto si hace referencia a esta clase de almacenamiento por su nombre. Debe tomar medidas si confiaba en tener un `StorageClass` predeterminado en el clúster. Debe hacer referencia a `StorageClass` por su nombre `gp2`. Como alternativa, puede implementar la clase de almacenamiento predeterminada recomendada por Amazon EBS configurando el parámetro `defaultStorageClass.enabled` en true al instalar `v1.31.0` o posteriormente el `aws-ebs-csi-driver add-on`.
- La política de IAM mínima requerida para el rol de IAM del clúster de Amazon EKS ha cambiado. La acción `ec2:DescribeAvailabilityZones` es obligatoria. Para obtener más información, consulte Rol de IAM del clúster de Amazon EKS.

Para ver el registro de cambios completo de Kubernetes `1.30`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.30.md.

Kubernetes 1.29

Kubernetes `1.29` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.29`, consulte el anuncio del lanzamiento oficial.

> **Important**
>
> - La versión de la API `flowcontrol.apiserver.k8s.io/v1beta2` en desuso de `FlowSchema` y de `PriorityLevelConfiguration` ya no se ofrece en Kubernetes `v1.29`. Si tiene manifiestos o software cliente que utiliza el grupo de API beta en desuso, debe cambiarlos antes de actualizar a la versión `v1.29`.

- El campo `.status.kubeProxyVersion` para los objetos de nodo ahora está en desuso y el proyecto Kubernetes propone eliminarlo en una versión futura. El campo en desuso no es preciso e históricamente ha sido gestionado por `kubelet`, el cual, en realidad, no conoce la versión `kube-proxy` ni si se ejecuta `kube-proxy`. Si utilizó este campo en un software cliente, deje de hacerlo; la información no es fiable y el campo está en desuso.

- Para reducir la posible superficie expuesta a ataque, la función `LegacyServiceAccountTokenCleanUp` en Kubernetes `1.29` etiqueta los tokens heredados basados en secretos generados automáticamente como no válidos si no se utilizaron por mucho tiempo (1 año de forma predeterminada) y los elimina automáticamente si no se intenta usarlos por mucho tiempo después de marcarlos como no válidos (1 año adicional de forma predeterminada). Para identificar estos tokens, puede ejecutar lo siguiente:

```
kubectl get cm kube-apiserver-legacy-service-account-token-tracking -nkube-system
```

Para ver el registro de cambios completo de Kubernetes `1.29`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.29.md#changelog-since-v1280.

Kubernetes 1.28

Kubernetes `1.28` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.28`, consulte el anuncio del lanzamiento oficial.

- Kubernetes `v1.28` amplió el sesgo admitido entre los componentes del nodo principal y el plano de control en una versión secundaria, de `n-2` a `n-3`, de modo que los componentes del nodo (`kubelet` y `kube-proxy`) de la versión secundaria compatible más antigua puedan funcionar con los componentes del plano de control (`kube-apiserver`, `kube-schedulerkube-controller-manager`, `cloud-controller-manager`) para la versión secundaria compatible más reciente.
- Las métricas `force_delete_pods_total` y `force_delete_pod_errors_total` de `Pod GC Controller` se han mejorado para que tengan en cuenta todas las eliminaciones forzosas de pods. Se ha agregado un motivo a la métrica para indicar si el pod se ha eliminado forzosamente porque se ha finalizado, ha quedado huérfano, va a finalizar con la taint fuera de servicio, o va finalizar y quedar sin programar.
- El controlador `PersistentVolume (PV)` se ha modificado para que asigne automáticamente un valor predeterminado de `StorageClass` a cualquier `PersistentVolumeClaim` que no tenga definido un valor para `storageClassName`. Además, el mecanismo de validación de admisión de `PersistentVolumeClaim` del servidor de API se ha ajustado para que permita cambiar los valores de un estado no establecido a un nombre de `StorageClass` real.

Para ver el registro de cambios completo de Kubernetes `1.28`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.28.md#changelog-since-v1270.

Kubernetes 1.27

Kubernetes `1.27` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.27`, consulte el anuncio del lanzamiento oficial.

Important

- Se ha eliminado la compatibilidad con las anotaciones de `seccomp.security.alpha.kubernetes.io/pod` de `seccomp` alfa y se eliminaron las anotaciones de `container.seccomp.security.alpha.kubernetes.io`. Las anotaciones `seccomp` alfa quedaron en desuso en `1.19`, y con su eliminación en `1.27`, los campos de `seccomp` ya no se rellenarán automáticamente para `Pods` con las anotaciones de `seccomp`. En su lugar, utilice el campo `securityContext.seccompProfile` para `Pods` o contenedores para configurar los perfiles de `seccomp`. Para comprobar si está utilizando las anotaciones de `seccomp` alfa en desuso en su clúster, ejecute el siguiente comando:

  ```
  kubectl get pods --all-namespaces -o json | grep
   -E 'seccomp.security.alpha.kubernetes.io/pod|
  container.seccomp.security.alpha.kubernetes.io'
  ```

- Se quitó el argumento de la línea de comandos `--container-runtime` para `kubelet`. El tiempo de ejecución predeterminado del contenedor para Amazon EKS es `containerd` desde la versión `1.24`, lo que elimina la necesidad de especificar el tiempo de ejecución del contenedor. A partir de `1.27` en adelante, Amazon EKS ignorará el argumento de `--container-runtime` que se pase a los scripts de arranque. Es importante que no pase este argumento a `--kubelet-extra-args` para evitar errores durante el proceso de arranque del nodo. Debe eliminar el argumento de `--container-runtime` de todos tus flujos de trabajo de creación de nodos y scripts de compilación.

- El `kubelet` en Kubernetes `1.27` aumentó el valor predeterminado de `kubeAPIQPS` a `50` y de `kubeAPIBurst` a `100`. Estas mejoras permiten que `kubelet` gestione un mayor volumen de consultas de API, lo que mejora los tiempos de respuesta y el rendimiento. Cuando las demandas de `Pods` aumentan, debido a los requisitos de escalamiento, los valores predeterminados revisados garantizan que `kubelet` pueda administrar de manera eficiente el aumento de la carga de trabajo. Como resultado, los lanzamientos de Pod son más rápidos y las operaciones del clúster son más eficaces.

- Puede utilizar una topología de Pod más detallada para difundir políticas como `minDomain`. Este parámetro le permite especificar el número mínimo de dominios en los que los `Pods` deben estar repartidos. `nodeAffinityPolicy` y `nodeTaintPolicy` permiten un nivel adicional de granularidad en la regulación de la distribución de Pod. Esto se realiza de acuerdo con las afinidades de los nodos, taints y el campo `matchLabelKeys` en el `topologySpreadConstraints` de su especificación de `Pod's`. Esto permite seleccionar los `Pods` para la dispersión de los cálculos tras una actualización progresiva.
- Kubernetes`1.27` promovió al estado beta un nuevo mecanismo de políticas de `StatefulSets` que controla la vida útil de sus `PersistentVolumeClaims` (`PVCs`). La nueva política de retención de PVC le permite especificar si los `PVCs` generados a partir de la plantilla de especificaciones de `StatefulSet` se eliminará o retendrá automáticamente cuando se elimine `StatefulSet` o si se reducen verticalmente las réplicas que contiene `StatefulSet`.
- La opción `goaway-chance` del servidor de API de Kubernetes ayuda a evitar que las conexiones de los clientes `HTTP/2` se bloqueen en una única instancia del servidor de API, al cerrar una conexión de forma aleatoria. Cuando se cierre la conexión, el cliente intentará volver a conectarse y es probable que aterrice en un servidor de API diferente como resultado del equilibrador de carga. La versión `1.27` de Amazon EKS tiene el indicador `goaway-chance` activado. Si su carga de trabajo que se ejecuta en el clúster de Amazon EKS utiliza un cliente que no es compatible con `HTTP GOAWAY`, le recomendamos que actualice su cliente para manejar `GOAWAY` volviendo a conectarse al finalizar la conexión.

Para ver el registro de cambios completo de Kubernetes `1.27`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.27.md#changelog-since-v1260.

Kubernetes 1.26

Kubernetes `1.26` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.26`, consulte el anuncio del lanzamiento oficial.

> **Important**
>
> Kubernetes `1.26` ya no es compatible con CRI `v1alpha2`. Esto hace que el `kubelet` deje de registrar el nodo si el tiempo de ejecución del contenedor no admite CRI `v1`. Esto también significa que Kubernetes `1.26` no es compatible con la versión secundaria `1.5` y anteriores de containerd. Si usa containerd, debe actualizar a la versión containerd `1.6.0` o posterior antes de actualizar cualquier nodo para Kubernetes `1.26`. También debe actualizar cualquier otro entorno de ejecución de contenedor que solo admita `v1alpha2`. Para obtener

más información, consulte al proveedor de tiempo de ejecución del contenedor. De forma predeterminada, las AMI de Amazon Linux y Bottlerocket incluyen la versión containerd `1.6.6`.

- Antes de actualizar a Kubernetes `1.26`, actualice su versión Amazon VPC CNI plugin for Kubernetes a `1.12` o posterior. Si no actualiza a la versión de Amazon VPC CNI plugin for Kubernetes `1.12`o posterior, el Amazon VPC CNI plugin for Kubernetes se bloqueará. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.
- La opción `goaway-chance` del servidor de API de Kubernetes ayuda a evitar que las conexiones de los clientes `HTTP/2` se bloqueen en una única instancia del servidor de API, al cerrar una conexión de forma aleatoria. Cuando se cierre la conexión, el cliente intentará volver a conectarse y es probable que aterrice en un servidor de API diferente como resultado del equilibrador de carga. La versión `1.26` de Amazon EKS tiene el indicador `goaway-chance` activado. Si su carga de trabajo que se ejecuta en el clúster de Amazon EKS utiliza un cliente que no es compatible con `HTTP GOAWAY`, le recomendamos que actualice su cliente para manejar `GOAWAY` volviendo a conectarse al finalizar la conexión.

Para ver el registro de cambios completo de Kubernetes `1.26`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.26.md#changelog-since-v1250.

Notas de la versión para las versiones con soporte extendido

En este tema se detallan cambios importantes que debe conocer sobre cada versión de Kubernetes con soporte extendido. Al actualizar, revise detenidamente los cambios que haya habido entre la versión antigua y la nueva de su clúster.

Kubernetes 1.25

Kubernetes `1.25` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.25`, consulte el anuncio del lanzamiento oficial.

Important

- A partir de la versión `1.25` de Kubernetes, ya no podrá utilizar instancias P2 de Amazon EC2 con las AMI de Amazon Linux aceleradas y optimizadas de Amazon EKS listas para

usar. Estas AMI para las versiones `1.25` o posteriores de Kubernetes serán compatibles con controladores de la serie `NVIDIA 525` o posteriores, que son incompatibles con las instancias `P2`. Sin embargo, los controladores de la serie `NVIDIA 525` o posteriores son compatibles con las instancias `P3`, `P4` y `P5`, de modo que puede utilizarlas con las AMI para la versión `1.25` o posterior de Kubernetes. Antes de que sus clústeres de Amazon EKS se actualicen a la versión `1.25`, migre todas las instancias `P2` a instancias `P3`, `P4` y `P5`. También debe actualizar sus aplicaciones de manera proactiva para que funcionen con la serie `NVIDIA 525` o posterior. Tenemos previsto volver a portar los controladores más recientes de la serie `NVIDIA 525` o posteriores a las versiones `1.23` y `1.24` de Kubernetes a finales de enero de 2024.

- `PodSecurityPolicy` (PSP) se elimina en `1.25` de Kubernetes. PSPs se sustituyen por los estándares Pod Security Admission (PSA) y Pod Security (PSS). PSA es un controlador de admisión integrado que implementa los controles de seguridad que se describen en los PSS. PSA y PSS se gradúan como estables en `1.25` de Kubernetesy están habilitados en Amazon EKS de forma predeterminada. Si tiene PSPs en el clúster, asegúrese de migrar de PSP a PSS de Kubernetes integrado o a una solución de política como código antes de actualizar el clúster a la versión `1.25`. Si no realizas la migración desde PSP, es posible que se produzcan interrupciones en las cargas de trabajo. Para obtener más información, consulte Preguntas frecuentes sobre la eliminación (PSP) de políticas de seguridad de pods.
- La versión `1.25` de Kubernetes contiene cambios que alteran el comportamiento de una característica existente conocida como Prioridad y Equidad de la API (APF). El APF sirve para proteger al servidor API de una posible sobrecarga durante los períodos de mayor volumen de solicitudes. Para ello, restringe el número de solicitudes simultáneas que se pueden procesar en un momento dado. Esto se logra mediante la aplicación de distintos niveles de prioridad y límites a las solicitudes que se originan en diversas cargas de trabajo o usuarios. Este enfoque garantiza que las aplicaciones críticas o las solicitudes de alta prioridad reciban un trato preferencial y, al mismo tiempo, evita que las solicitudes de menor prioridad sobrecarguen el servidor de API. Para obtener más información, consulte Prioridad y equidad de la API en la documentación de Kubernetes o Prioridad y equidad de la API en la Guía de prácticas recomendadas de EKS.

  Estas actualizaciones se introdujeron en PR #10352 y PR #118601. Anteriormente, APF trataba todos los tipos de solicitudes de manera uniforme, y cada solicitud consumía una sola unidad del límite de solicitudes simultáneas. El cambio de comportamiento de la APF asigna unidades de concurrencia más altas a solicitudes de `LIST` debido a la

carga excepcionalmente pesada que estas solicitudes suponen para el servidor de API. El servidor de API estima la cantidad de objetos que devolverá una solicitud de `LIST`. Asigna una unidad de concurrencia que es proporcional al número de objetos devueltos.

Al actualizar a la versión `1.25` de Amazon EKS o superior, este comportamiento actualizado puede provocar cargas de trabajo con solicitudes de `LIST` pesadas (que anteriormente funcionaban sin problemas) para encontrar una limitación de velocidad. Esto se indicaría mediante un código de respuesta HTTP 429. Para evitar posibles interrupciones en la carga de trabajo debido a que las solicitudes de `LIST` tienen una tarifa limitada, le recomendamos encarecidamente que reestructure sus cargas de trabajo para reducir la frecuencia de estas solicitudes. También puede solucionar este problema ajustando la configuración de APF para asignar más capacidad a las solicitudes esenciales y, al mismo tiempo, reducir la capacidad asignada a las no esenciales. Para obtener más información sobre estas técnicas de mitigación, consulte Prevención de solicitudes abandonadas en la Guía de prácticas recomendadas de EKS.

- `1.25` de Amazon EKS incluye mejoras en la autenticación de clústeres que contienen bibliotecas de YAML actualizadas. Si un valor de YAML en el `ConfigMap` de `aws-auth` que se encuentra en el espacio de nombres de `kube-system` comienza con una macro, en la que el primer carácter es una llave, debe añadir comillas (`" "`) antes y después de las llaves (`{ }`). Esto es necesario para garantizar que `v0.6.3` versión `aws-iam-authenticator` analice con precisión el`ConfigMap` de `aws-auth` en `1.25` de Amazon EKS.
- La versión beta de la API (`discovery.k8s.io/v1beta1`) de `EndpointSlice` quedó obsoleta en `1.21` de Kubernetes y ya no se ofrece desde `1.25` de Kubernetes. Esta API se ha actualizado a `discovery.k8s.io/v1`. Para obtener más información, consulte `EndpointSlice` en la documentación del Kubernetes. El `v2.4.6` de AWS Load Balancer Controller y el anterior utilizaban el punto de conexión de `v1beta1` para comunicarse con `EndpointSlices`. Si utiliza la configuración `EndpointSlices` para el AWS Load Balancer Controller, debe actualizar a la `v2.4.7` de AWS Load Balancer Controller antes de actualizar el clúster de Amazon EKS a `1.25`. Si actualiza a `1.25` mientras usa la configuración de `EndpointSlices` para el AWS Load Balancer Controller, el controlador se bloqueará y provocará interrupciones en las cargas de trabajo. Para actualizar el controlador, consulte ¿Qué es el AWS Load Balancer Controller?.

- `SeccompDefault` se promueve a beta en `1.25` de Kubernetes. Al establecer el indicador de `--seccomp-default` al configurar `kubelet`, el tiempo de ejecución del contenedor usa su perfil `seccomp` de `RuntimeDefault`, en lugar del modo no confinado (`seccomp disabled`). Los perfiles predeterminados proporcionan un conjunto sólido de valores predeterminados de seguridad y, al mismo tiempo, preservan la funcionalidad de la carga de trabajo. Aunque este indicador está disponible, Amazon EKS no lo habilita de forma predeterminada, por lo que el comportamiento de Amazon EKS permanece prácticamente sin cambios. Si lo desea, puede empezar a habilitarlo en sus nodos. Para obtener más información, consulte el tutorial Restringir las llamadas de sistema de un contenedor con seccomp en la documentación de Kubernetes.
- Se ha eliminado de `1.24` de Kubernetes y posteriores la compatibilidad con la interfaz de tiempo de ejecución de contenedores (CRI) para Docker (también conocida como `Dockershim`). El único tiempo de ejecución de contenedores en AMIs oficial de Amazon EKS oficial para clústeres `1.24` de Kubernetes y posteriores es `containerd`. Antes de pasar a `1.24` de Amazon EKS o a una versión más nueva, elimine cualquier referencia a los indicadores del script de arranque que ya no sean compatibles. Para obtener más información, consulte Amazon EKS dejó de ser compatible con `Dockershim`.
- La compatibilidad con consultas comodín quedó obsoleta en `1.8.7` de CoreDNS y se eliminó en `1.9` de CoreDNS. Esto se hizo como medida de seguridad. Las consultas comodín ya no funcionan y devuelven `NXDOMAIN` en lugar de una dirección IP.
- La opción `goaway-chance` del servidor de API de Kubernetes ayuda a evitar que las conexiones de los clientes `HTTP/2` se bloqueen en una única instancia del servidor de API, al cerrar una conexión de forma aleatoria. Cuando se cierre la conexión, el cliente intentará volver a conectarse y es probable que aterrice en un servidor de API diferente como resultado del equilibrador de carga. La versión `1.25` de Amazon EKS tiene el indicador `goaway-chance` activado. Si su carga de trabajo que se ejecuta en el clúster de Amazon EKS utiliza un cliente que no es compatible con `HTTP GOAWAY`, le recomendamos que actualice su cliente para manejar `GOAWAY` volviendo a conectarse al finalizar la conexión.

Para ver el registro de cambios completo de Kubernetes `1.25`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.25.md#changelog-since-v1240.

Kubernetes 1.24

Kubernetes `1.24` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.24`, consulte el anuncio del lanzamiento oficial.

Important

- A partir de las versión `1.24` de Kubernetes, las nuevas API beta no están habilitadas en los clústeres de forma predeterminada. De forma predeterminada, las API beta existentes y las nuevas versiones de las API beta existentes siguen habilitadas. Amazon EKS sigue el mismo comportamiento que en la versión anterior `1.24` de Kubernetes. Las puertas de características que controlan las nuevas características para las operaciones de API nuevas y existentes están habilitadas de forma predeterminada. Esto está alineado con la versión anterior de Kubernetes. Para obtener más información, consulte KEP-3136: Beta APIs Are Off by Default (KEP-3136: las API beta están desactivadas de forma predeterminada) en GitHub.
- Se ha eliminado la compatibilidad con la interfaz de tiempo de ejecución de contenedores (CRI) para Docker (también conocida como Dockershim) de Kubernetes `1.24`. Las AMI oficiales de Amazon EKS tienen containerd como único tiempo de ejecución. Antes de pasar a Amazon EKS `1.24` o a una versión superior, debe eliminar cualquier referencia a los indicadores del script de arranque que ya no sean compatibles. También debe asegurarse de que el reenvío de IP esté habilitado para sus nodos de trabajo. Para obtener más información, consulte Amazon EKS dejó de ser compatible con `Dockershim`.
- Si ya ha configurado Fluentd configurado para Container Insights, debe migrar Fluentd a Fluent Bit antes de actualizar su clúster. Los analizadores de Fluentd están configurados para analizar únicamente los mensajes de registro en formato JSON. A diferencia de `dockerd`, el tiempo de ejecución del contenedor `containerd`contiene mensajes de registro que no están en formato JSON. Si no migra a Fluent Bit, algunos de los analizadores de Fluentd's configurados generarán una enorme cantidad de errores dentro del contenedor de Fluentd. Para enviar registros a CloudWatch Logs, consulte Configurar Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs.
- En la versión `1.23` de Kubernetes y anteriores, los certificados de servicio de `kubelet` con nombres alternativos de asunto (SAN) de IP y DNS no verificables se emitían automáticamente con SAN no verificables. Estas SAN no verificables se omiten del certificado aprovisionado. En los clústeres de versiones `1.24` y posteriores, no se emiten certificados de servicio de `kubelet` si no se puede verificar alguna SAN. Esto impide que los comandos exec de `kubectl` y registros de `kubectl` funcionen. Para obtener más información, consulte Consideraciones sobre la firma de certificados antes de actualizar el clúster a la versión 1.24 de Kubernetes.

- Al actualizar un clúster de Amazon EKS `1.23` que utilice Fluent Bit, debe asegurarse de que `k8s/1.3.12` esté en ejecución o en una versión posterior. Para ello, vuelva a aplicar el último archivo YAML Fluent Bit aplicable desdeGitHub. Para obtener más información, consulte Configurar Fluent Bit en la Guía del usuario de Amazon CloudWatch.

- Puede utilizar las sugerencias de reconocimiento de topología para indicar su preferencia de mantener el tráfico en la zona cuando los nodos de trabajo del clúster se implementen en varias zonas de disponibilidad. El enrutamiento del tráfico dentro de una zona puede ayudar a reducir los costos y mejorar el rendimiento de la red. De forma predeterminada, las sugerencias con reconocimiento de topología están habilitadas en Amazon EKS `1.24`. Para obtener más información, consulte las sugerencias con reconocimiento de topología en la documentación de Kubernetes.
- La eliminación de `PodSecurityPolicy` (PSP) está programada en Kubernetes `1.25`. Las PSPs están siendo reemplazados por Pod Security Admission (PSA). La PSA es un controlador de admisión integrado que utiliza los controles de seguridad que se describen en Pod Security Standards (PSS). Tanto PSA como PSS son características beta y están habilitadas en Amazon EKS de forma predeterminada. Para hacer frente a la eliminación de PSP en la versión `1.25`, le recomendamos que implemente PSS en Amazon EKS. Para obtener más información, consulte Implementing Pod Security Standards in Amazon EKS (Implementación de estándares de seguridad para módulos en Amazon EKS) en el blog de AWS.
- La credencial exec de `client.authentication.k8s.io/v1alpha1` se elimina en Kubernetes `1.24`. La credencial exec API estaba generalmente disponible en Kubernetes `1.22`. Si utiliza un complemento de credenciales client-go que se basa en la API `v1alpha1`, póngase en contacto con el distribuidor de su complemento para saber cómo migrar a la API `v1`.
- Para Kubernetes `1.24`, contribuimos con una característica al proyecto del escalador automático de clústeres que simplifica el escalado de los grupos de nodos administrados de Amazon EKS a cero nodos y desde cero nodos. Anteriormente, para que el escalador automático de clústeres entendiera los recursos, las etiquetas y las taints de un grupo de nodos administrado que se escalaba a cero nodos, había que etiquetar el grupo subyacente de Amazon EC2 Auto Scaling con los detalles de los nodos de los que era responsable. Ahora, cuando no hay nodos en ejecución en el grupo de nodos administrado, el escalador automático de clústeres llama a la operación de API `DescribeNodegroup` de Amazon EKS. Esta operación de API proporciona la información que el escalador automático de clústeres requiere sobre los recursos, las etiquetas y las taints del grupo de nodos administrado. Esta característica requiere que agregue el permiso `eks:DescribeNodegroup` a la política de IAM de la cuenta de servicio del escalador automático

de clústeres. Cuando el valor de una etiqueta de escalador automático de clústeres del grupo de escalado automático que alimenta un grupo de nodos administrado por Amazon EKS entra en conflicto con el propio grupo de nodos, el escalador automático de clústeres prefiere el valor de la etiqueta de grupo de escalado automático. Esto es para que se puedan anular los valores según sea necesario. Para obtener más información, consulte Escalado automático.

- Si tiene la intención de utilizar los tipos de instancias Inferentia o Trainium con Amazon EKS `1.24`, debe actualizar a la versión 1.9.3.0 o posterior del complemento del dispositivo AWS Neuron. Para obtener más información, consulte la versión [1.9.3.0] de Neuron K8 en la documentación de AWS Neuron.
- `Containerd` tiene la opción `IPv6` activada para Pods, de forma predeterminada. Aplica la configuración del núcleo del nodo a los espacios de nombres de la red del Pod. Por este motivo, los contenedores de un Pod se enlazan a las direcciones de bucle `IPv4` (`127.0.0.1`) y `IPv6` (`::1`). `IPv6` es el protocolo de comunicación predeterminado. Antes de actualizar el clúster a la versión `1.24`, le recomendamos que pruebe los Pods de varios contenedores. Modifique las aplicaciones para que puedan enlazarse a todas las direcciones IP en las interfaces de bucle invertido. La mayoría de las bibliotecas permiten el enlace de `IPv6`, que es compatible con versiones anteriores`IPv4`. Si no es posible modificar el código de la aplicación, tiene dos opciones:
  - Ejecute un contenedor de `init` y configure `disable ipv6` a `true` (`sysctl -w net.ipv6.conf.all.disable_ipv6=1`).
  - Configure un webhook de admisión mutante para insertar un contenedor de `init` junto con los Pods de su aplicación.

  Si necesita bloquear `IPv6` para los Pods de todos los nodos, puede que tenga que deshabilitar `IPv6` en sus instancias.
- La opción `goaway-chance` del servidor de API de Kubernetes ayuda a evitar que las conexiones de los clientes `HTTP/2` se bloqueen en una única instancia del servidor de API, al cerrar una conexión de forma aleatoria. Cuando se cierre la conexión, el cliente intentará volver a conectarse y es probable que aterrice en un servidor de API diferente como resultado del equilibrador de carga. La versión `1.24` de Amazon EKS tiene el indicador `goaway-chance` activado. Si su carga de trabajo que se ejecuta en el clúster de Amazon EKS utiliza un cliente que no es compatible con `HTTP GOAWAY`, le recomendamos que actualice su cliente para manejar `GOAWAY` volviendo a conectarse al finalizar la conexión.

Para ver el registro de cambios completo de Kubernetes `1.24`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.24.md#changelog-since-v1230.

Kubernetes 1.23

Kubernetes `1.23` ya está disponible en Amazon EKS. Para obtener más información sobre Kubernetes `1.23`, consulte el anuncio del lanzamiento oficial.

Important

- La característica de migración de volumen en árbol a la interfaz de almacenamiento de contenedores (CSI) de Kubernetes está habilitada. Esta característica permite reemplazar los complementos de almacenamiento en árbol de Kubernetes para Amazon EBS con el controlador CSI de Amazon EBS correspondiente. Para obtener más información, consulte 1.17 Feature: Kubernetes In-Tree to CSI Volume Migration Moves to Beta (Característica 1.17 de Kubernetes: la migración del volumen en árbol a CSI de Kubernetes pasa a versión beta) en el blog de Kubernetes.

  La característica traduce las API del árbol a API de CSI equivalentes y delega las operaciones a un controlador de CSI de reemplazo. Con esta característica, si utiliza los objetos `StorageClass`, `PersistentVolume` y `PersistentVolumeClaim` existentes que pertenecen a estas cargas de trabajo, es probable que no haya ningún cambio notable. La característica permite que Kubernetes delegue todas las operaciones de administración de almacenamiento del complemento en árbol al controlador de CSI. Si utiliza volúmenes de Amazon EBS en un clúster existente, instale el controlador CSI de Amazon EBS en el clúster antes de actualizar el clúster a la versión `1.23`. Si no instala el controlador antes de actualizar un clúster existente, es posible que se produzcan interrupciones en las cargas de trabajo. Si va a implementar cargas de trabajo que utilizan volúmenes de Amazon EBS en un clúster `1.23` nuevo, instale el controlador CSI de Amazon EBS en el clúster antes de implementar las cargas de trabajo en el clúster. Para obtener instrucciones acerca de cómo instalar el controlador de CSI de Amazon EBS en su clúster, consulte Controlador CSI de Amazon EBS. Para conocer las preguntas frecuentes acerca de la característica de migración, consulte Preguntas frecuentes sobre migración de CSI de Amazon EBS.
- El soporte extendido para las AMI de Windows optimizadas para Amazon EKS que sean publicadas por AWS no está disponible para la versión `1.23` de Kubernetes; sin embargo, no está disponible para la versión `1.24` o una posterior de Kubernetes.

- Kubernetes dejó de admitir `dockershim` en versión `1.20` y eliminamos `dockershim` en versión `1.24`. Para obtener más información, consulte Kubernetes deja atrás Dockershim: compromisos y pasos siguientes en el blog de Kubernetes. Amazon EKS pondrá fin a la compatibilidad para `dockershim` a partir de la versión `1.24` de Amazon EKS. A partir de la versión `1.24` de Amazon EKS, las AMI oficiales de Amazon EKS tendrán `containerd` como único tiempo de ejecución.

  Si bien la versión `1.23` de Amazon EKS sigue admitiendo `dockershim`, le recomendamos que empiece a probar sus aplicaciones ahora para identificar y eliminar cualquier dependencia de Docker. De esta forma, estará preparado para actualizar el clúster a la versión `1.24`. Para obtener más información acerca de la eliminación de `dockershim`, consulte Amazon EKS dejó de ser compatible con `Dockershim`.
- Kubernetes graduó las redes de doble pila de `IPv4`/`IPv6` para Pods, servicios y nodos a disponibilidad general. Sin embargo, Amazon EKS y el Amazon VPC CNI plugin for Kubernetes no admiten redes de doble pila. Los clústeres pueden asignar direcciones `IPv4` o `IPv6` a los Pods y los servicios, pero no se pueden asignar ambos tipos de direcciones.
- Kubernetes graduó la característica de admisión de seguridad del pod (PSA) a beta. La característica está habilitada de forma predeterminada. Para obtener más información, consulte Pod Security Admission (Admisión de seguridad del pod) en la documentación de Kubernetes. La PSA reemplaza el controlador de admisión de Pod Security Policy (PSP). El controlador de admisión de PSP no es compatible y está programado su eliminación en Kubernetes versión `1.25`.

  El controlador de admisión de PSP hace cumplir los estándares de seguridad de Pod en Pods en un espacio de nombres basado en etiquetas de espacio de nombres específicas que establecen el nivel de cumplimiento. Para obtener más información, consulte Estándares de seguridad del pod (PSS) y admisión de seguridad del pod (PSA) en la guía de prácticas recomendadas de Amazon EKS.
- La imagen de `kube-proxy` implementada con clústeres ahora es la imagen base mínima mantenida por Amazon EKS Distro (EKS-D). La imagen contiene paquetes mínimos y no tiene administradores de paquetes ni intérpretes de comandos.
- Kubernetes graduó contenedores efímeros a beta. Los contenedores efímeros son contenedores temporales que se ejecutan en el mismo espacio de nombres que un Pod existente. Puede usarlos para observar el estado de los Pods y los contenedores con fines de solución de problemas y depuración. Esto resulta especialmente útil para solucionar problemas interactivos cuando `kubectl exec` es insuficiente porque se ha bloqueado un contenedor o una imagen de contenedor no incluye utilidades de depuración. Un ejemplo de contenedor que incluye una utilidad

de depuración es distroless images (Imágenes distroless). Para obtener más información, consulte Depuración con un contenedor de depuración efímero en la documentación de Kubernetes.

- Kubernetes graduó la API de `HorizontalPodAutoscaler` `autoscaling/v2` estable de disponibilidad general. La API `HorizontalPodAutoscaler` `autoscaling/v2beta2` ha quedado obsoleta. No estará disponible en `1.26`.
- La opción `goaway-chance` del servidor de API de Kubernetes ayuda a evitar que las conexiones de los clientes `HTTP/2` se bloqueen en una única instancia del servidor de API, al cerrar una conexión de forma aleatoria. Cuando se cierre la conexión, el cliente intentará volver a conectarse y es probable que aterrice en un servidor de API diferente como resultado del equilibrador de carga. La versión `1.23` de Amazon EKS tiene el indicador `goaway-chance` activado. Si su carga de trabajo que se ejecuta en el clúster de Amazon EKS utiliza un cliente que no es compatible con `HTTP GOAWAY`, le recomendamos que actualice su cliente para manejar `GOAWAY` volviendo a conectarse al finalizar la conexión.

Para ver el registro de cambios completo de Kubernetes `1.23`, consulte https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.23.md#changelog-since-v1220.

Notas de la versión para las versiones 1.21 y 1.22

> **Important**
>
> No puede crear clústeres nuevos con estas versiones.

En este tema se especifican cambios importantes que debe conocer sobre las versiones `1.22` y `1.21`. Al actualizar, revise detenidamente los cambios que haya habido entre la versión antigua y la nueva de su clúster.

Versión **`1.22`** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de `1.22`: `DefaultStorageClass`, `DefaultTolerationSeconds`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `ResourceQuota`, `ServiceAccount`, `ValidatingAdmissionWebhook`, `PodSecurityPolicy`, `TaintNodesByCondition`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `ExtendedResourceToleration`, `CertificateApproval`,

`PodPriority`, `CertificateSigning`, `CertificateSubjectRestriction`, `RuntimeClass` y `DefaultIngressClass`.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
`1.22.17`	`eks.28`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de mayo de 2024
`1.22.17`	`eks.26`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	1 de abril de 2024
`1.22.17`	`eks.14`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de junio de 2023
`1.22.17`	`eks.13`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
`1.22.17`	`eks.12`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de mayo de 2023
`1.22.17`	`eks.11`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	24 de marzo de 2023
`1.22.16`	`eks.10`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	27 de enero de 2023
`1.22.15`	`eks.9`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de diciembre de 2022
`1.22.15`	`eks.8`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de noviembre de 2022
`1.22.15`	`eks.7`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de noviembre de 2022
`1.22.13`	`eks.6`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	21 de septiembre de 2022

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.22.10	eks.5	Nueva versión de plataforma con mejoras de resiliencia de etcd.	15 de agosto de 2022
1.22.10	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad. Esta versión de plataforma también presenta un nuevo controlador de etiquetado que etiqueta todos los nodos de trabajo con aws:eks:cluster-name para facilitar la asignación de costos para estos nodos de trabajo. Para obtener más información, consulte Etiquetado de los recursos para facturación.	21 de julio de 2022
1.22.10	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de julio de 2022
1.22.9	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	31 de mayo de 2022
1.22.6	eks.1	Versión inicial 1.22 de Kubernetes para Amazon EKS.	4 de abril de 2022

Versión **1.21** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.21: DefaultStorageClass, DefaultTolerationSeconds, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, ResourceQuota, ServiceAccount, ValidatingAdmissionWebhook, PodSecurityPolicy, TaintNodesByCondition, StorageObjectInUseProtection, PersistentVolumeClaimResize, ExtendedResourceToleration, CertificateApproval,

`PodPriority`, `CertificateSigning`, `CertificateSubjectRestriction`, `RuntimeClass` y `DefaultIngressClass`.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
`1.21.14`	`eks.33`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de mayo de 2024
`1.21.14`	`eks.31`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	1 de abril de 2024
`1.21.14`	`eks.18`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
`1.21.14`	`eks.17`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de mayo de 2023
`1.21.14`	`eks.16`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	24 de marzo de 2023
`1.21.14`	`eks.15`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	27 de enero de 2023
`1.21.14`	`eks.14`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de diciembre de 2022
`1.21.14`	`eks.13`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de noviembre de 2022

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.21.14	eks.12	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de noviembre de 2022
1.21.13	eks.11	Nueva versión de plataforma con mejoras de resiliencia de etcd.	10 de octubre de 2022
1.21.13	eks.10	Nueva versión de plataforma con mejoras de resiliencia de etcd.	15 de agosto de 2022
1.21.13	eks.9	Nueva versión de la plataform a con mejoras y correcciones de seguridad. Esta versión de plataforma también presenta un nuevo controlador de etiquetad o que etiqueta todos los nodos de trabajo con aws:eks:c luster-name para facilitar la asignación de costos para estos nodos de trabajo. Para obtener más información, consulte Etiquetado de los recursos para facturación.	21 de julio de 2022
1.21.13	eks.8	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de julio de 2022
1.21.12	eks.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	31 de mayo de 2022

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.21.9	eks.6	El punto de conexión AWS Security Token Service vuelve al punto de conexión global desde la versión anterior de la plataforma. Si desea utilizar el punto de conexión regional al utilizar roles de IAM para cuentas de servicio, debe habilitarlo. Para obtener instrucciones sobre cómo habilitar el punto de conexión regional, consulte Configure el punto de conexión AWS Security Token Service de una cuenta de servicio.	8 de abril de 2022
1.21.5	eks.5	Cuando se utiliza Roles de IAM para cuentas de servicio, el punto de conexión regional AWS Security Token Service ahora se utiliza de forma predeterminada en lugar del punto de conexión global. Sin embargo, este cambio se revertió al punto de conexión global en eks.6. Un programador Fargate actualizado aprovisiona nodos a un ritmo significativamente mayor durante implement aciones grandes.	10 de marzo de 2022

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.21.5	eks.4	La versión 1.10.1-eksbuild.1 del complemento de Amazon VPC CNI autoadministrado y Amazon EKS es ahora la versión predeterminada implementada.	13 de diciembre de 2021
1.21.2	eks.3	Nueva versión de plataforma compatible con la administración de direcciones IPv4 de Windows en el controlador de recursos de VPC que se ejecuta en el plano de control de Kubernetes. Se ha agregado la directiva de filtros de Kubernetes para el registro de Fluent Bit en Fargate.	8 de noviembre de 2021
1.21.2	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de septiembre de 2021
1.21.2	eks.1	Versión inicial 1.21 de Kubernetes para Amazon EKS.	19 de julio de 2021

Versiones de la plataforma de Amazon EKS

Las versiones de la plataforma de Amazon EKS representan las funciones del plano de control del clúster de Amazon EKS, como qué marcas de servidor de API de Kubernetes están habilitadas, así como la versión de parche de Kubernetes actual. Cada versión secundaria de Kubernetes tiene una o varias versiones de la plataforma de Amazon EKS asociadas. Las versiones de la plataforma para las diferentes versiones secundarias de Kubernetes son independientes. Puede recuperar la versión de la plataforma actual de su clúster mediante AWS CLI o AWS Management Console. Si tiene un

clúster local en AWS Outposts, consulte Versiones de la plataforma de clústeres locales de Amazon EKS en lugar de este tema.

Cuando se encuentra disponible una versión secundaria de Kubernetes nueva en Amazon EKS, como, por ejemplo, la 1.30, la versión inicial de la plataforma de Amazon EKS para esa versión secundaria de Kubernetes comienza por `eks.1`. Sin embargo, Amazon EKS lanza nuevas versiones de la plataforma de forma periódica para habilitar nuevos ajustes de plano de control de Kubernetes y proporcionar revisiones de seguridad.

Cuando nuevas versiones de la plataforma de Amazon EKS se encuentran disponibles para una versión secundaria:

- El número de versión de la plataforma de Amazon EKS se incrementa (`eks.`*n+1*).
- Amazon EKS actualiza automáticamente todos los clústeres existentes a la última versión de la plataforma de Amazon EKS para su versión secundaria de Kubernetes correspondiente. Las actualizaciones automáticas de las versiones de la plataforma de Amazon EKS existentes se implementan de forma incremental. El proceso de implementación puede tardar algún tiempo. Si necesita las características de la última versión de la plataforma de Amazon EKS de forma inmediata, debe crear un nuevo clúster de Amazon EKS.

  Si el clúster está más de dos versiones de plataforma por detrás de la versión de la plataforma actual, es posible que Amazon EKS no haya podido actualizar automáticamente el clúster. Para obtener más información sobre lo que puede causar esto, consulte La versión de la plataforma Amazon EKS está más de dos versiones por detrás de la versión de plataforma actual.
- Amazon EKS puede publicar una nueva AMI de nodo con la versión de parche correspondiente. Sin embargo, todas las versiones de parche son compatibles entre el plano de control de EKS y las AMI de nodo para una versión secundaria de Kubernetes determinada.

Las nuevas versiones de la plataforma de Amazon EKS no introducen cambios bruscos ni provocan interrupciones de servicio.

Los clústeres siempre se crean con la última versión de la plataforma de Amazon EKS disponible (`eks.`*n*) para la versión de Kubernetes especificada. Si actualiza su clúster a una nueva versión secundaria de Kubernetes, el clúster recibe la versión de la plataforma de Amazon EKS actual para la versión secundaria de Kubernetes a la que se actualizó.

Las versiones de la plataforma de Amazon EKS actuales y recientes se describen en las siguientes tablas.

Versión `1.30` de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de `1.30`: `NodeRestriction`, `ExtendedResourceToleration`, `NamespaceLifecycle`, `LimitRanger`, `ServiceAccount`, `TaintNodesByCondition`, `PodSecurity`, `Priority`, `DefaultTolerationSeconds`, `DefaultStorageClass`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `RuntimeClass`, `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `MutatingAdmissionWebhook`, `ValidatingAdmissionWebhook`, `ResourceQuota`.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
`1.30.1`	`eks.3`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
`1.30.0`	`eks.2`	Versión inicial de Kubernetes `1.30` para Amazon EKS. Para obtener más información, consulte Kubernetes 1.30.	23 de mayo de 2024

Versión `1.29` de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de `1.29`: `NodeRestriction`, `ExtendedResourceToleration`, `NamespaceLifecycle`, `LimitRanger`, `ServiceAccount`, `TaintNodesByCondition`, `PodSecurity`, `Priority`, `DefaultTolerationSeconds`, `DefaultStorageClass`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `RuntimeClass`, `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `MutatingAdmissionWebhook`, `ValidatingAdmissionWebhook`, `ResourceQuota`.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.29.5	eks.8	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
1.29.4	eks.7	Nueva versión de la plataforma con escalado automático de CoreDNS, mejoras y correcciones de seguridad. Para obtener más información sobre el escalado automático de CoreDNS, consulte Escalado automático CoreDNS.	16 de mayo de 2024
1.29.3	eks.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
1.29.1	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024
1.29.1	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
1.29.1	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
1.29.0	eks.1	Versión inicial de Kubernetes 1.29 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.29.	23 de enero de 2024

Versión **1.28** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.28: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority,

`DefaultTolerationSeconds`, `DefaultStorageClass`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `RuntimeClass`, `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `MutatingAdmissionWebhook`, `ValidatingAdmissionWebhook`, `ResourceQuota`.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
`1.28.10`	`eks.14`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
`1.28.9`	`eks.13`	Nueva versión de la plataforma con escalado automático de CoreDNS, mejoras y correcciones de seguridad. Para obtener más información sobre el escalado automático de CoreDNS, consulte Escalado automático CoreDNS.	16 de mayo de 2024
`1.28.8`	`eks.12`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
`1.28.7`	`eks.11`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024
`1.28.7`	`eks.10`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
`1.28.6`	`eks.9`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
`1.28.5`	`eks.7`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de enero de 2024
`1.28.4`	`eks.6`	Nueva versión de la plataforma con entradas de acceso, mejoras y correcciones de seguridad.	14 de diciembre de 2023

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.28.4	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de diciembre de 2023
1.28.3	eks.4	Nueva versión de la plataforma con Pod Identities de EKS, mejoras y correcciones de seguridad.	10 de noviembre de 2023
1.28.3	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	3 de noviembre de 2023
1.28.2	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de octubre de 2023
1.28.1	eks.1	Versión inicial de Kubernetes 1.28 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.28.	26 de septiembre de 2023

Versión **1.27** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.27: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, MutatingAdmissionWebhook, ValidatingAdmissionWebhook, ResourceQuota.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.27.14	eks.18	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
1.27.13	eks.17	Nueva versión de la plataforma con escalado automático de CoreDNS, mejoras y correcciones de seguridad. Para obtener más información sobre el escalado automático de CoreDNS, consulte Escalado automático CoreDNS.	16 de mayo de 2024
1.27.12	eks.16	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
1.27.11	eks.15	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024
1.27.11	eks.14	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
1.27.10	eks.13	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
1.27.9	eks.11	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de enero de 2024
1.27.8	eks.10	Nueva versión de la plataforma con entradas de acceso, mejoras y correcciones de seguridad.	14 de diciembre de 2023
1.27.8	eks.9	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de diciembre de 2023

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.27.7	eks.8	Nueva versión de la plataforma con Pod Identities de EKS, mejoras y correcciones de seguridad.	10 de noviembre de 2023
1.27.7	eks.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	3 de noviembre de 2023
1.27.6	eks.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de octubre de 2023
1.27.4	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de agosto de 2023
1.27.4	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de julio de 2023
1.27.3	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de junio de 2023
1.27.2	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
1.27.1	eks.1	Versión inicial de Kubernetes 1.27 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.27.	24 de mayo de 2023

Versión **1.26** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.26: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle, LimitRanger, ServiceAccount, TaintNodesByCondition, PodSecurity, Priority, DefaultTolerationSeconds, DefaultStorageClass, StorageObjectInUseProtection, PersistentVolumeClaimResize, RuntimeClass, CertificateApproval,

`CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `MutatingAdmissionWebhook`, `ValidatingAdmissionWebhook`, `ResourceQuota`.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
`1.26.15`	`eks.19`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
`1.26.15`	`eks.18`	Nueva versión de la plataforma con escalado automático de CoreDNS, mejoras y correcciones de seguridad. Para obtener más información sobre el escalado automático de CoreDNS, consulte Escalado automático CoreDNS.	16 de mayo de 2024
`1.26.15`	`eks.17`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
`1.26.14`	`eks.16`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024
`1.26.14`	`eks.15`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
`1.26.13`	`eks.14`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
`1.26.12`	`eks.12`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de enero de 2024
`1.26.11`	`eks.11`	Nueva versión de la plataforma con entradas de acceso, mejoras y correcciones de seguridad.	14 de diciembre de 2023
`1.26.11`	`eks.10`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de diciembre de 2023

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.26.10	eks.9	Nueva versión de la plataforma con Pod Identities de EKS, mejoras y correcciones de seguridad.	10 de noviembre de 2023
1.26.10	eks.8	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	3 de noviembre de 2023
1.26.9	eks.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de octubre de 2023
1.26.7	eks.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de agosto de 2023
1.26.7	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de julio de 2023
1.26.6	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de junio de 2023
1.26.5	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
1.26.4	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de mayo de 2023
1.26.2	eks.1	Versión inicial de Kubernetes 1.26 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.26.	11 de abril de 2023

Versión **1.25** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.25: NodeRestriction, ExtendedResourceToleration, NamespaceLifecycle,

`LimitRanger`, `ServiceAccount`, `TaintNodesByCondition`, `PodSecurity`, `Priority`, `DefaultTolerationSeconds`, `DefaultStorageClass`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `RuntimeClass`, `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `MutatingAdmissionWebhook`, `ValidatingAdmissionWebhook`, `ResourceQuota`.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
`1.25.16`	`eks.20`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
`1.25.16`	`eks.19`	Nueva versión de la plataforma con escalado automático de CoreDNS, mejoras y correcciones de seguridad. Para obtener más información sobre el escalado automático de CoreDNS, consulte Escalado automático CoreDNS.	16 de mayo de 2024
`1.25.16`	`eks.18`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
`1.25.16`	`eks.17`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024
`1.25.16`	`eks.16`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
`1.25.16`	`eks.15`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
`1.25.16`	`eks.13`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de enero de 2024

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.25.16	eks.12	Nueva versión de la plataforma con entradas de acceso, mejoras y correcciones de seguridad.	14 de diciembre de 2023
1.25.16	eks.11	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de diciembre de 2023
1.25.15	eks.10	Nueva versión de la plataforma con Pod Identities de EKS, mejoras y correcciones de seguridad.	10 de noviembre de 2023
1.25.15	eks.9	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	3 de noviembre de 2023
1.25.14	eks.8	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de octubre de 2023
1.25.12	eks.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de agosto de 2023
1.25.12	eks.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de julio de 2023
1.25.11	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de junio de 2023
1.25.10	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
1.25.9	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de mayo de 2023
1.25.8	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	24 de marzo de 2023

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.25.6	eks.1	Versión inicial de Kubernetes 1.25 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.25.	21 de febrero de 2023

Versión **1.24** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.24: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurityPolicy, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition y ValidatingAdmissionWebhook.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.24.17	eks.23	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
1.24.17	eks.22	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de mayo de 2024
1.24.17	eks.21	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
1.24.17	eks.20	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.24.17	eks.19	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
1.24.17	eks.18	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
1.24.17	eks.16	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de enero de 2024
1.24.17	eks.15	Nueva versión de la plataforma con entradas de acceso, mejoras y correcciones de seguridad.	14 de diciembre de 2023
1.24.17	eks.14	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de diciembre de 2023
1.24.17	eks.13	Nueva versión de la plataforma con Pod Identities de EKS, mejoras y correcciones de seguridad.	10 de noviembre de 2023
1.24.17	eks.12	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	3 de noviembre de 2023
1.24.17	eks.11	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de octubre de 2023
1.24.16	eks.10	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de agosto de 2023
1.24.16	eks.9	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de julio de 2023
1.24.15	eks.8	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de junio de 2023

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.24.14	eks.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
1.24.13	eks.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de mayo de 2023
1.24.12	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	24 de marzo de 2023
1.24.8	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	27 de enero de 2023
1.24.7	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de diciembre de 2022
1.24.7	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de noviembre de 2022
1.24.7	eks.1	Versión inicial de Kubernetes 1.24 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.24.	15 de noviembre de 2022

Versión **1.23** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de 1.23: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurityPolicy, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition y ValidatingAdmissionWebhook.

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.23.17	eks.25	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de junio de 2024
1.23.17	eks.24	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de mayo de 2024
1.23.17	eks.23	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de abril de 2024
1.23.17	eks.22	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	29 de marzo de 2024
1.23.17	eks.21	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	20 de marzo de 2024
1.23.17	eks.20	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de marzo de 2024
1.23.17	eks.18	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	17 de enero de 2024
1.23.17	eks.17	Nueva versión de la plataforma con entradas de acceso, mejoras y correcciones de seguridad.	14 de diciembre de 2023
1.23.17	eks.16	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	12 de diciembre de 2023
1.23.17	eks.15	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	10 de noviembre de 2023
1.23.17	eks.14	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	3 de noviembre de 2023

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.23.17	eks.13	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	16 de octubre de 2023
1.23.17	eks.12	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de agosto de 2023
1.23.17	eks.11	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de julio de 2023
1.23.17	eks.10	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	30 de junio de 2023
1.23.17	eks.9	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	9 de junio de 2023
1.23.17	eks.8	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de mayo de 2023
1.23.17	eks.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	24 de marzo de 2023
1.23.14	eks.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	27 de enero de 2023
1.23.13	eks.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	5 de diciembre de 2022
1.23.13	eks.4	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	18 de noviembre de 2022
1.23.12	eks.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	7 de noviembre de 2022
1.23.10	eks.2	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	21 de septiembre de 2022

Versión de Kubernetes	Versión de la plataforma de EKS	Notas de la versión	Fecha de publicación
1.23.7	eks.1	Versión inicial de Kubernetes 1.23 para Amazon EKS. Para obtener más información, consulte Kubernetes 1.23.	11 de agosto de 2022

Obtenga la versión actual de la plataforma

Para obtener la versión de la plataforma actual para su clúster (consola)

1. Abra la consola de Amazon EKS.
2. En el panel de navegación, seleccione Clusters (Clústeres).
3. En la lista de clústeres, elija el nombre del clúster para comprobar la versión de la plataforma.
4. Elija la pestaña Overview (Información general).
5. La versión de la plataforma está disponible en la sección de detalles.

Para obtener la versión de la plataforma actual de su clúster (AWS CLI)

1. Determine el nombre del clúster del que desea comprobar la versión de la plataforma.
2. Ejecute el siguiente comando:

```
aws eks describe-cluster --name my-cluster --query cluster.platformVersion
```

Un ejemplo de salida sería el siguiente.

```
"eks.10"
```

Escalado automático

El escalado automático es una función que escala y reduce horizontalmente los recursos de manera automática para satisfacer las demandas cambiantes. Esta es una función importante de Kubernetes que, de otro modo, requeriría muchos recursos humanos para funcionar manualmente.

Amazon EKS admite dos productos de escalado automático:

Karpenter

Karpenter es un escalador automático de clústeres de Kubernetes flexible y de alto rendimiento que ayuda a mejorar la disponibilidad de las aplicaciones y la eficiencia de los clústeres. Karpenter lanza recursos de computación del tamaño correcto (por ejemplo, instancias de Amazon EC2) en respuesta a los cambios en la carga de las aplicaciones en menos de un minuto. Mediante la integración de Kubernetes con AWS, Karpenter puede aprovisionar recursos de computación justo a tiempo que cumplan con precisión los requisitos de su carga de trabajo. Karpenter aprovisiona automáticamente nuevos recursos de computación en función de los requisitos específicos de las cargas de trabajo de los clústeres. Estos incluyen requisitos de computación, almacenamiento, aceleración y programación. Amazon EKS admite clústeres que utilizan Karpenter, aunque Karpenter funciona con cualquier clúster de Kubernetes conforme. Para obtener más información, consulte la documentación de Karpenter.

Escalador automático del clúster

El Cluster Autoscaler de Kubernetes ajusta de forma automática el número de nodos del clúster cuando los pods fallan o se reprograman en otros nodos. El escalador automático de clústeres utiliza grupos de escalado automático. Para obtener más información, consulte Escalador automático de clústeres en AWS.

Administración de acceso

Obtenga información sobre cómo administrar el acceso a su clúster de Amazon EKS. El uso de Amazon EKS requiere saber cómo Kubernetes y AWS Identity and Access Management (AWS IAM) gestionan el control de acceso.

Esta sección incluye:

the section called “Concesión de acceso a las API de Kubernetes”: obtenga información sobre cómo permitir que las aplicaciones o los usuarios se autentiquen en la API de Kubernetes. Puede usar entradas de acceso, el aws-auth ConfigMap o un proveedor de OIDC externo.

the section called “Acceso a mi clúster con kubectl”: obtenga información sobre cómo configurar kubectl para que se comunique con su clúster de Amazon EKS. Use la CLI de AWS para crear un archivo kubeconfig.

the section called “Concesión de acceso a las cargas de trabajo a AWS”: obtenga información sobre cómo asociar una cuenta de servicio de Kubernetes con roles de AWS IAM. Puede usar los roles de IAM o de Pod Identity para las cuentas de servicio (IRSA).

Tareas comunes:

- Conceda a los desarrolladores acceso a la API de Kubernetes. Vea los recursos de Kubernetes en la AWS Management Console.
  - Solución: utilice las entradas de acceso para asociar los permisos de Kubernetes RBAC a los usuarios o roles de AWS IAM.
- Configure kubectl para que se comunique con un clúster de Amazon EKS mediante credenciales de AWS.
  - Solución: utilice la CLI de AWS para crear un archivo kubeconfig.
- Utilice un proveedor de identidad externo, como Ping Identity, para autenticar a los usuarios en la API de Kubernetes.
  - Solución: vincule un proveedor de OIDC externo.
- Conceda a las cargas de trabajo de su clúster de Kubernetes la capacidad de llamar a las API de AWS.
  - Solución: use Pod Identity para asociar un rol de AWS IAM a una cuenta de servicio de Kubernetes.

Introducción:

- Obtenga información sobre cómo funcionan las cuentas de servicio de Kubernetes.
- Revise el modelo de control de acceso basado en roles (RBAC) de Kubernetes
- Para obtener más información sobre la gestión del acceso a los recursos de AWS, consulte la Guía del usuario de AWS IAM. Como alternativa, haga una formación introductoria gratuita sobre el uso de AWS IAM.

Concesión de acceso a las API de Kubernetes

Su clúster tiene un punto de conexión de la API de Kubernetes. Kubectl usa esta API. Puede autenticarse en esta API mediante dos tipos de identidades:

- Una entidad principal de (IAM) (rol o usuario) de AWS Identity and Access Management: este tipo requiere autenticación ante IAM. Los usuarios pueden iniciar sesión en AWS como un usuario de IAM o con una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. Los usuarios solo pueden iniciar sesión con una identidad federada si su administrador previamente configuró la federación de identidades mediante los roles de IAM. Cuando los usuarios acceden a AWS mediante la federación, están asumiendo un rol de forma indirecta. Cuando los usuarios utilizan este tipo de identidad, usted:
  - Puede asignarles permisos de Kubernetes para que puedan trabajar con objetos de Kubernetes en su clúster. Para obtener más información sobre cómo asignar permisos a las entidades principales de IAM para que puedan acceder a los objetos de Kubernetes en su clúster, consulte Administración de entradas de acceso.
  - Puede asignarles permisos de IAM para que puedan trabajar con su clúster de Amazon EKS y sus recursos mediante la API de Amazon EKS, la AWS CLI, la AWS CloudFormation, la AWS Management Console o el `eksctl`. Para obtener más información, consulte Acciones definidas por Amazon Elastic Kubernetes Service en la Referencia de autorización de servicios.
  - Los nodos se unen al clúster asumiendo un rol de IAM. El acceso al clúster mediante las entidades principales de IAM está habilitado por el Autenticador de IAM de AWS para Kubernetes, que se ejecuta en el plano de control de Amazon EKS.
- Un usuario en su propio proveedor de OpenID Connect (OIDC): este tipo requiere la autenticación de su proveedor de OIDC. Para obtener más información acerca de cómo configurar su propio proveedor de OIDC con su clúster de Amazon EKS, consulte Autenticación de usuarios para el

clúster desde un proveedor de identidad de OpenID Connect. Cuando los usuarios utilizan este tipo de identidad, usted:

- Puede asignarles permisos de Kubernetes para que puedan trabajar con los objetos de Kubernetes en su clúster.
- No puede asignarles permisos de IAM para que puedan trabajar con su clúster de Amazon EKS y sus recursos mediante la API de Amazon EKS, la AWS CLI, la AWS CloudFormation, la AWS Management Console o el `eksctl`.

Puede utilizar ambos tipos de identidades con su clúster. El método de autenticación de IAM no se puede deshabilitar. El método de autenticación de OIDC es opcional.

Asociación de las identidades de IAM con los permisos de Kubernetes

El Autenticador de IAM de AWS para Kubernetes se instala en el plano de control del clúster. Permite a las entidades principales (IAM) (roles y usuarios) de AWS Identity and Access Management acceder a los recursos de Kubernetes en su clúster. Puede permitir que las entidades principales de IAM accedan a los objetos de Kubernetes en su clúster mediante uno de los siguientes métodos:

- Creación de entradas de acceso: si su clúster tiene la versión de la plataforma igual o posterior a la que se indica en la sección de Requisitos previos de la versión de Kubernetes de su clúster, le recomendamos que utilice esta opción.

  Utilice las entradas de acceso para administrar los permisos de Kubernetes de las entidades principales de IAM ajenos al clúster. Puede agregar y administrar el acceso al clúster mediante la API de EKS, la AWS Command Line Interface, los SDK de AWS, la AWS CloudFormation y la AWS Management Console. Esto significa que puede administrar los usuarios con las mismas herramientas con las que creó el clúster.

  Para empezar, siga los pasos de Configuración de las entradas de acceso y luego Migración de las entradas existentes de `aws-auth ConfigMap` a entradas de acceso.
- Añadir entradas al **`ConfigMap`** de **`aws-auth`**: si la versión de la plataforma de su clúster es anterior a la que aparece en la sección de Requisitos previos, debe utilizar esta opción. Si la versión de la plataforma de su clúster es igual o posterior a la versión de la plataforma que aparece en la sección de Requisitos previos de la versión de Kubernetes de su clúster y ha agregado entradas al `ConfigMap`, le recomendamos que migre esas entradas para acceder a las ellas. Sin embargo, no puede migrar las entradas que Amazon EKS haya agregado al `ConfigMap`, como las entradas para los roles de IAM utilizadas con los grupos de nodos administrados o los perfiles

de Fargate. Para obtener más información, consulte the section called “Concesión de acceso a las API de Kubernetes”.

- Si tiene que usar la opción de `ConfigMap` de `aws-auth`, puede añadir entradas al `ConfigMap` mediante el comando **`eksctl create iamidentitymapping`**. Para obtener más información, consulte Administrar usuarios y roles de IAM en la documentación de `eksctl`.

Establecimiento de nodos de autenticación del clúster

Cada clúster tiene un modo de autenticación. El modo de autenticación determina los métodos que puede utilizar para permitir que las entidades principales de IAM accedan a los objetos de Kubernetes en su clúster. Existen tres modos de autenticación.

> **Important**
>
> Una vez que se habilita el método de entrada de acceso, no se puede deshabilitar.
> Si el método `ConfigMap` no está habilitado durante la creación del clúster, no se podrá habilitar más adelante. Todos los clústeres creados antes de la introducción de las entradas de acceso tienen el método `ConfigMap` activado.

El `ConfigMap` de `aws-auth` dentro del clúster

Este es el modo de autenticación original de los clústeres de Amazon EKS. La entidad principal de IAM que creó el clúster es el usuario inicial que puede acceder al clúster con `kubectl`. El usuario inicial debe añadir otros usuarios a la lista en el `ConfigMap` de `aws-auth` y asignar los permisos que afecten a los demás usuarios del clúster. Estos otros usuarios no pueden administrar ni eliminar al usuario inicial, ya que no hay ninguna entrada en el `ConfigMap` que administrar.

Tanto el `ConfigMap` como las entradas de acceso

Con este modo de autenticación, puede utilizar ambos métodos para añadir las entidades principales de IAM al clúster. Tenga en cuenta que cada método almacena entradas independientes; por ejemplo, si agrega una entrada de acceso desde la AWS CLI, el `ConfigMap` de `aws-auth` no se actualiza.

Solo entradas de acceso

Con este modo de autenticación, puede utilizar la API de EKS, la AWS Command Line Interface, los SDK de AWS, la AWS CloudFormation y la AWS Management Console para administrar el acceso al clúster de las entidades principales de IAM.

Cada entrada de acceso tiene un tipo y puede utilizar la combinación de un enlace de acceso para limitar a la entidad principal a un espacio de nombres específico y una política de acceso para establecer políticas de permisos preconfiguradas y reutilizables. Como alternativa, puede usar el tipo STANDARD y los grupos de Kubernetes RBAC para asignar permisos personalizados.

Modo de autenticación	Métodos
Solo `ConfigMap` (`CONFIG_MAP`)	`aws-auth` `ConfigMap`
API de EKS y `ConfigMap` (`API_AND_CONFIG_MAP`)	acceder a las entradas de la API de EKS, la AWS Command Line Interface, los SDK de AWS, la AWS CloudFormation y la AWS Management Console, así como el `ConfigMap` de `aws-auth`.
Solo la API de EKS (`API`)	acceder a las entradas de la API de EKS, la AWS Command Line Interface, los SDK de AWS, la AWS CloudFormation y la AWS Management Console

Administración de entradas de acceso

Requisitos previos

- Familiaridad de con las opciones de acceso al clúster para su clúster de Amazon EKS. Para obtener más información, consulte Concesión de acceso a las API de Kubernetes.
- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS. Para utilizar las entradas de acceso y cambiar el modo de autenticación de un clúster, el clúster debe tener una versión de la plataforma igual o posterior a la que se indica en la siguiente tabla o una versión de Kubernetes posterior a las versiones que se muestran en la tabla.

Versión de Kubernetes	Versión de la plataforma
1.30	eks.2
1.29	eks.1
1.28	eks.6
1.27	eks.10
1.26	eks.11
1.25	eks.12
1.24	eks.15
1.23	eks.17

Para verificar su versión actual de Kubernetes y de la versión de la plataforma, reemplace *my-cluster* en el siguiente comando por el nombre de su clúster y luego ejecute el comando modificado: **aws eks describe-cluster --name *my-cluster* --query 'cluster.{"Kubernetes Version": version, "Platform Version": platformVersion}'**.

> **Important**
>
> Después de que Amazon EKS actualice el clúster a la versión de la plataforma que aparece en la tabla, Amazon EKS crea una entrada de acceso con permisos de administrador al clúster para la entidad principal de IAM que lo creó originalmente. Si no quiere que la entidad principal de IAM tenga permisos de administrador en el clúster, elimine la entrada de acceso que Amazon EKS creó.
>
> En el caso de los clústeres con versiones de la plataforma anteriores a las enumeradas en la tabla anterior, el creador del clúster es siempre el administrador del clúster. No se pueden eliminar los permisos de administrador del clúster del rol o usuario de IAM que creó el clúster.

- Una entidad principal de IAM con los siguientes permisos en el clúster:`CreateAccessEntry`, `ListAccessEntries`, `DescribeAccessEntry`, `DeleteAccessEntry` y

`UpdateAccessEntry`. Para obtener más información sobre los permisos de Amazon EKS, consulte las Acciones definidas por Amazon Elastic Kubernetes Service en la Referencia de autorizaciones del servicio.

- Una entidad principal de IAM existente para la que crear una entrada de acceso o una entrada de acceso existente para actualizar o eliminar.

Configuración de las entradas de acceso

Para empezar a utilizar las entradas de acceso, cambie el modo de autenticación del clúster a los modos `API_AND_CONFIG_MAP` o `API`. Esto añade la API para las entradas de acceso.

AWS Management Console

Cómo crear una entrada de acceso

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster en el que desea crear una entrada de acceso.
3. Elija la pestaña Acceso.
4. El Modo de autenticación muestra el modo de autenticación actual del clúster. Si el modo indica EKS API, ya puede añadir entradas de acceso y puede omitir los pasos restantes.
5. Elija Administrar acceso.
6. En Modo de autenticación del clúster, seleccione un modo con la EKS API. Tenga en cuenta que no puede volver a cambiar el modo de autenticación a un modo que elimine el EKS API y las entradas de acceso.
7. Elija Guardar cambios. Amazon EKS comienza a actualizar el clúster, el estado del clúster cambia a Updating y el cambio se registra en la pestaña Historial de actualizaciones.
8. Espere a que el estado del clúster vuelva a ser Active. Cuando el clúster esté Active, puede seguir los pasos que se indican en Creación de entradas de acceso para añadir acceso al clúster para las entidades principales de IAM.

AWS CLI

Requisito previo

La última versión de la AWS CLI v1 instalada y configurada en su dispositivo o AWS CloudShell. La AWS CLI v2 no admite nuevas características durante algunos días. Puede comprobar

su versión actual con `aws --version | cut -d / -f2 | cut -d ' ' -f1`. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con `aws configure` en la Guía del usuario de AWS Command Line Interface. La versión de la AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Installing AWS CLI to your home directory (Instalación de la en el directorio de inicio) en la Guía del usuario de AWS CloudShell.

1.

2. Ejecute el siguiente comando de la . Reemplace *`my-cluster`* por el nombre de su clúster. Si desea deshabilitar permanentemente el método `ConfigMap`, reemplace`API_AND_CONFIG_MAP` por `API`.

   Amazon EKS comienza a actualizar el clúster, el estado del clúster cambia a UPDATING y el cambio se registra en aws eks list-updates.

   ```
   aws eks update-cluster-config --name my-cluster --access-config
    authenticationMode=API_AND_CONFIG_MAP
   ```

3. Espere a que el estado del clúster vuelva a ser Active. Cuando el clúster esté Active, puede seguir los pasos que se indican en Creación de entradas de acceso para añadir acceso al clúster para las entidades principales de IAM.

Creación de entradas de acceso

Consideraciones

Antes de crear entradas de acceso, tenga en cuenta lo siguiente:

- Una entrada de acceso incluye el Nombre de recurso de Amazon (ARN) de una sola entidad principal de IAM existente. No se puede incluir una entidad principal de IAM en más de una entrada de acceso. Consideraciones adicionales para el ARN que especifique:
  - Las prácticas recomendadas de IAM sugieren acceder al clúster mediante roles de IAM que tengan credenciales a corto plazo, en lugar de usuarios de IAM que tengan credenciales a largo plazo. Para obtener más información, consulte Solicitar que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS mediante credenciales temporales en la Guía del usuario de IAM.

  - Si el ARN es para un rol de IAM, puede incluir una ruta. Los ARN de las entradas del `ConfigMap` de `aws-auth` no pueden incluir una ruta. Por ejemplo, su ARN puede ser `arn:aws:iam::111122223333:role/development/apps/my-role` o `arn:aws:iam::111122223333:role/my-role`.
  - Si el tipo de entrada de acceso es distinto a `STANDARD` (consulte la siguiente consideración sobre los tipos), el ARN debe estar en la misma Cuenta de AWS que el clúster. Si el tipo es `STANDARD`, el ARN puede estar en la misma Cuenta de AWS o en una cuenta diferente a la cuenta en la que se encuentra el clúster.
  - Después de crear la entrada de acceso, no se puede cambiar la entidad principal de IAM.
  - Si alguna vez elimina la entidad principal de IAM con este ARN, la entrada de acceso no se elimina automáticamente. Le recomendamos que elimine la entrada de acceso con un ARN para la entidad principal de IAM que desea eliminar. Si no elimina la entrada de acceso y vuelve a crear la entidad principal de IAM, la entrada de acceso no funcionará aunque tenga el mismo ARN. Esto se debe a que, aunque el ARN es el mismo para la entidad principal de IAM recreada, el `roleID` o `userID` (puede verlo con el comando `aws sts get-caller-identity` de la AWS CLI) es diferente para la entidad principal de IAM recreada que para la entidad principal de IAM original. Aunque no vea el `roleID` o el `userID` de la entidad principal de IAM para una entrada de acceso, Amazon EKS lo almacena junto con la entrada de acceso.
- Cada entrada de acceso tiene un tipo. Puede especificar `EC2 Linux` (para un rol de IAM utilizado con nodos autoadministrados de Linux o Bottlerocket), `EC2 Windows` (para un rol de IAM utilizado con nodos autoadministrados de Windows), `FARGATE_LINUX` (para un rol de IAM utilizado con AWS Fargate (Fargate)) o `STANDARD` como un tipo. Si no especifica ningún tipo, Amazon EKS establece automáticamente el tipo en `STANDARD`. No es necesario crear una entrada de acceso para un rol de IAM que se utiliza para un grupo de nodos administrados o un perfil de Fargate, ya que Amazon EKS agrega entradas para estos roles al `ConfigMap` de `aws-auth`, independientemente de la versión de la plataforma en la que se encuentre su clúster.

  Después de crear la entrada de acceso, no se puede cambiar el tipo.
- Si el tipo de entrada de acceso es `STANDARD`, puede especificar un nombre de usuario para la entrada de acceso. Si no especifica un valor para el nombre de usuario, Amazon EKS establece uno de los siguientes valores en función del tipo de entrada de acceso y de si la entidad principal de IAM que especificó es un rol de IAM o un usuario de IAM. A menos que tenga un motivo específico para especificar su propio nombre de usuario, le recomendamos que no especifique ninguno y deje que Amazon EKS lo genere automáticamente. Si especifica su propio nombre de usuario:

- No puede empezar con `system:`, `eks:`, `aws:`, `amazon:` o `iam:`.
- Si el nombre de usuario corresponde a un rol de IAM, le recomendamos que añada `{{SessionName}}` al final de este. Si añade `{{SessionName}}` a su nombre de usuario, este debe incluir dos puntos antes de {{SessionName}}. Cuando se asume este rol, el nombre de la sesión especificada al asumir el rol se transfiere automáticamente al clúster y aparece en los registros de CloudTrail. Por ejemplo, no puede tener un nombre de usuario como `john{{SessionName}}`. El nombre de usuario tendría que ser `:john{{SessionName}}` o `jo:hn{{SessionName}}`. Los dos puntos deben estar antes de `{{SessionName}}`. El nombre de usuario generado por Amazon EKS en la siguiente tabla incluye un ARN. Como un ARN incluye dos puntos, cumple con este requisito. Los dos puntos no son obligatorios si no incluye `{{SessionName}}` en su nombre de usuario.

Tipo de entidad principal de IAM	Tipo	Valor de nombre de usuario que Amazon EKS establece automáticamente
Usuario	`STANDARD`	El ARN del usuario. Ejemplo: `arn:aws:iam::` *`111122223333`* `:user/`*`my-user`*
Rol	`STANDARD`	El ARN de STS del rol cuando se asume. Amazon EKS agrega `{{SessionName}}` al rol. Ejemplo: `arn:aws:sts::` *`111122223333`* `:assumed-role/`*`my-role`*`/{{SessionName}}` Si el ARN del rol que especificó contenía una ruta, Amazon EKS la elimina del nombre de usuario generado.

Tipo de entidad principal de IAM	Tipo	Valor de nombre de usuario que Amazon EKS establece automáticamente
Rol	`EC2 Linux` o `EC2 Windows`	`system:node:{{EC2PrivateDNSName}}`
Rol	`FARGATE_LINUX`	`system:node:{{SessionName}}`

Puede cambiar el nombre de usuario después de crear la entrada de acceso.

- Si el tipo de entrada de acceso es `STANDARD` y desea utilizar la autorización RBAC de Kubernetes, puede añadir uno o más nombres de grupo a la entrada de acceso. Después de crear una entrada de acceso, puede añadir y eliminar nombres de grupos. Para que la entidad principal de IAM tenga acceso a los objetos de Kubernetes del clúster, debe crear y administrar los objetos de autorización basados en roles (RBAC) de Kubernetes. Cree objetos `RoleBinding` o `ClusterRoleBinding` de Kubernetes en el clúster que especifiquen el nombre del grupo como un `subject` para `kind: Group`. Kubernetes autoriza el acceso a la entidad principal de IAM a cualquier objeto del clúster que haya especificado en un objeto `Role` o `ClusterRole` de Kubernetes que también haya especificado en el `roleRef` del enlace. Si especifica nombres de grupo, le recomendamos que esté familiarizado con los objetos de autorizaciones basados en roles (RBAC) de Kubernetes. Para obtener más información, consulte Uso de la autorización de RBAC en la documentación de Kubernetes.

> **Important**
>
> Amazon EKS no confirma que ningún objeto RBAC de Kubernetes que exista en el clúster incluya alguno de los nombres de grupo que especifique.

En lugar de autorizar a Kubernetes para que la entidad principal de IAM acceda a los objetos de Kubernetes de su clúster, o además de ello, puede asociar las políticas de acceso de Amazon EKS a una entrada de acceso. Amazon EKS autoriza a las entidades principales de IAM a acceder a los objetos de Kubernetes del clúster con los permisos de la política de acceso. Puede limitar los permisos de una política de acceso a los espacios de nombres de Kubernetes que especifique. El uso de políticas de acceso no requiere que administre los objetos RBAC de Kubernetes. Para

obtener más información, consulte Asociación y desasociación de políticas de acceso a las entradas de acceso y desde las mismas.

- Si crea una entrada de acceso con el tipo `EC2 Linux` o `EC2 Windows`, la entidad principal de IAM que crea la entrada de acceso debe tener el permiso `iam:PassRole`. Para obtener más información, consulte Conceder permisos a un usuario para transferir un rol a un Servicio de AWS en la Guía del usuario de IAM.
- Al igual que el comportamiento de IAM estándar, la creación y las actualizaciones de las entradas de acceso son eventualmente uniformes y pueden tardar varios segundos en hacerse efectivas una vez que la llamada inicial a la API se haya completado con éxito. Debe diseñar sus aplicaciones teniendo en cuenta estos posibles retrasos. Le recomendamos que no incluya las creaciones o las actualizaciones de las entradas de acceso en las rutas de código de gran importancia y alta disponibilidad de su aplicación. En su lugar, realice los cambios de en otra rutina de inicialización o configuración que ejecute con menos frecuencia. Además, asegúrese de verificar que los cambios se han propagado antes de que los flujos de trabajo de producción dependan de ellos.
- Las entradas de acceso no admiten roles vinculados a servicios. No puede crear entradas de acceso en las que el ARN principal sea un rol vinculado al servicio. Puede identificar los roles vinculados al servicio por su ARN, que está en el formato `arn:aws:iam::*:role/aws-service-role/*`.

Puede crear una entrada de acceso mediante la AWS Management Console o la AWS CLI.

AWS Management Console

Cómo crear una entrada de acceso

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster en el que desea crear una entrada de acceso.
3. Elija la pestaña Acceso.
4. Elija Crear entrada de acceso.
5. En Entidad principal de IAM, seleccione un usuario o rol de IAM existente. Las prácticas recomendadas de IAM sugieren acceder al clúster mediante roles de IAM que tengan credenciales a corto plazo, en lugar de usuarios de IAM que tengan credenciales a largo plazo. Para obtener más información, consulte Solicitar que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS mediante credenciales temporales en la Guía del usuario de IAM.

6. En Tipo, si la entrada de acceso es para el rol de nodo utilizado para los nodos Amazon EC2 autoadministrados, seleccione EC2 Linux o EC2 Windows. De lo contrario, acepte el valor predeterminado (Estándar).
7. Si el Tipo que ha elegido es Estándar y desea especificar un Nombre de usuario, introdúzcalo.
8. Si el Tipo que ha elegido es Estándar y desea utilizar la autorización RBAC de Kubernetes para la entidad principal de IAM, especifique uno o más nombres para los Grupos. Si no especifica ningún nombre de grupo y desea utilizar la autorización de Amazon EKS, puede asociar una política de acceso en un paso posterior o después de crear la entrada de acceso.
9. (Opcional) En Etiquetas, asigne etiquetas a la entrada de acceso. Por ejemplo, para facilitar la búsqueda de todos los recursos con la misma etiqueta.
10. Elija Siguiente.
11. En la página Añadir política de acceso, si el tipo que ha elegido es Estándar y quiere que Amazon EKS autorice a la entidad principal de IAM a tener permisos para los objetos de Kubernetes de su clúster, complete los siguientes pasos. En caso contrario, elija Siguiente.

    a. En Nombre de la política, elija una política de acceso. No puede ver los permisos de las políticas de acceso, pero incluyen permisos similares a los de los objetos `ClusterRole` orientados al usuario de Kubernetes. Para obtener más información, consulte Roles orientados al usuario en la documentación de Kubernetes.

    b. Seleccione una de las siguientes opciones:

    - Clúster: elija esta opción si desea que Amazon EKS autorice a la entidad principal de IAM a tener los permisos de la política de acceso para todos los objetos de Kubernetes de su clúster.
    - Espacio de nombres de Kubernetes: elija esta opción si desea que Amazon EKS autorice a la entidad principal de IAM a tener los permisos de la política de acceso para todos los objetos de Kubernetes en un espacio de nombres específico de Kubernetes en su clúster. En Espacio de nombres, introduzca el nombre del espacio de nombres de Kubernetes en el clúster. Si quiere añadir espacios de nombres adicionales, seleccione Añadir nuevo espacio de nombres e ingrese el nombre del espacio de nombres.

    c. Si desea añadir políticas adicionales, seleccione Añadir política. Puede establecer el ámbito de cada política de forma diferente, pero puede añadir cada política solo una vez.

d. Elija Siguiente.

12. Revise la configuración de su entrada de acceso. Si algo parece incorrecto, seleccione Anterior para volver a repasar los pasos y corregir el error. Si la configuración es correcta, seleccione Crear.

AWS CLI

Requisito previo

La última versión de la AWS CLI v1 instalada y configurada en su dispositivo o AWS CloudShell. La AWS CLI v2 no admite nuevas características durante algunos días. Puede comprobar su versión actual con `aws --version | cut -d / -f2 | cut -d ' ' -f1`. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con `aws configure` en la Guía del usuario de AWS Command Line Interface. La versión de la AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Installing AWS CLI to your home directory (Instalación de la en el directorio de inicio) en la Guía del usuario de AWS CloudShell.

Cómo crear una entrada de acceso

Puede utilizar cualquiera de los siguientes ejemplos para crear entradas de acceso:

- Cree una entrada de acceso para un grupo de nodos autoadministrados de Amazon EC2 Linux. Sustituya *my-cluster* por el nombre de su clúster, *111122223333* por el ID de su Cuenta de AWS y *EKS-my-cluster-self-managed-ng-1* por el nombre del rol de IAM del nodo. Si su grupo de nodos es un grupo de nodos de Windows, sustituya *EC2_Linux* por **EC2_Windows**.

  ```
  aws eks create-access-entry --cluster-name my-cluster --principal-arn
   arn:aws:iam::111122223333:role/EKS-my-cluster-self-managed-ng-1 --type EC2_Linux
  ```

  No puede usar la opción `--kubernetes-groups` cuando especifica un tipo que no sea `STANDARD`. No puede asociar una política de acceso a esta entrada de acceso porque su tipo es un valor distinto a `STANDARD`.
- Cree una entrada de acceso que permita un rol de IAM que no se utilice para un grupo de nodos autoadministrado de Amazon EC2 con el que desee que Kubernetes autorice el acceso a su clúster. Reemplace *my-cluster* por el nombre de su clúster, *111122223333*

por el ID de su Cuenta de AWS y *my-role* por el nombre de su rol de IAM. Sustituya *Espectadores* por el nombre de un grupo que haya especificado en un objeto `RoleBinding` o `ClusterRoleBinding` de Kubernetes de su clúster.

```
aws eks create-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role --type STANDARD --user Viewers --
kubernetes-groups Viewers
```

- Cree una entrada de acceso que permita a un usuario de IAM autenticarse en su clúster. Este ejemplo se proporciona porque es posible, aunque las prácticas recomendadas de IAM sugieren acceder a su clúster mediante roles de IAM que tengan credenciales a corto plazo, en lugar de usuarios de IAM que tengan credenciales a largo plazo. Para obtener más información, consulte Solicitar que los usuarios humanos utilicen la federación con un proveedor de identidades para acceder a AWS mediante credenciales temporales en la Guía del usuario de IAM.

  ```
  aws eks create-access-entry --cluster-name my-cluster --principal-arn
   arn:aws:iam::111122223333:user/my-user --type STANDARD --username my-user
  ```

  Si desea que este usuario tenga más acceso a su clúster que los permisos en los roles de detección de la API de Kubernetes, debe asociar una política de acceso a la entrada de acceso, ya que la opción `--kubernetes-groups` no se utiliza. Para obtener más información, consulte Asociación y desasociación de políticas de acceso a las entradas de acceso y desde las mismas y roles de detección de la API en la documentación de Kubernetes.

Actualización de las entradas de acceso

Puede actualizar una entrada de acceso mediante la AWS Management Console o la AWS CLI.

AWS Management Console

Cómo actualizar una entrada de acceso

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster en el que desea crear una entrada de acceso.
3. Elija la pestaña Acceso.
4. Elija la entrada de acceso que desea actualizar.
5. Elija Editar.

6. En Nombre de usuario, puede cambiar el valor existente.
7. En Grupos, puede eliminar los nombres de los grupos existentes o añadir nuevos nombres de grupos. Si existen los siguientes nombres de grupos, no los elimine: system:nodes o system:bootstrappers. Si elimina estos grupos, puede provocar que el clúster no funcione correctamente. Si no especifica ningún nombre de grupo y desea utilizar la autorización de Amazon EKS, asocie una política de acceso en un paso posterior.
8. En Etiquetas, puede asignar etiquetas a la entrada de acceso. Por ejemplo, para facilitar la búsqueda de todos los recursos con la misma etiqueta. También puede eliminar las etiquetas existentes.
9. Elija Guardar cambios.
10. Si desea asociar una política de acceso a la entrada, consulte Asociación y desasociación de políticas de acceso a las entradas de acceso y desde las mismas.

AWS CLI

Requisito previo

La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

Cómo actualizar una entrada de acceso

Sustituya *`my-cluster`* por el nombre de su clúster, *`111122223333`* por el ID de su Cuenta de AWS y *`EKS-my-cluster-my-namespace-Viewers`* por el nombre del rol de IAM.

```
aws eks update-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/EKS-my-cluster-my-namespace-Viewers --kubernetes-
groups Viewers
```

No puede usar la opción de `--kubernetes-groups` si el tipo de entrada de acceso es un valor distinto a `STANDARD`. Tampoco puede asociar una política de acceso a una entrada de acceso de un tipo distinto a `STANDARD`.

Eliminar entradas de acceso

Si descubre que ha eliminado una entrada de acceso por error, siempre podrá volver a crearla. Si la entrada de acceso que va a eliminar está asociada a alguna política de acceso, las asociaciones se eliminarán automáticamente. No es necesario desasociar las políticas de acceso de una entrada de acceso antes de eliminarla.

Puede eliminar una entrada de acceso de mediante la AWS Management Console o la AWS CLI.

AWS Management Console

Cómo eliminar una entrada de acceso

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster del que desea eliminar una entrada de acceso.
3. Elija la pestaña Acceso.
4. En la lista de Entradas de acceso, elija la entrada de acceso que desea eliminar.
5. Elija Eliminar.
6. En el cuadro de diálogo de confirmación, elija Eliminar.

AWS CLI

Requisito previo

La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto

de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

Cómo eliminar una entrada de acceso

Sustituya *my-cluster* por el nombre de su clúster, *111122223333* por el ID de su Cuenta de AWS y *my-role* por el nombre del rol de IAM que ya no quiere que tenga acceso a su clúster.

```
aws eks delete-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role
```

Asociación y desasociación de políticas de acceso a las entradas de acceso y desde las mismas

Puede asignar una o más políticas de acceso a las entradas de acceso del tipo `STANDARD`. Amazon EKS concede automáticamente a los demás tipos de entradas de acceso los permisos necesarios para funcionar correctamente en el clúster. Las políticas de acceso de Amazon EKS incluyen permisos de Kubernetes, no permisos de IAM. Antes de asociar una política de acceso a una entrada de acceso, asegúrese de estar familiarizado con los permisos de Kubernetes incluidos en cada política de acceso. Para obtener más información, consulte Permisos de política de acceso. Si ninguna de las políticas de acceso cumple con sus requisitos, no asocie una política de acceso a una entrada de acceso. En su lugar, especifique uno o más nombres de grupo para la entrada de acceso, y cree y administre objetos de control de acceso basados en roles de Kubernetes. Para obtener más información, consulte Creación de entradas de acceso.

Requisitos previos

- Una entrada de acceso existente. Para crear uno, consulte Creación de entradas de acceso.
- Un rol o usuario de AWS Identity and Access Management con los siguientes permisos: `ListAccessEntries`, `DescribeAccessEntry`, `UpdateAccessEntry`, `ListAccessPolicies`, `AssociateAccessPolicy` y `DisassociateAccessPolicy`. Para obtener más información, consulte Acciones definidas por Amazon Elastic Kubernetes Service en la Referencia de autorización de servicios.

Antes de asociar políticas de acceso con entradas de acceso, tenga en cuenta los siguientes requisitos:

- Puede asociar varias políticas de acceso a cada entrada de acceso, pero solo puede asociar cada política a una entrada de acceso una vez. Si asocia varias políticas de acceso, la entidad principal de IAM de la entrada de acceso tendrá todos los permisos incluidos en todas las políticas de acceso asociadas.
- Puede limitar una política de acceso a todos los recursos de un clúster o especificar el nombre de uno o más espacios de nombres de Kubernetes. Puede utilizar caracteres comodín para el nombre de un espacio de nombres. Por ejemplo, si desea limitar una política de acceso a todos los espacios de nombres que comiencen con `dev-`, puede especificar `dev-*` como nombre de un espacio de nombres. Asegúrese de que los espacios de nombres existan en su clúster y de que la ortografía coincida con el nombre real del espacio de nombres del clúster. Amazon EKS no confirma la ortografía ni la existencia de los espacios de nombres del clúster.
- Puede cambiar el alcance del acceso de una política de acceso después de asociarla a una entrada de acceso. Si limitó la política de acceso a los espacios de nombres de Kubernetes, puede añadir y eliminar espacios de nombres para la asociación, según sea necesario.
- Si asocia una política de acceso a una entrada de acceso que también tenga nombres de grupo especificados, la entidad principal de IAM tendrá todos los permisos en todas las políticas de acceso asociadas. También tiene todos los permisos en cualquier objeto de `Role` o `ClusterRole` de Kubernetes que estén especificados en cualquier objeto de `Role` y `RoleBinding` de Kubernetes que especifican los nombres de grupo.
- Si ejecuta el comando `kubectl auth can-i --list`, no verá ningún permiso de Kubernetes asignado por las políticas de acceso asociadas a una entrada de acceso para la entidad principal de IAM que esté utilizando cuando ejecute el comando. El comando solo muestra los permisos de Kubernetes si los concedió en los objetos de `Role` o `ClusterRole` de Kubernetes que vinculó a los nombres de grupo o al nombre de usuario que especificó para una entrada de acceso.
- Si se hace pasar por un usuario o un grupo de Kubernetes al interactuar con los objetos de Kubernetes del clúster, por ejemplo, al utilizar el comando `kubectl` con **`--as`** ***`username`*** o **`--as-group`** ***`group-name`***, está obligando el uso de una autorización RBAC de Kubernetes. Como resultado, la entidad principal de IAM no tiene permisos asignados por ninguna política de acceso asociada a la entrada de acceso. Los únicos permisos de Kubernetes que tiene el usuario o grupo al que se hace pasar la entidad principal de IAM son los permisos de Kubernetes que usted concedió en los objetos de `Role` o `ClusterRole` de Kubernetes que vinculó a los nombres de grupo o al nombre de usuario. Para que su entidad principal de IAM tenga los permisos de las políticas de acceso asociadas, no se haga pasar por un usuario o un grupo de Kubernetes. La entidad principal de IAM también seguirá teniendo todos los permisos que le haya concedido en los objetos de `Role` o `ClusterRole` de Kubernetes que usted haya vinculado con los nombres

de grupo o el nombre de usuario que especificó para la entrada de acceso. Para obtener más información, consulte Suplantación del usuario en la documentación de Kubernetes.

Puede asociar una política de acceso a una entrada de acceso mediante la AWS Management Console o la AWS CLI.

AWS Management Console

Cómo asociar una política de acceso a una entrada de acceso mediante la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster que tenga una entrada de acceso a la que desea asociar una política de acceso.
3. Elija la pestaña Acceso.
4. Si el tipo de entrada de acceso es Estándar, puede asociar o desasociar las políticas de acceso de Amazon EKS. Si el tipo de entrada de acceso no es Estándar, entonces esta opción no está disponible.
5. Elija Asociar política de acceso.
6. En Nombre de la política, seleccione la política con los permisos que desea que tenga la entidad principal de IAM. Para ver los permisos incluidos en cada política, consulte Permisos de política de acceso.
7. En Alcance del acceso, elija un alcance del acceso. Si elige Clúster, los permisos en la política de acceso se otorgan a la entidad principal de IAM para los recursos de todos los espacios de nombres de Kubernetes. Si elige espacio de nombres de Kubernetes, luego puede elegir Añadir nuevo espacio de nombres. En el campo Espacio de nombres que aparece, puede introducir el nombre de un espacio de nombres de Kubernetes del clúster. Si quiere que la entidad principal de IAM tenga los permisos en varios espacios de nombres, puede introducir varios espacios de nombres.
8. Seleccione Añadir política de acceso.

AWS CLI

Requisito previo

La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

Cómo asociar una política de acceso a una entrada de acceso

1. Ver las políticas de acceso disponibles.

```
aws eks list-access-policies --output table
```

Un ejemplo de salida sería el siguiente.

```
-------------------------------------------------------------------------------------------
|                                        ListAccessPolicies
                        |
+-----------------------------------------------------------------------------------------
+
||                                          accessPolicies
                       ||
|+-------------------------------------------------------------------
+------------------------------+|
||                                 arn                                |
     name              ||
|+-------------------------------------------------------------------
+------------------------------+|
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy       |
 AmazonEKSAdminPolicy         ||
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSClusterAdminPolicy |
 AmazonEKSClusterAdminPolicy  ||
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy        |
 AmazonEKSEditPolicy          ||
||  arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy        |
 AmazonEKSViewPolicy          ||
```

```
|+-------------------------------------------------------------------
+------------------------------+|
```

Para ver los permisos incluidos en cada política, consulte Permisos de política de acceso.

2. Ver las entradas de acceso existentes. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks list-access-entries --cluster-name my-cluster
```

Un ejemplo de salida sería el siguiente.

```
{
    "accessEntries": [
        "arn:aws:iam::111122223333:role/my-role",
        "arn:aws:iam::111122223333:user/my-user"
    ]
}
```

3. Asocie una política de acceso a una entrada de acceso. El siguiente ejemplo asocia la política de acceso de AmazonEKSViewPolicy a una entrada de acceso. Siempre que el rol de IAM *my-role* intente acceder a los objetos de Kubernetes del clúster, Amazon EKS autorizará al rol a usar los permisos en la política para acceder únicamente a los objetos de Kubernetes en los espacios de nombres *my-namespace1* y *my-namespace2* de Kubernetes. Reemplace *my-cluster* por el nombre de su clúster, *111122223333* con su ID de Cuenta de AWS y *my-role* por el nombre del rol de IAM para el que desea que Amazon EKS autorice el acceso a los objetos del clúster de Kubernetes.

```
aws eks associate-access-policy --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role \
    --access-scope type=namespace,namespaces=my-namespace1,my-namespace2 --
policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy
```

Si desea que la entidad principal de IAM tenga los permisos en todo el clúster, reemplace **type=namespace,namespaces=*my-namespace1*,*my-namespace2*** con **type=cluster**. Si desea asociar varias políticas de acceso a la entrada de acceso, ejecute el comando varias veces, cada una con una política de acceso única. Cada política de acceso asociada tiene su propio alcance.

> **Note**
>
> Si más adelante desea cambiar el alcance de una política de acceso asociada, vuelva a ejecutar el comando anterior con el nuevo alcance. Por ejemplo, si quisiera eliminar *my-namespace2*, volvería a ejecutar el comando usando solamente **type=namespace,namespaces=*my-namespace1***. Si quisiera cambiar el alcance de **namespace** a **cluster**, volvería a ejecutar el comando usando **type=cluster** y eliminando **type=namespace,namespaces=*my-namespace1*,*my-namespace2***.

Cómo desasociar una política de acceso de una entrada de acceso

1. Determine qué políticas de acceso están asociadas a una entrada de acceso.

```
aws eks list-associated-access-policies --cluster-name my-cluster --principal-
arn arn:aws:iam::111122223333:role/my-role
```

Un ejemplo de salida sería el siguiente.

```
{
    "clusterName": "my-cluster",
    "principalArn": "arn:aws:iam::111122223333",
    "associatedAccessPolicies": [
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-
policy/AmazonEKSViewPolicy",
            "accessScope": {
                "type": "cluster",
                "namespaces": []
            },
            "associatedAt": "2023-04-17T15:25:21.675000-04:00",
            "modifiedAt": "2023-04-17T15:25:21.675000-04:00"
        },
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-
policy/AmazonEKSAdminPolicy",
            "accessScope": {
                "type": "namespace",
                "namespaces": [
                    "my-namespace1",
```

```
                    "my-namespace2"
                ]
            },
            "associatedAt": "2023-04-17T15:02:06.511000-04:00",
            "modifiedAt": "2023-04-17T15:02:06.511000-04:00"
        }
    ]
}
```

En el ejemplo anterior, la entidad principal de IAM para esta entrada de acceso tiene permisos de visualización en todos los espacios de nombres del clúster y permisos de administrador en dos espacios de nombres de Kubernetes.

2. Desasocie una política de acceso de una entrada de acceso. En este ejemplo, la política de `AmazonEKSAdminPolicy` está disociada de una entrada de acceso. Sin embargo, la entidad principal de IAM conserva los permisos de la política de acceso de `AmazonEKSViewPolicy` para los objetos en los espacios de nombres *my-namespace1* y *my-namespace2*, ya que esa política de acceso no está disociada de la entrada de acceso.

```
aws eks disassociate-access-policy --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/my-role \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy
```

Permisos de política de acceso

Las políticas de acceso incluyen `rules` que contienen (permisos) de `verbs` y `resources` de Kubernetes. Las políticas de acceso no incluyen los permisos ni los recursos de IAM. Al igual que los objetos de `Role` y `ClusterRole` de Kubernetes, las políticas de acceso solo incluyen `rules` de `allow`. No puede modificar el contenido de una política de acceso. No puede crear sus propias políticas de acceso. Si los permisos de las políticas de acceso no satisfacen sus necesidades, cree objetos RBAC de Kubernetes y especifique los nombres de grupo para las entradas de acceso. Para obtener más información, consulte Creación de entradas de acceso. Los permisos contenidos en las políticas de acceso son similares a los permisos de los roles del clúster orientados a los usuarios de Kubernetes. Para obtener más información, consulte Roles orientados a los usuarios en la documentación de Kubernetes.

Elija cualquier política de acceso para ver su contenido. Cada fila de cada tabla en cada política de acceso es una regla independiente.

AmazonEKSAdminPolicy

Esta política de acceso incluye permisos que otorgan a una entidad principal de IAM la mayoría de los permisos a los recursos. Cuando se asocia a una entrada de acceso, su alcance de acceso suele ser uno o más espacios de nombres de Kubernetes. Si desea que una entidad principal de IAM tenga acceso de administrador a todos los recursos de su clúster, asocie la política de acceso de AmazonEKSClusterAdminPolicy a su entrada de acceso.

ARN: `arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy`

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
apps	daemonsets , deployments , deployments/rollback , deployments/scale , replicasets , replicasets/scale, statefulsets , statefulsets/scale	create, delete, deletecollection , patch, update
apps	controllerrevisions , daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , replicasets , replicasets/scale , replicasets/status , statefulsets , statefulsets/scale , statefulsets/status	get, list, watch
authorization.k8s.io	localsubjectaccessreviews	create
autoscaling	horizontalpodautoscalers	create, delete, deletecollection , patch, update

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
autoscaling	horizontalpodautoscalers , horizontalpodautoscalers/status	get, list, watch
batch	cronjobs, jobs	create, delete, deletecollection , patch, update
batch	cronjobs, cronjobs/status , jobs, jobs/status	get, list, watch
discovery.k8s.io	endpointslices	get, list, watch
extensions	daemonsets , deployments , deployments/rollback , deployments/scale , ingresses , networkpolicies , replicasets , replicasets/scale , replicationcontrollers/scale	create, delete, deletecollection , patch, update
extensions	daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , ingresses , ingresses/status , networkpolicies , replicasets , replicasets/scale , replicasets/status , replicationcontrollers/scale	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
networking.k8s.io	ingresses , ingresses /status , networkpo licies	get, list, watch
networking.k8s.io	ingresses , networkpo licies	create, delete, deletecol lection , patch, update
policy	poddisruptionbudgets	create, delete, deletecol lection , patch, update
policy	poddisruptionbudgets , poddisruptionbudge ts/status	get, list, watch
rbac.authorization .k8s.io	rolebindings , roles	create, delete, deletecol lection , get, list, patch, update, watch
	configmaps , endpoints , persistentvolumecl aims , persisten tvolumeclaims/stat us , pods, replicati oncontrollers , replicationcontrol lers/scale , serviceac counts , services, services/status	get,list, watch
	pods/attach , pods/exec , pods/portforward , pods/proxy , secrets, services/proxy	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
	configmaps , events, persistentvolumeclaims , replicationcontrollers , replicationcontrollers/scale , secrets, serviceaccounts , services, services/proxy	create, delete, deletecollection , patch, update
	pods, pods/attach , pods/exec , pods/portforward , pods/proxy	create, delete, deletecollection , patch, update
	serviceaccounts	impersonate
	bindings, events, limitranges , namespaces/status , pods/log, pods/status , replicationcontrollers/status , resourcequotas , resourcequotas/status	get, list, watch
	namespaces	get,list, watch

AmazonEKSClusterAdminPolicy

Esta política de acceso incluye permisos que permiten a un administrador de la entidad principal de IAM acceder a un clúster. Cuando se asocia a una entrada de acceso, su alcance de acceso suele ser el clúster, en lugar de un espacio de nombres de Kubernetes. Si desea que una entidad principal de IAM tenga un alcance administrativo más limitado, considere la posibilidad de asociar la política de acceso de AmazonEKSAdminPolicy a su entrada de acceso.

ARN: `arn:aws:eks::aws:cluster-access-policy/AmazonEKSClusterAdminPolicy`

Grupos de la API de Kubernetes	Kubernetes nonResourceURLs	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
*		*	*
	*		*

AmazonEKSAdminViewPolicy

Esta política de acceso incluye permisos que permiten a una entidad principal de IAM acceder para enumerar/visualizar todos los recursos en un clúster. Tenga en cuenta que esto incluye los secretos de Kubernetes.

ARN: `arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminViewPolicy`

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
*	*	`get, list, watch`

AmazonEKSEditPolicy

Esta política de acceso incluye permisos que permiten a una entidad principal de IAM editar la mayoría de los recursos de Kubernetes.

ARN: `arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy`

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
`apps`	`daemonsets` , `deployments` , `deployments/rollback` , `deployments/scale` , `replicasets` , `replicasets/`	`create`, `delete`, `deletecollection` , `patch`, `update`

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
	scale, statefulsets , statefulsets/scale	
apps	controllerrevisions , daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , replicasets , replicasets/scale , replicasets/status , statefulsets , statefulsets/scale , statefulsets/status	get, list, watch
autoscaling	horizontalpodautoscalers , horizontalpodautoscalers/status	get, list, watch
autoscaling	horizontalpodautoscalers	create, delete, deletecollection , patch, update
batch	cronjobs, jobs	create, delete, deletecollection , patch, update
batch	cronjobs, cronjobs/status , jobs, jobs/status	get, list, watch
discovery.k8s.io	endpointslices	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
extensions	daemonsets , deployments , deployments/rollback , deployments/scale , ingresses , networkpolicies , replicasets , replicasets/scale , replicationcontrollers/scale	create, delete, deletecollection , patch, update
extensions	daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , ingresses , ingresses/status , networkpolicies , replicasets , replicasets/scale , replicasets/status , replicationcontrollers/scale	get, list, watch
networking.k8s.io	ingresses , networkpolicies	create, delete, deletecollection , patch, update
networking.k8s.io	ingresses , ingresses/status , networkpolicies	get, list, watch
policy	poddisruptionbudgets	create, delete, deletecollection , patch, update
policy	poddisruptionbudgets , poddisruptionbudgets/status	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
	namespaces	get, list, watch
	pods/attach , pods/exec , pods/portforward , pods/proxy , secrets, services/proxy	get, list, watch
	serviceaccounts	impersonate
	pods, pods/attach , pods/exec , pods/portforward , pods/proxy	create, delete, deletecollection , patch, update
	configmaps , events, persistentvolumeclaims , replicationcontrollers , replicationcontrollers/scale , secrets, serviceaccounts , services, services/proxy	create, delete, deletecollection , patch, update
	configmaps , endpoints , persistentvolumeclaims , persistentvolumeclaims/status , pods, replicationcontrollers , replicationcontrollers/scale , serviceaccounts , services, services/status	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
	bindings, events, limitranges , namespaces/status , pods/log, pods/status , replicationcontrollers/status , resourcequotas , resourcequotas/status	get, list, watch

AmazonEKSViewPolicy

Esta política de acceso incluye permisos que permiten a una entidad principal de IAM ver la mayoría de los recursos de Kubernetes.

ARN: arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
apps	controllerrevisions , daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , replicasets , replicasets/scale , replicasets/status , statefulsets , statefulsets/scale , statefulsets/status	get, list, watch
autoscaling	horizontalpodautoscalers , horizonta	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
	lpodautoscalers/status	
batch	cronjobs, cronjobs/status , jobs, jobs/status	get, list, watch
discovery.k8s.io	endpointslices	get, list, watch
extensions	daemonsets , daemonsets/status , deployments , deployments/scale , deployments/status , ingresses , ingresses/status , networkpolicies , replicasets , replicasets/scale , replicasets/status , replicationcontrollers/scale	get, list, watch
networking.k8s.io	ingresses , ingresses/status , networkpolicies	get, list, watch
policy	poddisruptionbudgets , poddisruptionbudgets/status	get, list, watch

Grupos de la API de Kubernetes	Recursos de Kubernetes	Verbos de Kubernetes (permisos)
	`configmaps` , `endpoints` , `persistentvolumeclaims` , `persistentvolumeclaims/status` , `pods`, `replicationcontrollers` , `replicationcontrollers/scale` , `serviceaccounts` , `services`, `services/status`	`get`, `list`, `watch`
	`bindings`, `events`, `limitranges` , `namespaces/status` , `pods/log`, `pods/status` , `replicationcontrollers/status` , `resourcequotas` , `resourcequotas/status`	`get`, `list`, `watch`
	`namespaces`	`get`, `list`, `watch`

Actualizaciones de la política de acceso

Vea detalles sobre las actualizaciones de las políticas de acceso desde su introducción. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la Página de historial de documentos de Amazon EKS.

Cambio	Descripción	Fecha
Add `AmazonEKSAdminViewPolicy`	Agregue una nueva política para ampliar el acceso a las vistas, incluidos recursos como Secretos.	23 de abril de 2024

Cambio	Descripción	Fecha
Se introdujeron políticas de acceso.	Amazon EKS introdujo políticas de acceso.	29 de mayo de 2023

Migración de las entradas existentes de `aws-auth ConfigMap` a entradas de acceso

Si ha agregado entradas al `ConfigMap` de `aws-auth` en su clúster, le recomendamos que cree entradas de acceso para las entradas existentes en su `ConfigMap` de `aws-auth`. Después de crear las entradas de acceso, puede eliminarlas de su `ConfigMap`. No puede asociar políticas de acceso a las entradas del `ConfigMap` de `aws-auth`. Si desea asociar políticas de acceso a sus entidades principales de IAM, cree entradas de acceso.

> **Important**
>
> No elimine las entradas existentes del `ConfigMap` de `aws-auth` que Amazon EKS creó al añadir un grupo de nodos administrados o un perfil de Fargate a su clúster. Si elimina las entradas que Amazon EKS creó en el `ConfigMap`, el clúster no funcionará correctamente. Sin embargo, puede eliminar cualquier entrada de los grupos de nodos autoadministrados una vez que haya creado las entradas de acceso para ellos.

Requisitos previos

- Familiaridad con las entradas de acceso y las políticas de acceso. Para obtener más información, consulte Administración de entradas de acceso y Asociación y desasociación de políticas de acceso a las entradas de acceso y desde las mismas.
- Un clúster existente con una versión de la plataforma que es igual o posterior a las versiones enumeradas en los requisitos previos del tema Permitir el acceso a usuarios o roles de IAM a objetos de Kubernetes en su clúster de Amazon EKS.
- La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.
- Permisos de Kubernetes para modificar el `ConfigMap` de `aws-auth` en el espacio de nombres `kube-system`.

- Un rol o usuario de AWS Identity and Access Management con los siguientes permisos: `CreateAccessEntry` y `ListAccessEntries`. Para obtener más información, consulte Acciones definidas por Amazon Elastic Kubernetes Service en la Referencia de autorizaciones del servicio.

Cómo migrar una entrada de su **`aws-auth ConfigMap`** a una entrada de acceso

1. Vea las entradas existentes en su `aws-auth ConfigMap`. Reemplace *`my-cluster`* por el nombre de su clúster.

   ```
   eksctl get iamidentitymapping --cluster my-cluster
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   ARN
                USERNAME                                GROUPS
                                ACCOUNT
   arn:aws:iam::111122223333:role/EKS-my-cluster-Admins
                Admins                                  system:masters
   arn:aws:iam::111122223333:role/EKS-my-cluster-my-namespace-Viewers
                my-namespace-Viewers                    Viewers
   arn:aws:iam::111122223333:role/EKS-my-cluster-self-managed-ng-1
                                system:node:{{EC2PrivateDNSName}}
    system:bootstrappers,system:nodes
   arn:aws:iam::111122223333:user/my-user
                my-user
   arn:aws:iam::111122223333:role/EKS-my-cluster-fargateprofile1
                                system:node:{{SessionName}}
    system:bootstrappers,system:nodes,system:node-proxier
   arn:aws:iam::111122223333:role/EKS-my-cluster-managed-ng
                                system:node:{{EC2PrivateDNSName}}
    system:bootstrappers,system:nodes
   ```

2. Cree entradas de acceso para cualquiera de las entradas del `ConfigMap` que haya creado y que se mostraron en el resultado anterior. Al crear las entradas de acceso, asegúrese de especificar los mismos valores para `ARN`, `USERNAME`, `GROUPS` y `ACCOUNT` que aparecen en el resultado. En el resultado de ejemplo, crearía entradas de acceso para todas las entradas, excepto las dos últimas, ya que Amazon EKS creó esas entradas para un perfil de Fargate y un grupo de nodos administrados.

3. Elimine las entradas del `ConfigMap` para cualquier entrada de acceso que haya creado. Si no elimina la entrada del `ConfigMap`, la configuración de la entrada de acceso para el ARN de entidad principal de IAM anula la entrada del `ConfigMap`. Reemplace *`111122223333`* por su ID de Cuenta de AWS y *`EKS-my-cluster-my-namespace-Viewers`* por el nombre del rol en la entrada de su `ConfigMap`. Si la entrada que va a eliminar es para un usuario de IAM y no para un rol de IAM, reemplace el **`role`** por el **`user`** y *`EKS-my-cluster-my-namespace-Viewers`* por el nombre de usuario.

```
eksctl delete iamidentitymapping --arn arn:aws:iam::111122223333:role/EKS-my-cluster-my-namespace-Viewers --cluster my-cluster
```

Habilitación del acceso principal de IAM al clúster

> **Important**
>
> `aws-auth` ConfigMap se ha quedado obsoleto. El método recomendado para administrar el acceso a las API de Kubernetes es Entradas de acceso.

El acceso al clúster mediante las entidades principales de IAM está habilitado por el AWSAutenticador de IAM deKubernetes, que se ejecuta en el plano de control de Amazon EKS. El autenticador obtiene la información de la configuración del `ConfigMap` de `aws-auth`. Para todos las configuraciones del `ConfigMap` de `aws-auth`, consulte el formato de configuración completa en GitHub.

Agregar las entidades principales de IAM al clúster de Amazon EKS

Cuando se crea un clúster de Amazon EKS, la entidad de IAM que crea el clúster recibe permisos de `system:masters` de forma automática en la configuración del role-based access control (RBAC, control de acceso basado en roles) del clúster en el plano de control de Amazon EKS. Esta entidad principal no aparece en ninguna configuración visible, así que asegúrese de realizar un seguimiento de la entidad principal que creó el clúster originalmente. Para conceder a entidades principales de IAM la capacidad de interactuar con el clúster, edite el `ConfigMap` de `aws-auth` dentro de Kubernetes y cree un `rolebinding` de Kubernetes o `clusterrolebinding` con el nombre de un `group` que especifique en el `ConfigMap` de`aws-auth`.

> **Note**
>
> Para obtener más información sobre la configuración del control de acceso basado en roles (RBAC) de Kubernetes, consulte Uso de la autorización de RBAC en la documentación de Kubernetes.

Para agregar una entidad principal de IAM a un clúster de Amazon EKS, haga lo siguiente:

1. Determine qué credenciales de `kubectl` utiliza para obtener acceso al clúster. En la computadora, puede ver qué credenciales de `kubectl` utiliza con el siguiente comando. Reemplace ***~/.kube/config*** por la ruta al archivo `kubeconfig` si no utiliza la ruta predeterminada.

   ```
   cat ~/.kube/config
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   [...]
   contexts:
   - context:
       cluster: my-cluster.region-code.eksctl.io
       user: admin@my-cluster.region-code.eksctl.io
     name: admin@my-cluster.region-code.eksctl.io
   current-context: admin@my-cluster.region-code.eksctl.io
   [...]
   ```

   En el salida de ejemplo anterior, se configuran las credenciales de un usuario denominado *`admin`* para un clúster denominado *`my-cluster`*. Si este es el usuario que creó el clúster, entonces ya tiene acceso a él. Si no es el usuario el que creó el clúster, deberá completar los pasos restantes para habilitar el acceso al clúster para otras entidades principales de IAM. Según las prácticas recomendadas de IAM, se recomienda conceder permisos a los roles en lugar de a los usuarios. Puede ver qué otras entidades principales tienen acceso actualmente al clúster con el siguiente comando:

   ```
   kubectl describe -n kube-system configmap/aws-auth
   ```

   Un ejemplo de salida sería el siguiente.

```
Name:         aws-auth
Namespace:    kube-system
Labels:       <none>
Annotations:  <none>

Data
====
mapRoles:
----
- groups:
  - system:bootstrappers
  - system:nodes
  rolearn: arn:aws:iam::111122223333:role/my-node-role
  username: system:node:{{EC2PrivateDNSName}}


BinaryData
====

Events:  <none>
```

El ejemplo anterior es un valor predeterminado `aws-auth` `ConfigMap`. Solo el rol de la instancia de nodos tiene acceso al clúster.

2. Asegúrese de tener `roles` y `rolebindings` o `clusterroles` y `clusterrolebindings` existentes de Kubernetes a los que pueda asignar entidades principales de IAM. Para obtener más información sobre estos recursos, consulte Utilización de la autorización de RBAC en la documentación de Kubernetes.

   1. Vea sus `roles` o `clusterroles` existentes de Kubernetes. Los `Roles` están asignados a un `namespace`, pero los `clusterroles` se ajustan al clúster.

      ```
      kubectl get roles -A
      ```

      ```
      kubectl get clusterroles
      ```

   2. Consulte los detalles de cualquier `role` o `clusterrole` devuelto en la salida anterior y confirme que tiene los permisos (`rules`) que desea que las entidades principales de IAM tengan en el clúster.

Reemplace *role-name* con un nombre de `role` devuelto en el resultado del comando anterior. Reemplace *kube-system* con el espacio de nombres del `role`.

```
kubectl describe role role-name -n kube-system
```

Reemplace *cluster-role-name* con un nombre de `clusterrole` devuelto en el resultado del comando anterior.

```
kubectl describe clusterrole cluster-role-name
```

3. Vea sus `rolebindings` o `clusterrolebindings` existentes de Kubernetes. Los `Rolebindings` están asignados a un `namespace`, pero los `clusterrolebindings` se ajustan al clúster.

```
kubectl get rolebindings -A
```

```
kubectl get clusterrolebindings
```

4. Vea los detalles de cualquier `rolebinding` o `clusterrolebinding` y confirme que tiene un `role` o `clusterrole` del paso anterior enumerado como `roleRef` y un nombre de grupo enumerado para `subjects`.

   Reemplace *role-binding-name* con un nombre de `rolebinding` devuelto en el resultado del comando anterior. Reemplace *kube-system* con el `namespace` de la `rolebinding`.

```
kubectl describe rolebinding role-binding-name -n kube-system
```

   Un ejemplo de salida sería el siguiente.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eks-console-dashboard-restricted-access-role-binding
  namespace: default
subjects:
- kind: Group
  name: eks-console-dashboard-restricted-access-group
  apiGroup: rbac.authorization.k8s.io
```

```
roleRef:
  kind: Role
  name: eks-console-dashboard-restricted-access-role
  apiGroup: rbac.authorization.k8s.io
```

Reemplace *cluster-role-binding-name* con un nombre de `clusterrolebinding` devuelto en el resultado del comando anterior.

```
kubectl describe clusterrolebinding cluster-role-binding-name
```

Un ejemplo de salida sería el siguiente.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks-console-dashboard-full-access-binding
subjects:
- kind: Group
  name: eks-console-dashboard-full-access-group
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: eks-console-dashboard-full-access-clusterrole
  apiGroup: rbac.authorization.k8s.io
```

3. Edite el `ConfigMap` de `aws-auth`. Puede utilizar una herramienta como `eksctl` para actualizar el `ConfigMap` o puede actualizarlo manualmente editándolo.

> **Important**
>
> Recomendamos utilizar `eksctl`, u otra herramienta, para editar el `ConfigMap`. Para obtener información acerca de otras herramientas que puede utilizar, consulte Utilice herramientas para realizar cambios en el `aws-auth` `ConfigMap` en las guías de prácticas recomendadas de Amazon EKS. Un formato incorrecto de `aws-auth` `ConfigMap` puede provocar que pierda el acceso a su clúster.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

1. Vea las asignaciones actuales en la `ConfigMap`. Reemplace *`my-cluster`* por el nombre del clúster. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster.

```
eksctl get iamidentitymapping --cluster my-cluster --region=region-code
```

   Un ejemplo de salida sería el siguiente.

```
ARN
                    USERNAME                                GROUPS
              ACCOUNT
arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
NodeInstanceRole-1XLS7754U3ZPA     system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
```

2. Agregue una asignación para un rol. Reemplace *`my-role`* por el nombre de su rol. Reemplace *`eks-console-dashboard-full-access-group`* con el nombre del grupo especificado en su objeto Kubernetes `RoleBinding` o `ClusterRoleBinding`. Reemplace *`111122223333`* por su ID de cuenta. Puede reemplazar *`admin`* (administrador) con cualquier nombre que elija.

```
eksctl create iamidentitymapping --cluster my-cluster --region=region-code \
    --arn arn:aws:iam::111122223333:role/my-role --username admin --group eks-
console-dashboard-full-access-group \
    --no-duplicate-arns
```

   > **Important**
   >
   > El ARN de rol no puede incluir una ruta como `role/my-team/developers/my-role`. El formato del ARN debe ser `arn:aws:iam::111122223333:role/my-role`. En este ejemplo, se necesita eliminar `my-team/developers/`.

   Un ejemplo de salida sería el siguiente.

```
[...]
2022-05-09 14:51:20 [#]  adding identity "arn:aws:iam::111122223333:role/my-
role" to auth ConfigMap
```

3. Agregue una asignación para un usuario. Según las prácticas recomendadas de IAM, se recomienda conceder permisos a los roles en lugar de a los usuarios. Reemplace *my-user* por el nombre de usuario. Reemplace *eks-console-dashboard-restricted-access-group* con el nombre del grupo especificado en su objeto Kubernetes `RoleBinding` o `ClusterRoleBinding`. Reemplace *111122223333* por su ID de cuenta. Puede reemplazar *my-user* (mi usuario) con cualquier nombre que elija.

```
eksctl create iamidentitymapping --cluster my-cluster --region=region-code \
    --arn arn:aws:iam::111122223333:user/my-user --username my-user --
group eks-console-dashboard-restricted-access-group \
    --no-duplicate-arns
```

Un ejemplo de salida sería el siguiente.

```
[...]
2022-05-09 14:53:48 [#]  adding identity "arn:aws:iam::111122223333:user/my-
user" to auth ConfigMap
```

4. Vea las asignaciones en el `ConfigMap` de nuevo.

```
eksctl get iamidentitymapping --cluster my-cluster --region=region-code
```

Un ejemplo de salida sería el siguiente.

```
ARN
                  USERNAME                                GROUPS
                    ACCOUNT
arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
arn:aws:iam::111122223333:role/admin
                      my-role                                 eks-console-
dashboard-full-access-group
```

```
arn:aws:iam::111122223333:user/my-user
                        my-user                                    eks-console-
dashboard-restricted-access-group
```

Edit ConfigMap manually

1. Abra el ícono `ConfigMap` para editar.

```
kubectl edit -n kube-system configmap/aws-auth
```

> **Note**
>
> Si recibe un error que indica “`Error from server (NotFound): configmaps "aws-auth" not found`”, entonces utilice el procedimiento en Aplique el `ConfigMap` de `aws-auth` en su clúster para aplicar el `ConfigMap` de stock.

2. Agregue las entidades principales de IAM al `ConfigMap`. Un grupo de IAM no es una entidad principal de IAM, por lo que no se puede agregar al `ConfigMap`.
   - A fin de agregar un rol de IAM (por ejemplo, para usuarios federados): agregue los detalles del rol a la sección `mapRoles` del `ConfigMap`, en `data`. Agregue esta sección si no existe todavía en el archivo. Cada entrada admite los siguientes parámetros:
     - rolearn: ARN del rol de IAM que se va a agregar. Este valor no puede incluir una ruta. Por ejemplo, no puede especificar un ARN como `arn:aws:iam::111122223333:role/my-team/developers/role-name`. El ARN se debe `arn:aws:iam::111122223333:role/role-name` en su lugar.
     - username (nombre de usuario): nombre del usuario de Kubernetes al que se mapea el rol de IAM.
     - groups (grupos): el grupo o la lista de grupos de Kubernetes a los que asignar el rol. El grupo puede ser un grupo predeterminado o un grupo especificado en un `clusterrolebinding` o `rolebinding`. Para obtener más información, consulte Roles predeterminados y enlaces de roles en la documentación de Kubernetes.
   - Para agregar un usuario de IAM: Según las prácticas recomendadas de IAM, se recomienda conceder permisos a los roles en lugar de a los usuarios. Agregue los detalles del usuario a la sección `mapUsers` del `ConfigMap`, en `data`. Agregue

esta sección si no existe todavía en el archivo. Cada entrada admite los siguientes parámetros:

- userarn: ARN del usuario de IAM que se va a agregar.
- username (nombre de usuario): el nombre de usuario dentro de Kubernetes al que se mapea el usuario de IAM.
- groups (grupos): el grupo o la lista de grupos de Kubernetes a los que asignar el usuario. El grupo puede ser un grupo predeterminado o un grupo especificado en un `clusterrolebinding` o `rolebinding`. Para obtener más información, consulte Roles predeterminados y enlaces de roles en la documentación de Kubernetes.

Por ejemplo, el siguiente bloque YAML contiene:

- Una sección de `mapRoles` que asigna la instancia de nodos de IAM a grupos de Kubernetes para que los nodos puedan registrarse en el clúster y el rol de IAM `my-console-viewer-role` que se asigna a un grupo de Kubernetes que puede ver todos los recursos de Kubernetes para todos los clústeres. Para obtener una lista de los permisos de grupo de IAM y Kubernetes necesarios para el rol de IAM de `my-console-viewer-role`, consulte Permisos necesarios.
- Una sección de `mapUsers` que asigna el usuario de IAM `admin` desde la cuenta de valor predeterminado de AWS al grupo de Kubernetes `system:masters` y el usuario `my-user` de otra cuenta AWS asignada a un grupo de Kubernetes que puede ver los recursos de Kubernetes para un espacio de nombres específico. Para obtener una lista de los permisos de grupo de IAM y Kubernetes necesarios para el usuario de IAM de `my-user`, consulte Permisos necesarios.

Agregue o quite líneas según sea necesario y sustituya todos los *`example values`* con sus propios valores.

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this
 file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
```

```
      rolearn: arn:aws:iam::111122223333:role/my-role
      username: system:node:{{EC2PrivateDNSName}}
    - groups:
      - eks-console-dashboard-full-access-group
      rolearn: arn:aws:iam::111122223333:role/my-console-viewer-role
      username: my-console-viewer-role
  mapUsers: |
    - groups:
      - system:masters
      userarn: arn:aws:iam::111122223333:user/admin
      username: admin
    - groups:
      - eks-console-dashboard-restricted-access-group
      userarn: arn:aws:iam::444455556666:user/my-user
      username: my-user
```

3. Guarde el archivo y salga del editor de texto.

Aplique el **ConfigMap** de **aws-auth** en su clúster

El `ConfigMap` de `aws-auth` se crea y aplica de forma automática al clúster cuando crea un grupo de nodos administrados o cuando crea un grupo de nodos mediante `eksctl`. En un principio, se crea para permitir que los nodos se unan al clúster, pero también se utiliza este `ConfigMap` para agregar acceso de control de acceso basado en roles (RBAC) a las entidades principales de IAM. Si ha lanzado nodos autoadministrados y aplicado el `ConfigMap` de `aws-auth` al clúster, puede hacerlo con el siguiente procedimiento.

Para aplicar **aws-authConfigMap** en su clúster

1. Verifique si ya ha aplicado el `ConfigMap` de `aws-auth`.

   ```
   kubectl describe configmap -n kube-system aws-auth
   ```

   Si recibe un error con estado “`Error from server (NotFound): configmaps "aws-auth" not found`”, continúe con los siguientes pasos para aplicar el `ConfigMap` de stock.

2. Descargue, edite y aplique el mapa de configuración del autenticador de AWS.

   a. Descargue el mapa de configuración.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/aws-auth-cm.yaml
```

b. En el archivo `aws-auth-cm.yaml`, establezca el `rolearn` para el Nombre de recurso de Amazon (ARN) o el rol de IAM asociado con sus nodos. Puede hacerlo con un editor de texto o al reemplazar *my-node-instance-role* y ejecutar el siguiente comando:

```
sed -i.bak -e 's|<ARN of instance role (not instance profile)>|my-node-
instance-role|' aws-auth-cm.yaml
```

No modifique ninguna otra línea de este archivo.

> **Important**
>
> El ARN de rol no puede incluir una ruta como `role/my-team/developers/my-role`. El formato del ARN debe ser `arn:aws:iam::111122223333:role/my-role`. En este ejemplo, se debe eliminar `my-team/developers/`.

Puede inspeccionar las salidas de la pila de AWS CloudFormation para los grupos de nodos y buscar los siguientes valores:

- InstanceRoleARN: para grupos de nodos que se crearon con `eksctl`
- NodeInstanceRole: para grupos de nodos que se crearon con plantillas de Amazon EKS incluidas en AWS CloudFormation en la AWS Management Console

c. Aplique la configuración. Este comando puede tardar varios minutos en finalizar.

```
kubectl apply -f aws-auth-cm.yaml
```

> **Note**
>
> Si recibe cualquier error de tipo de recurso o autorización, consulte Acceso denegado o no autorizado (`kubectl`) en el tema de solución de problemas.

3. Observe el estado de los nodos y espere a que aparezca el estado `Ready`.

```
kubectl get nodes --watch
```

Ingrese `Ctrl+C` para obtener un símbolo del intérprete de comandos.

Autenticación de usuarios para el clúster desde un proveedor de identidad de OpenID Connect

Amazon EKS admite el uso de proveedores de identidad OpenID Connect (OIDC) como método para autenticar usuarios en su clúster. Los proveedores de identidad OIDC se pueden utilizar con o como una alternativa a AWS Identity and Access Management (IAM). Para obtener más información acerca del uso de IAM, consulte the section called “Concesión de acceso a las API de Kubernetes”. Después de configurar la autenticación de su clúster, puede crear `roles` y `clusterroles` de Kubernetes para asignar permisos a los roles y, a continuación, vincular los roles a las identidades con `rolebindings` y `clusterrolebindings` de Kubernetes. Para obtener más información, consulte Utilización de la autorización de RBAC en la documentación de Kubernetes.

Consideraciones

- Puede asociar un proveedor de identidad OIDC al clúster.
- Kubernetes no proporciona un proveedor de identidad OIDC por sí mismo. Puede utilizar un proveedor de identidad OIDC público existente o ejecutar su propio proveedor de identidad. Para obtener una lista de proveedores certificados, consulte Certificación de OpenID en la página web de OpenID.
- La URL del emisor del proveedor de identidad OIDC debe ser accesible de manera pública para que Amazon EKS pueda descubrir las claves de firma. Amazon EKS no admite proveedores de identidad OIDC con certificados autofirmados.
- No es posible desactivar el autenticador de IAM en el clúster, ya que aún es necesario para unir nodos al clúster.
- Aún debe crearse un clúster de Amazon EKS mediante una entidad principal de IAM de AWS, en lugar de un usuario de proveedor de identidad OIDC. Esto se debe a que el creador del clúster interactúa con las API de Amazon EKS, en lugar de con las API de Kubernetes.
- Los usuarios autenticados por el proveedor de identidad OIDC aparecen en el registro de auditoría del clúster si los registros de CloudWatch están habilitados para el plano de control. Para obtener más información, consulte Habilitar y deshabilitar registros de plano de control.

- No es posible iniciar sesión en la AWS Management Console con una cuenta de un proveedor de identidad OIDC. Solo puede ver los recursos de Kubernetes en la consola al iniciar sesión en la AWS Management Console con una cuenta de AWS Identity and Access Management.

Asociación de un proveedor de identidad OIDC

Antes de asociar un proveedor de identidad OIDC al clúster, necesita la siguiente información de su proveedor:

URL del emisor

La URL del proveedor de identidad de OIDC que permite al servidor de API descubrir claves de firma públicas para verificar tokens. La URL debe comenzar con `https://` y debe corresponder a la afirmación `iss` en los tokens de ID de OIDC del proveedor. De acuerdo con el estándar de OIDC, los componentes de ruta están permitidos, pero los parámetros de consulta no lo están. Normalmente, la URL consta de solo un nombre de host, como `https://server.example.org` o `https://example.com`. Esta URL debe apuntar al siguiente nivel a `.well-known/openid-configuration` y debe ser de acceso público a través de Internet.

ID de cliente (también conocido como público)

El ID de la aplicación cliente que realiza solicitudes de autenticación al proveedor de identidad de OIDC.

Puede asociar un proveedor de identidad mediante `eksctl` o la AWS Management Console.

eksctl

Para asociar un proveedor de identidad OIDC al clúster con **`eksctl`**

1. Cree un archivo denominado *`associate-identity-provider.yaml`* con el siguiente contenido. Reemplace los *`example values`* por los de su propiedad. Los valores de la sección `identityProviders` se obtienen de su proveedor de identidad OIDC. Los valores solo son necesarios para la configuración de `name`, `type`, `issuerUrl` y `clientId` en `identityProviders`.

```
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
```

```
metadata:
  name: my-cluster
  region: your-region-code

identityProviders:
  - name: my-provider
    type: oidc
    issuerUrl: https://example.com
    clientId: kubernetes
    usernameClaim: email
    usernamePrefix: my-username-prefix
    groupsClaim: my-claim
    groupsPrefix: my-groups-prefix
    requiredClaims:
      string: string
    tags:
      env: dev
```

> **Important**
>
> No especifique `system:`, o cualquier parte de esa cadena, para `groupsPrefix` o `usernamePrefix`.

2. Cree el proveedor.

```
eksctl associate identityprovider -f associate-identity-provider.yaml
```

3. Para utilizar `kubectl` para trabajar con su clúster y el proveedor de identidad de OIDC, consulte Uso de `kubectl` en la documentación de Kubernetes.

AWS Management Console

Cómo asociar un proveedor de identidad de OIDC al clúster con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Seleccione el clúster y, a continuación, seleccione la pestaña Acceso.
3. En la sección Proveedores de identidad de OIDC, seleccione Proveedor de identidad de asociado.
4. En la página Asociar proveedor de identidad OIDC, ingrese o seleccione las siguientes opciones y, a continuación, seleccione Asociar.

- En Name (Nombre), ingrese un nombre único para el proveedor.
- En Issuer URL (URL del emisor), ingrese la URL del proveedor. Debe ser posible el acceso a esta URL a través de Internet.
- En ID de cliente, ingrese el ID de cliente del proveedor de identidad OIDC (también conocido como audiencia).
- En Username claim (Afirmación de nombre de usuario), ingrese la afirmación que desea utilizar como nombre de usuario.
- En Groups claim (Afirmación de grupo), ingrese la afirmación que desea utilizar como grupo del usuario.
- (Opcional) Seleccione Opciones avanzadas, e ingrese o seleccione la siguiente información.
  - Username prefix (Prefijo de nombre de usuario): ingrese un prefijo para anteponer a las afirmaciones de nombre de usuario. El prefijo se antepone a las afirmaciones de nombre de usuario para evitar conflictos con nombres existentes. Si no proporciona un valor y el nombre de usuario es un valor distinto de `email`, el prefijo se establece de forma predeterminada en el valor de Issuer URL (URL del emisor). Puede utilizar el valor `-` para desactivar todos los prefijos. No especifique `system:` o cualquier parte de esa cadena.
  - Groups prefix (Prefijo de grupos): ingrese un prefijo para anteponer a las afirmaciones de grupos. El prefijo se antepone a las afirmaciones de grupo para evitar conflictos con nombres existentes (por ejemplo, `system: groups`). Por ejemplo, el valor `oidc:` crea nombres de grupo como `oidc:engineering` y `oidc:infra`. No especifique `system:` o cualquier parte de esa cadena.
  - Required claims (Afirmaciones requeridas): seleccione Add claim (Agregar afirmación) e ingrese uno o más pares de valor de clave que describan las afirmaciones requeridas en el token del ID de cliente. Los pares describen las afirmaciones requeridas en el token del ID. Si se establece, se verifica que cada afirmación esté presente en el token del ID con un valor coincidente.

5. Para utilizar `kubectl` para trabajar con su clúster y el proveedor de identidad OIDC, consulte Uso de `kubectl` en la documentación de Kubernetes.

Cómo desasociar un proveedor de identidad OIDC del clúster

Si desasocia un proveedor de identidad OIDC del clúster, los usuarios incluidos en el proveedor ya no podrán acceder al clúster. Sin embargo, sigue teniendo acceso al clúster con las entidades principales de IAM.

Cómo desasociar un proveedor de identidad OIDC del clúster con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En la sección Proveedores de identidad OIDC, seleccione Desasociar, ingrese el nombre del proveedor de identidad y, a continuación, seleccione `Disassociate`.

Política de IAM de ejemplo

Si desea evitar que un proveedor de identidad OIDC se asocie a un clúster, cree y asocie la siguiente política de IAM a las cuentas de IAM de sus administradores de Amazon EKS. Para obtener más información, consulte Creación de políticas de IAM y Agregación de permisos de identidad de IAM en la Guía del usuario de IAM y Claves de condición, recursos y acciones de Amazon Elastic Kubernetes Service en la Referencia de autorizaciones de servicio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "denyOIDC",
            "Effect": "Deny",
            "Action": [
                "eks:AssociateIdentityProviderConfig"
            ],
            "Resource": "arn:aws:eks:us-west-2.amazonaws.com:111122223333:cluster/*"

        },
        {
            "Sid": "eksAdmin",
            "Effect": "Allow",
            "Action": [
                "eks:*"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

La siguiente política de ejemplo permite la asociación de proveedores de identidad OIDC si el `clientID` es `kubernetes` y la `issuerUrl` es `https://cognito-idp.us-west-2amazonaws.com/*`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCognitoOnly",
            "Effect": "Deny",
            "Action": "eks:AssociateIdentityProviderConfig",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-instance",
            "Condition": {
                "StringNotLikeIfExists": {
                    "eks:issuerUrl": "https://cognito-idp.us-west-2.amazonaws.com/*"
                }
            }
        },
        {
            "Sid": "DenyOtherClients",
            "Effect": "Deny",
            "Action": "eks:AssociateIdentityProviderConfig",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-instance",
            "Condition": {
                "StringNotEquals": {
                    "eks:clientId": "kubernetes"
                }
            }
        },
        {
            "Sid": "AllowOthers",
            "Effect": "Allow",
            "Action": "eks:*",
            "Resource": "*"
        }
    ]
}
```

Proveedores de identidad OIDC validados por socios

Amazon EKS mantiene relaciones con una red de socios que ofrecen soporte para proveedores de identidad OIDC compatibles. Consulte la siguiente documentación de socios para obtener información detallada sobre cómo integrar al proveedor de identidades con Amazon EKS.

Socio	Producto	Documentación
PingIdentity	PingOne para empresas	Instrucciones de instalación

Amazon EKS tiene como objetivo darle una amplia selección de opciones para cubrir todos los casos de uso. Si desarrolla un proveedor de identidades OIDC compatible comercialmente que no aparece aquí, póngase en contacto con nuestro equipo de socios en aws-container-partners@amazon.com para obtener más información.

Creación o actualización de un archivo **`kubeconfig`** para un clúster de Amazon EKS

En este tema, creará un archivo `kubeconfig` para su clúster (o actualizará uno existente).

La herramienta de línea de comandos `kubectl` usa la información de configuración en los archivos `kubeconfig` para comunicarse con el servidor de API de un clúster. Para obtener más información, consulte Organización del acceso al clúster mediante archivos kubeconfig en la documentación de Kubernetes.

Amazon EKS usa el comando `aws eks get-token` con `kubectl` para la autenticación del clúster. De forma predeterminada, la AWS CLI utiliza las mismas credenciales que se devuelven con el siguiente comando:

```
aws sts get-caller-identity
```

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a

la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- Un rol o usuario de IAM con permisos para utilizar la acción de API `eks:DescribeCluster` en el clúster que especifique. Para obtener más información, consulte Ejemplos de políticas de Amazon EKS basadas en identidades. Si utiliza una identidad de su propio proveedor de OpenID Connect para acceder al clúster, consulte Uso de `kubectl` en la documentación de Kubernetes para crear o actualizar el archivo de `kube config`.

Crear el archivo **`kubeconfig`** de forma automática

Requisitos previos

- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- Permiso para usar la acción de API `eks:DescribeCluster` en el clúster que especifique. Para obtener más información, consulte Ejemplos de políticas de Amazon EKS basadas en identidades.

Para crear el archivo **kubeconfig** con la AWS CLI

1. Cree o actualice un archivo de kubeconfig para el clúster. Reemplace *region-code* con la Región de AWS en la que se encuentra su clúster y reemplace *my-cluster* con el nombre del clúster.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   De forma predeterminada, el archivo de configuración resultante se crea en la ruta de kubeconfig predeterminada (.kube) en el directorio de inicio o en combinación con un archivo config existente en dicha ubicación. Puede especificar otra ruta con la opción **--kubeconfig**.

   Puede especificar un ARN de rol de IAM con la opción **--role-arn** para utilizar en la autenticación al emitir comandos kubectl. De lo contrario, se utilizará la entidad principal de IAM de la AWS CLI predeterminada o las credenciales del SDK. Puede ver su identidad de AWS CLI o SDK predeterminada ejecutando el comando aws sts get-caller-identity.

   Para ver todas las opciones disponibles, ejecute el comando aws eks update-kubeconfig help o consulte update-kubeconfig en la Referencia de los comandos de la AWS CLI.
2. Pruebe la configuración.

   ```
   kubectl get svc
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME             TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
   svc/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   1m
   ```

   Si recibe cualquier error de tipo de recurso o autorización, consulte Acceso denegado o no autorizado (kubectl) en el tema de solución de problemas.

Concesión de acceso a las cargas de trabajo de Kubernetes a AWS mediante las cuentas de servicio de Kubernetes

Una cuenta de servicio de Kubernetes proporciona una identidad para los procesos que se ejecutan en un Pod. Para obtener más información, consulte Administración de cuentas de servicio en la documentación de Kubernetes. Si su Pod necesita acceso a servicios de AWS, puede asignar

la cuenta de servicio a una identidad AWS Identity and Access Management para conceder ese acceso. Para obtener más información, consulte Roles de IAM para cuentas de servicio.

Tokens de cuenta de servicio

La característica `BoundServiceAccountTokenVolume` está habilitada de forma predeterminada en las versiones Kubernetes. Esta función mejora la seguridad de los tokens de cuenta de servicio al permitir que las cargas de trabajo se ejecuten en Kubernetes para solicitar tokens web JSON que estén vinculados a la audiencia, la hora y la clave. Los tokens de cuenta de servicio tienen una caducidad de una hora. En versiones anteriores de Kubernetes, los tokens no tenían caducidad. Esto significa que los clientes que confían en estos tokens deben actualizar los tokens en una hora. Los siguientes ejemplos SDK de cliente de Kubernetes actualizan los tokens automáticamente dentro del plazo requerido:

- Versión de Go `0.15.7` y posteriores
- Versión de Python `12.0.0` y posteriores
- Versión de Java `9.0.0` y posterior
- Versión de JavaScript `0.10.3` y posterior
- Rama de Ruby `master`
- Versión de Haskell `0.3.0.0`
- Versión C# `7.0.5` y posterior

Si su carga de trabajo utiliza una versión de cliente anterior, debe actualizarla. Para permitir una migración fluida de los clientes a los tokens de cuenta de servicio con límite de tiempo más nuevos, Kubernetes agrega un período de vencimiento extendido al token de cuenta de servicio durante la hora predeterminada. Para los clústeres de Amazon EKS, el período de caducidad extendido es de 90 días. El servidor API de Kubernetes de los clústeres de Amazon EKS rechaza solicitudes con tokens de más de 90 días de antigüedad. Le recomendamos que compruebe sus aplicaciones y sus dependencias para asegurarse de que los SDK de cliente de Kubernetes son iguales o posteriores a las versiones indicadas anteriormente.

Cuando el servidor API recibe solicitudes con tokens de más de una hora de antigüedad, anota el evento de registro de auditoría de la API con `annotations.authentication.k8s.io/stale-token`. El valor de la anotación es similar al siguiente ejemplo:

```
subject: system:serviceaccount:common:fluent-bit, seconds after warning threshold:
 4185802.
```

Si el clúster tiene un registro de plano de control habilitado, entonces las anotaciones se encuentran en los registros de auditoría. Puede utilizar la siguiente consulta de CloudWatch Logs Insights para identificar todos los Pods del clúster de Amazon EKS que utilizan tokens obsoletos:

```
fields @timestamp
| filter @logStream like /kube-apiserver-audit/
| filter @message like /seconds after warning threshold/
| parse @message "subject: *, seconds after warning threshold:*\"" as subject,
 elapsedtime
```

El `subject` hace referencia a la cuenta de servicio que utilizó el Pod. El `elapsedtime` indica el tiempo transcurrido (en segundos) tras leer el último token. Las solicitudes al servidor API se denegan cuando el `elapsedtime` supera los 90 días (7 776 000 segundos). Debe actualizar de forma proactiva el SKD del cliente Kubernetes de las aplicaciones para utilizar una de las versiones enumeradas anteriormente que actualiza automáticamente el token. Si el token de cuenta de servicio utilizado dura casi 90 días y no tiene tiempo suficiente para actualizar las versiones del SDK de cliente antes de que el token caduque, puede terminar los Pods existentes y crear otros nuevos. Esto da como resultado la reactivación del token de la cuenta de servicio, lo que le da 90 días adicionales para actualizar los SDK de la versión de cliente.

Si el Pod forma parte de una implementación, la forma sugerida de finalizar los Pods y mantener la alta disponibilidad es realizar un despliegue con el siguiente comando. Reemplace *my-deployment* con el nombre de la implementación.

```
kubectl rollout restart deployment/my-deployment
```

Complementos de clúster

Los siguientes complementos de clúster se han actualizado para utilizar los SDK del cliente Kubernetes que reactivan automáticamente los tokens de cuentas de servicio. Recomendamos asegurarse de que las versiones enumeradas, o versiones posteriores, estén instaladas en su clúster.

- Amazon VPC CNI plugin for Kubernetes y la versión de los complementos auxiliares de métricas `1.8.0` y posteriores. Para comprobar su versión actual o actualizarla, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS y cni-metrics-helper.
- Versión CoreDNS `1.8.4` y posterior. Para comprobar su versión actual o actualizarla, consulte Trabajar con el complemento CoreDNS de Amazon EKS.
- Versión AWS Load Balancer Controller `2.0.0` y posterior. Para comprobar su versión actual o actualizarla, consulte ¿Qué es el AWS Load Balancer Controller?.
- Una versión actual de `kube-proxy`. Para comprobar su versión actual o actualizarla, consulte Trabajar con el complemento `kube-proxy` Kubernetes.
- AWS para Fluent Bit versión `2.25.0` o posterior. Para actualizar la versión actual, consulte Releases (Versiones) en GitHub.
- Versión de imagen de Fluentd 1.14.6-1.2 o posterior y versión del complemento de filtro de Fluentd para metadatos de Kubernetes 2.11.1 o posterior.

Concesión de permisos AWS Identity and Access Management a cargas de trabajo en clústeres de Amazon Elastic Kubernetes Service

Amazon EKS ofrece dos formas de conceder AWS Identity and Access Management permisos a las cargas de trabajo que se ejecutan en los clústeres de Amazon EKS: los roles de IAM para cuentas de servicio y las identidades de Pod de EKS.

Roles de IAM para cuentas de servicio

> Los roles de IAM para cuentas de servicio (IRSA) configuran las aplicaciones de Kubernetes que se ejecutan en AWS con permisos de IAM detallados para acceder a otros recursos de AWS, como los buckets de Amazon S3, las tablas de Amazon DynamoDB y más. Puede ejecutar varias aplicaciones juntas en el mismo clúster de Amazon EKS y asegurarse de que cada aplicación tenga solo el conjunto mínimo de permisos que necesita. IRSA se creó para admitir varias opciones de implementación de Kubernetes compatibles por AWS como Amazon EKS, Amazon EKS Anywhere, Red Hat OpenShift Service en AWS y clústeres de Kubernetes autogestionados en instancias de Amazon EC2. Por lo tanto, IRSA se creó utilizando un servicio de AWS básico como IAM y no dependía directamente del servicio Amazon EKS ni de la API de EKS. Para obtener más información, consulte Roles de IAM para cuentas de servicio.

Pod Identities de EKS

Pod Identity de EKS ofrece a los administradores de clústeres un flujo de trabajo simplificado para autenticar las aplicaciones con el fin de acceder a otros recursos de AWS, como los buckets de Amazon S3, las tablas de Amazon DynamoDB y más. Pod Identity de EKS está dedicada solo para EKS y, como resultado, simplifica la forma en que los administradores de clústeres pueden configurar las aplicaciones de Kubernetes para obtener permisos de IAM. Estos permisos ahora se pueden configurar fácilmente con menos pasos: directamente a través de AWS Management Console, la API de EKS y AWS CLI, no es necesario realizar ninguna acción dentro del clúster ni en ningún objeto de Kubernetes. Los administradores de clústeres no necesitan cambiar entre los servicios de EKS y de IAM, ni utilizar operaciones de IAM privilegiadas para configurar los permisos que requieren sus aplicaciones. Los roles de IAM ahora se pueden usar en varios clústeres sin necesidad de actualizar la política de confianza de roles al crear nuevos clústeres. Las credenciales de IAM proporcionadas por Pod Identity de EKS incluyen etiquetas de sesión de rol, con atributos como el nombre del clúster, el espacio de nombres y el nombre de la cuenta de servicio. Las etiquetas de sesión de rol permiten a los administradores crear un único rol que puede funcionar en todas las cuentas de servicio, ya que permiten el acceso a los recursos de AWS en función de las etiquetas coincidentes. Para obtener más información, consulte Pod Identities de EKS.

Comparación de Pod Identity de EKS e IRSA

A un nivel alto, tanto Pod Identity de EKS como IRSA permiten conceder permisos de IAM a las aplicaciones que se ejecutan en clústeres de Kubernetes. Sin embargo, son fundamentalmente diferentes en cuanto a la forma de configurarlos, los límites admitidos y las funciones habilitadas. A continuación, comparamos algunos de los aspectos clave de ambas soluciones.

	Pod Identity de EKS	IRSA
Extensibilidad de roles	Debe configurar cada rol una vez para establecer una relación de confianza con el recién introducido `pods.eks.amazonaws.com` de director de servicio de Amazon EKS. Tras este único paso, no necesitar	Debe actualizar la política de confianza del rol de IAM con el nuevo punto de conexión del proveedor OIDC de clústeres de EKS cada vez que desee utilizar el rol en un clúster nuevo.

	Pod Identity de EKS	IRSA
	á actualizar la política de confianza del rol cada vez que lo utilice en un clúster nuevo.	
Escalabilidad de los clústeres	Pod Identity de EKS no requiere que los usuarios configuren un proveedor de IAM OIDC, por lo que este límite no se aplica.	Cada clúster de EKS tiene una URL de emisor de OpenID Connect (OIDC) asociada. Para usar IRSA, es necesario crear un proveedor OpenID Connect único para cada clúster de EKS de IAM. IAM tiene un límite global predeterminado de 100 proveedores de OIDC para cada Cuenta de AWS. Si planea tener más de 100 clústeres de EKS para cada Cuenta de AWS con IRSA, alcanzará el límite de proveedor OIDC de IAM.

	Pod Identity de EKS	IRSA
Escalabilidad de las funciones	Pod Identity de EKS no exige que los usuarios definan la relación de confianza entre el rol de IAM y la cuenta de servicio en la política de confianza, por lo que este límite no se aplica.	En IRSA, usted define la relación de confianza entre un rol de IAM y una cuenta de servicio en la política de confianza del rol. De forma predeterminada, el tamaño de la política de confianza es 2048. Esto significa que normalmente se pueden definir 4 relaciones de confianza en una sola política de confianza. Si bien puede aumentar el límite de duración de la política de confianza, normalmente está limitado a un máximo de 8 relaciones de confianza dentro de una sola política de confianza.

	Pod Identity de EKS	IRSA
Reutilización de roles	Las credenciales temporales de AWS STS proporcionadas por Pod Identity de EKS incluyen etiquetas de sesión de rol, como el nombre del clúster, el espacio de nombres y el nombre de la cuenta de servicio. Las etiquetas de sesión de rol permiten a los administradores crear un único rol de IAM que se puede usar con varias cuentas de servicio, con diferentes permisos efectivos, ya que permiten el acceso a los recursos de AWS basados en las etiquetas adjuntas a ellas. Esto también se conoce como control de acceso basado en atributos (ABAC). Para obtener más información, consulte Definición de permisos para que Pod Identities de EKS asuman roles basados en etiquetas.	No se admiten etiquetas de sesión de AWS STS. Puede reutilizar un rol entre clústeres, pero cada pod recibe todos los permisos del rol.
Entornos compatibles	Pod Identity de EKS solo está disponible en Amazon EKS.	Se puede usar IRSA como Amazon EKS, Amazon EKS Anywhere, Red Hat OpenShift Service en AWS y clústeres de Kubernetes autogestionados en instancias de Amazon EC2.

	Pod Identity de EKS	IRSA
Versiones compatibles de EKS	Versiones `1.24` de Kubernetes de EKS o posteriores. Para saber las versiones de la plataforma específicas, consulte Versiones del clúster de Pod Identity de EKS.	Todas las versiones del clúster EKS compatibles.

Pod Identities de EKS

Las aplicaciones de los contenedores de un Pod pueden usar un SDK de AWS o la AWS CLI para llevar a cabo solicitudes de API a Servicios de AWS mediante permisos de AWS Identity and Access Management (IAM). Las aplicaciones deben firmar sus solicitudes de API AWS con credenciales de AWS.

Pod Identities de EKS ofrecen la posibilidad de administrar las credenciales para las aplicaciones, de un modo similar a cómo los perfiles de instancia de Amazon EC2 proporcionan credenciales a instancias de Amazon EC2. En lugar de crear y distribuir las credenciales de AWS a los contenedores o de utilizar el rol de la instancia de Amazon EC2, puede asociar el rol de IAM con una cuenta de servicio de Kubernetes y configurar los Pods para usar la cuenta de servicio.

Cada asociación de Pod Identity de EKS asigna un rol a una cuenta de servicio en un espacio de nombres del clúster especificado. Si tiene la misma aplicación en varios clústeres, puede crear asociaciones idénticas en cada clúster sin modificar la política de confianza del rol.

Si un pod usa una cuenta de servicio que tiene una asociación, Amazon EKS establece las variables de entorno en los contenedores del pod. Las variables de entorno configuran los SDK de AWS, incluida la AWS CLI, para usar las credenciales de la Pod Identity de EKS.

Ventajas de las Pod Identities de EKS

Las Pod Identities de EKS proporcionan los siguientes beneficios:

- Privilegio mínimo: puede limitar los permisos de IAM a una cuenta de servicio y solo los Pods que utilizan esa cuenta de servicio tienen acceso a esos permisos. Esta característica también elimina la necesidad de soluciones de terceros como `kiam` o `kube2iam`.

- Aislamiento de credenciales: los contenedores de un Pod's solo pueden recuperar las credenciales para el rol de IAM asociado a la cuenta de servicio que usa el contenedor. Un contenedor nunca tiene acceso a credenciales que utilizan otros contenedores de otros Pods. Al utilizar Pod Identities, los contenedores de Pod's también tienen los permisos asignados al rol de IAM del nodo de Amazon EKS, a menos que bloquee el acceso del Pod al servicio de metadatos de instancias (IMDS) de Amazon EC2. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.
- Auditabilidad: El acceso y el registro de eventos está disponible a través de AWS CloudTrail para facilitar una auditoría retrospectiva.

Pod Identity de EKS es un método más sencillo que Roles de IAM para cuentas de servicio, ya que no utiliza proveedores de identidad OIDC. Pod Identity de EKS incluye las siguientes mejoras:

- Operaciones independientes: En muchas organizaciones, la creación de proveedores de identidad OIDC es una responsabilidad de equipos diferentes a la de administrar los clústeres Kubernetes. Pod Identity de EKS tiene una clara separación de funciones, en donde toda la configuración de las asociaciones de Pod Identity de EKS se realiza en Amazon EKS y toda la configuración de los permisos de IAM se realiza en IAM.
- Reutilización: La Pod Identity de EKS utiliza una única entidad principal de IAM en lugar de los principios independientes para cada clúster que utilizan los roles de IAM para cuentas de servicio. El administrador de IAM añade la siguiente entidad principal a la política de confianza de cualquier función para que Pod Identities de EKS puedan utilizarla.

```
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            }
```

- Escalabilidad: cada conjunto de credenciales temporales lo asume el servicio de EKS Auth en Pod Identity de EKS, en lugar de cada SDK de AWS que se ejecuta en cada pod. A continuación, el agente de Pod Identity de Amazon EKS que se ejecuta en cada nodo emite las credenciales de los SDK. De este modo, la carga se reduce a una vez para cada nodo y no se duplica en cada pod. Para obtener más información del proceso, consulte Cómo funciona Pod Identity de EKS.

Para obtener más información sobre cómo comparar las dos alternativas, consulte Concesión de acceso a las cargas de trabajo de Kubernetes a AWS mediante las cuentas de servicio de Kubernetes.

Información general de configuración de las Pod Identities de EKS

Siga estos procedimientos para activar Pod Identities de EKS:

1. Configuración del agente de Pod Identity de Amazon EKS: solo complete este procedimiento una vez para cada clúster.
2. Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM con Pod Identity de EKS: complete este procedimiento para cada conjunto único de permisos que desee que tenga una aplicación.
3. Configuración de Pods para usar una cuenta de servicio de Kubernetes: complete este procedimiento para cada Pod que necesite acceso a Servicios de AWS.
4. Uso de un AWS SDK compatible: confirme que la carga de trabajo utilice un SDK de AWS de una versión compatible y que utilice la cadena de credenciales predeterminada.

Consideraciones sobre Pod Identity de EKS

- Puede asociar un rol de IAM a cada cuenta de servicio de Kubernetes en cada clúster. Puede cambiar el rol asignado a la cuenta de servicio editando la asociación de Pod Identity de EKS.
- Solo puede asociar roles que estén en el mismo Cuenta de AWS como el clúster. Puede delegar el acceso desde otra cuenta al rol de esta cuenta que haya configurado para que lo utilice Pod Identities de EKS. Para ver un tutorial de cómo delegar el acceso y `AssumeRole`, consulte Delegate access across AWS accounts using IAM roles en la Guía de usuario de IAM.
- Se requiere el agente de Pod Identity de EKS. Se ejecuta como Kubernetes `DaemonSet` en sus nodos y solo proporciona credenciales a los pods del nodo en el que se ejecuta. Para obtener más información acerca de la compatibilidad del agente de Pod Identity de EKS, consulte la siguiente sección Restricciones de Pod Identity de EKS.
- El agente de Pod Identity de EKS utiliza el `hostNetwork` del nodo y utiliza el puerto `80` y el puerto `2703` en una dirección de enlace local del nodo. Esta dirección es `169.254.170.23` para IPv4 y `[fd00:ec2::23]` para los clústeres IPv6.

  Si deshabilita las direcciones `IPv6` o evita las direcciones IP `IPv6` de localhost, el agente no podrá iniciarse. Para iniciar el agente en los nodos que no pueden usar `IPv6`, siga los pasos que se indican en Desactivar `IPv6` en el agente de Pod Identity de EKS a fin de deshabilitar la configuración `IPv6`.

Versiones del clúster de Pod Identity de EKS

Para usar Pod Identities de EKS, el clúster debe tener una versión de la plataforma igual o posterior a la que se indica en la siguiente tabla, o una versión de Kubernetes posterior a las versiones que se muestran en la tabla.

Versión de Kubernetes	Versión de la plataforma
`1.30`	`eks.2`
`1.29`	`eks.1`
`1.28`	`eks.4`
`1.27`	`eks.8`
`1.26`	`eks.9`
`1.25`	`eks.10`
`1.24`	`eks.13`

Versiones de complementos compatibles con Pod Identity de EKS

> **Important**
>
> Para utilizar Pod Identity de EKS con un complemento de EKS, debe crear la asociación de Pod Identity de EKS de forma manual. No elija un rol de IAM en la configuración del complemento en la AWS Management Console, ya que ese rol solo se usa con IRSA.

Los complementos de Amazon EKS y los complementos autoadministrados que necesitan credenciales de IAM pueden usar Pod Identity de EKS, IRSA o el rol de instancia. En la lista de complementos que utilizan credenciales de IAM compatibles con Pod Identity de EKS, se incluyen los siguientes:

- Amazon VPC CNI plugin for Kubernetes `1.15.5-eksbuild.1` o posterior

- AWS Load Balancer Controller `2.7.0` o posterior Tenga en cuenta que AWS Load Balancer Controller no está disponible como complemento de EKS, pero está disponible como complemento autoadministrado.

Restricciones de Pod Identity de EKS

Pod Identities de EKS están disponibles en las siguientes ubicaciones:

- Versiones del clúster de Amazon EKS enumeradas en el tema anterior Versiones del clúster de Pod Identity de EKS.
- Nodos de trabajo del clúster que son instancias Linux Amazon EC2.

Pod Identities de EKS no están disponibles en las siguientes ubicaciones:

- Regiones de China.
- AWS GovCloud (US).
- AWS Outposts.
- Amazon EKS Anywhere.
- Los clústeres Kubernetes que se crean y ejecutan en Amazon EC2. Los componentes de Pod Identity de EKS solo están disponibles en Amazon EKS.

No puede usar Pod Identities de EKS con:

- Pods que se ejecutan en cualquier lugar, excepto instancias Linux Amazon EC2. No se admiten los pods de Linux y Windows que se ejecutan en AWS Fargate (Fargate). No se admiten pods que se ejecutan en instancias Windows Amazon EC2.
- Complementos de Amazon EKS que necesitan credenciales de IAM. En su lugar, los complementos de EKS solo pueden utilizar roles de IAM para cuentas de servicio. La lista de complementos de EKS que utilizan credenciales de IAM incluye:
  - Los controladores de almacenamiento de CSI: EBS CSI, EFS CSI, controlador CSI Amazon FSx para Lustre, controlador CSI Amazon FSx para NetApp ONTAP, controlador de CSI de Amazon FSx para OpenZFS, controlador CSI Amazon File Cache, AWS Secrets and Configuration Provider (ASCP) para el controlador CSI Kubernetes Secrets Store.

> **Note**
>
> Si estos controladores, controladores y complementos se instalan como complementos autogestionados en lugar de como complementos de EKS, son compatibles con las identidades de Pod de EKS siempre que estén actualizados para utilizar los SDK de AWS más recientes.

Cómo funciona Pod Identity de EKS

Las asociaciones de Pod Identity de Amazon EKS ofrecen la posibilidad de administrar las credenciales para las aplicaciones, de un modo similar a cómo los perfiles de instancia de Amazon EC2 proporcionan credenciales a instancias de Amazon EC2.

Pod Identity de Amazon EKS proporciona credenciales a sus cargas de trabajo con una API de autenticación de EKS adicional y un pod de agente que se ejecuta en cada nodo.

En sus complementos, como los complementos de Amazon EKS y el controlador autogestionado, los operadores y otros complementos, el autor debe actualizar el software para utilizar los SDK de AWS más recientes. Para ver la lista de compatibilidad entre Pod Identity de EKS y los complementos fabricados por Amazon EKS, consulte la sección anterior Restricciones de Pod Identity de EKS.

Uso de Identidades de pod de EKS en el código

En su código, puede usar los SDK de AWS para acceder a los servicios de AWS. El código se escribe para crear un cliente para un servicio de AWS con un SDK y, de forma predeterminada, el SDK busca en una cadena de ubicaciones las credenciales de AWS Identity and Access Management que se van a utilizar. Una vez que se ha comprobado que las credenciales son válidas, se detiene la búsqueda. Para obtener más información sobre las ubicaciones predeterminadas utilizadas, consulte la cadena de proveedores de credenciales en la Guía de referencia de herramientas y SDK de AWS.

Se han agregado las Pod Identities de EKS al proveedor de credenciales del contenedor, que se busca en un paso de la cadena de credenciales predeterminada. Si sus cargas de trabajo utilizan actualmente credenciales que se encuentran en una fase anterior de la cadena de credenciales, esas credenciales seguirán utilizándose aunque configure una asociación de Pod Identity de EKS para la misma carga de trabajo. De esta forma, puede migrar de forma segura desde otros tipos de credenciales creando primero la asociación antes de eliminar las credenciales antiguas.

El proveedor de credenciales del contenedor proporciona credenciales temporales de un agente que se ejecuta en cada nodo. En Amazon EKS, el agente de Pod Identity de Amazon EKS y en Servicio de contenedor elástico de Amazon, el agente es el `amazon-ecs-agent`. Los SDK utilizan variables de entorno para localizar el agente al que conectarse.

Por el contrario, los roles de IAM para las cuentas de servicio proporcionan un token de identidad web que el SDK de AWS debe intercambiar con AWS Security Token Service usando `AssumeRoleWithWebIdentity`.

Cómo funciona el agente de Pod Identity de EKS con un Pod

1. Cuando Amazon EKS inicia un nuevo pod que utiliza una cuenta de servicio con una asociación de Pod Identity de EKS, el clúster agrega el siguiente contenido al manifiesto de Pod:

```
    env:
    - name: AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE
      value: "/var/run/secrets/pods.eks.amazonaws.com/serviceaccount/eks-pod-
identity-token"
    - name: AWS_CONTAINER_CREDENTIALS_FULL_URI
      value: "http://169.254.170.23/v1/credentials"
    volumeMounts:
    - mountPath: "/var/run/secrets/pods.eks.amazonaws.com/serviceaccount/"
      name: eks-pod-identity-token
  volumes:
  - name: eks-pod-identity-token
    projected:
      defaultMode: 420
      sources:
      - serviceAccountToken:
          audience: pods.eks.amazonaws.com
          expirationSeconds: 86400 # 24 hours
          path: eks-pod-identity-token
```

2. Kubernetes selecciona en qué nodo se va a ejecutar el pod. A continuación, el agente de Pod Identity de Amazon EKS del nodo utiliza la acción AssumeRoleForPodIdentity para recuperar las credenciales temporales de la API de autenticación de EKS.
3. El agente de Pod Identity de EKS pone estas credenciales a disposición de los SDK de AWS que ejecuta en sus contenedores.
4. Utilice el SDK en su aplicación sin especificar un proveedor de credenciales para utilizar la cadena de credenciales predeterminada. O bien, puede especificar el proveedor de credenciales del contenedor. Para obtener más información sobre las ubicaciones predeterminadas utilizadas,

consulte la cadena de proveedores de credenciales en la Guía de referencia de herramientas y SDK de AWS.

5. El SDK utiliza las variables de entorno para conectarse al agente de Pod Identity de EKS y recuperar las credenciales.

> **Note**
>
> Si sus cargas de trabajo utilizan actualmente credenciales que se encuentran en una fase anterior de la cadena de credenciales, esas credenciales seguirán utilizándose aunque configure una asociación de Pod Identity de EKS para la misma carga de trabajo.

Configuración del agente de Pod Identity de Amazon EKS

Las asociaciones de Pod Identity de Amazon EKS ofrecen la posibilidad de administrar las credenciales para las aplicaciones, de un modo similar a cómo los perfiles de instancia de Amazon EC2 proporcionan credenciales a instancias de Amazon EC2.

Pod Identity de Amazon EKS proporciona credenciales a sus cargas de trabajo con una API de autenticación de EKS adicional y un pod de agente que se ejecuta en cada nodo.

Consideraciones

- **`IPv6`**

  De forma predeterminada, el agente de Pod Identity de EKS escucha en una dirección `IPv4` e `IPv6` para que los pods soliciten credenciales. El agente usa la dirección IP de bucle invertido (localhost) `169.254.170.23` para `IPv4` y la dirección IP de localhost `[fd00:ec2::23]` para `IPv6`.

  Si deshabilita las direcciones `IPv6` o evita las direcciones IP `IPv6` de localhost, el agente no podrá iniciarse. Para iniciar el agente en los nodos que no pueden usar `IPv6`, siga los pasos que se indican en Desactivar `IPv6` en el agente de Pod Identity de EKS a fin de deshabilitar la configuración `IPv6`.

Creación del agente de Pod Identity de Amazon EKS

Requisitos previos de agente

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS. La versión del clúster y la versión de la plataforma deben ser iguales o posteriores a las versiones indicadas en Versiones del clúster de Pod Identity de EKS.
- El rol de nodo tiene permisos para que el agente realice la acción `AssumeRoleForPodIdentity` en la API de autenticación de EKS. Puede usar Política administrada por AWS: AmazonEKSWorkerNodePolicy o agregar una política personalizada similar a la siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks-auth:AssumeRoleForPodIdentity"
            ],
            "Resource": "*"
        }
    ]
}
```

  Esta acción se puede limitar mediante etiquetas para restringir los roles que pueden asumir los pods que utilizan el agente.
- Los nodos pueden acceder y descargar imágenes de Amazon ECR. La imagen del contenedor del complemento se encuentra en los registros que figuran en Registros de imágenes de contenedor de Amazon.

  Tenga en cuenta que puede cambiar la ubicación de la imagen y proporcionar `imagePullSecrets` para los complementos de EKS en Valores de configuración opcionales en la AWS Management Console, y en `--configuration-values` en la AWS CLI.
- Los nodos pueden acceder a la API de autenticación de Amazon EKS. En el caso de los clústeres privados, se requiere el punto de conexión `eks-auth` en AWS PrivateLink.

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

2. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento de agente de Pod Identity de EKS.
3. Elija la pestaña Complementos.
4. Escoja Obtener más complementos.
5. Seleccione la casilla situada en la parte superior derecha del cuadro de complementos para el agente de Pod Identity de EKS y, a continuación, elija Siguiente.
6. En la página Configurar las opciones de complementos seleccionados, selecciona cualquier versión de la lista desplegable Versión.
7. (Opcional) Expanda Valores de configuración opcionales para introducir una configuración adicional. Por ejemplo, puede proporcionar una ubicación de imagen de contenedor alternativa e `ImagePullSecrets`. El JSON Schema con las claves aceptadas se muestra en Esquema de configuración de complementos.

   Introduzca las claves y los valores de configuración en Valores de configuración.
8. Elija Siguiente.
9. Confirme que los pods de agente de Pod Identity de EKS se estén ejecutando en su clúster.

   ```
   kubectl get pods -n kube-system | grep 'eks-pod-identity-agent'
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   eks-pod-identity-agent-gmqp7                                          1/1
    Running   1 (24h ago)   24h
   eks-pod-identity-agent-prnsh                                          1/1
    Running   1 (24h ago)   24h
   ```

   Ahora puede usar las asociaciones de Pod Identity de EKS en su clúster. Para obtener más información, consulte Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM con Pod Identity de EKS.

AWS CLI

1. Ejecute el siguiente comando de la AWS CLI. Reemplace `my-cluster` por el nombre del clúster.

```
aws eks create-addon --cluster-name my-cluster --addon-name eks-pod-identity-
agent --addon-version v1.0.0-eksbuild.1
```

> **Note**
>
> El agente de Pod Identity de EKS no utiliza el `service-account-role-arn` en Roles de IAM para cuentas de servicio. Debe proporcionar al agente de Pod Identity de EKS los permisos en el rol de nodo.

2. Confirme que los pods de agente de Pod Identity de EKS se estén ejecutando en su clúster.

```
kubectl get pods -n kube-system | grep 'eks-pod-identity-agent'
```

Un ejemplo de salida sería el siguiente.

```
eks-pod-identity-agent-gmqp7                                          1/1
 Running   1 (24h ago)   24h
eks-pod-identity-agent-prnsh                                          1/1
 Running   1 (24h ago)   24h
```

Ahora puede usar las asociaciones de Pod Identity de EKS en su clúster. Para obtener más información, consulte Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM con Pod Identity de EKS.

Actualización del agente de Pod Identity de Amazon EKS

Actualice el tipo de Amazon EKS del complemento. Si no ha agregado el tipo Amazon EKS del complemento a su clúster, consulte Creación del agente de Pod Identity de Amazon EKS.

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento de agente de Pod Identity de EKS.
3. Elija la pestaña Complementos.

4. Si hay disponible una nueva versión del complemento, el agente de Pod Identity de EKS tiene un botón denominado Actualizar versión. Seleccione Actualizar versión.
5. En la página Configurar el agente de Pod Identity de Amazon EKS, seleccione la nueva versión en la lista desplegable Versión.
6. Seleccione Guardar cambios.

   La actualización puede tardar varios segundos en completarse. A continuación, confirme que la versión del complemento se ha actualizado comprobando el Estado.

AWS CLI

1. Consulte qué versión del complemento está instalada en el clúster. Reemplace *my-cluster* por el nombre del clúster.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name eks-pod-identity-agent --query "addon.addonVersion" --output text
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v1.0.0-eksbuild.1
   ```

   Debe crear el complemento antes de poder actualizarlo mediante este procedimiento.
2. Actualice el complemento con la AWS CLI. Si desea utilizar AWS Management Console o `eksctl` para actualizar el complemento, consulte Actualización de un complemento. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado.

   - Reemplace *my-cluster* por el nombre del clúster.
   - Reemplace *v1.0.0-eksbuild.1* por la versión que desee.
   - Reemplace *111122223333* por el ID de su cuenta.
   - Ejecute el siguiente comando:

     ```
     aws eks update-addon --cluster-name my-cluster --addon-name eks-pod-identity-agent --addon-version v1.0.0-eksbuild.1'
     ```

     La actualización puede tardar varios segundos en completarse.

3. Confirme que la versión del complemento se ha actualizado. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name eks-pod-identity-agent
```

La actualización puede tardar varios segundos en completarse.

Un ejemplo de salida sería el siguiente.

```
{
    "addon": {
        "addonName": "eks-pod-identity-agent",
        "clusterName": "my-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.0.0-eksbuild.1",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:region:111122223333:addon/my-cluster/eks-pod-identity-agent/74c33d2f-b4dc-8718-56e7-9fdfa65d14a9",
        "createdAt": "2023-04-12T18:25:19.319000+00:00",
        "modifiedAt": "2023-04-12T18:40:28.683000+00:00",
        "tags": {}
    }
}
```

Configuración del agente de Pod Identity de EKS

Desactivar **IPv6** en el agente de Pod Identity de EKS

AWS Management Console

Desactivar **IPv6** en la AWS Management Console

1. Para deshabilitar `IPv6` en el agente de Pod Identity de EKS, añada la siguiente configuración a los ajustes de configuración opcionales del complemento de EKS.

   a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

b. En el panel de navegación izquierdo, seleccione Clusters (Clústeres) y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento.

c. Elija la pestaña Complementos.

d. Seleccione la casilla situada en la parte superior derecha del cuadro de complementos para el agente de Pod Identity de EKS y, a continuación, elija Editar.

e. En la página de configuración del agente de Pod Identity de EKS, haga lo siguiente:

   i. Seleccione la Version (Versión) que desea utilizar. Le recomendamos que mantenga la misma versión que en el paso anterior y que actualice la versión y la configuración en acciones separadas.

   ii. Seleccione Ajustes de configuración opcionales.

   iii. Introduzca la clave JSON `"agent":` y el valor de un objeto JSON anidado con una clave `"additionalArgs":` en Valores de configuración. El texto resultante debe ser un objeto JSON válido. Si esta clave y este valor son los únicos datos del cuadro de texto, rodee la clave y el valor entre corchetes `{}`. En el siguiente ejemplo se muestra que la política de red está habilitada:

```
{
    "agent": {
        "additionalArgs": {
            "-b": "169.254.170.23"
        }
    }
}
```

   En esta configuración, se establece que la dirección `IPv4` sea la única dirección que utilice el agente.

f. Para aplicar la nueva configuración mediante la sustitución de los pods del agente de Pod Identity de EKS, seleccione Guardar cambios.

   Amazon EKS aplica los cambios a los complementos de EKS mediante la implementación de los Kubernetes `DaemonSet` para el agente de Pod Identity de EKS. Puede hacer un seguimiento del estado del lanzamiento en el historial de actualizaciones en la AWS Management Console y con `kubectl rollout status daemonset/eks-pod-identity-agent --namespace kube-system`.

   `kubectl rollout` tiene los siguientes comandos:

```
$ kubectl rollout

history  -- View rollout history
pause    -- Mark the provided resource as paused
restart  -- Restart a resource
resume   -- Resume a paused resource
status   -- Show the status of the rollout
undo     -- Undo a previous rollout
```

Si la implementación lleva demasiado tiempo, Amazon EKS la anulará y se agregará al historial de actualizaciones del complemento un mensaje con el tipo de actualización del complemento y el estado Fallido. Para investigar cualquier problema, comience por el historial de la implementación y ejecute `kubectl logs` en un pod del agente de Pod Identity de EKS para ver los registros del agente de Pod Identity de EKS.

2. Si la nueva entrada en el historial de actualizaciones tiene el estado Correcto, esto significa que la implementación se ha completado y que el complemento está utilizando la nueva configuración en todos los pods del agente de Pod Identity de EKS.

AWS CLI

Desactivar **`IPv6`** en la AWS CLI

- Para deshabilitar `IPv6` en el agente de Pod Identity de EKS, agregue la siguiente configuración a los valores de configuración del complemento de EKS.

  Ejecute el siguiente comando de la AWS CLI. Reemplace `my-cluster` por el nombre del clúster y el ARN del rol de IAM por el rol que va a usar.

  ```
  aws eks update-addon --cluster-name my-cluster --addon-name eks-pod-identity-
  agent \
      --resolve-conflicts PRESERVE --configuration-values '{"agent":
  {"additionalArgs": { "-b": "169.254.170.23"}}}'
  ```

  En esta configuración, se establece que la dirección `IPv4` sea la única dirección que utilice el agente.

Amazon EKS aplica los cambios a los complementos de EKS mediante la implementación de los Kubernetes DaemonSet para el agente de Pod Identity de EKS. Puede hacer un seguimiento del estado del lanzamiento en el historial de actualizaciones en la AWS Management Console y con `kubectl rollout status daemonset/eks-pod-identity-agent --namespace kube-system`.

`kubectl rollout` tiene los siguientes comandos:

```
kubectl rollout

history  -- View rollout history
pause    -- Mark the provided resource as paused
restart  -- Restart a resource
resume   -- Resume a paused resource
status   -- Show the status of the rollout
undo     -- Undo a previous rollout
```

Si la implementación lleva demasiado tiempo, Amazon EKS la anulará y se agregará al historial de actualizaciones del complemento un mensaje con el tipo de actualización del complemento y el estado Fallido. Para investigar cualquier problema, comience por el historial de la implementación y ejecute `kubectl logs` en un pod del agente de Pod Identity de EKS para ver los registros del agente de Pod Identity de EKS.

Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM con Pod Identity de EKS

En este tema se explica cómo configurar una cuenta de servicio de Kubernetes para asumir un rol de AWS Identity and Access Management (IAM) con el Pod Identity de EKS. Los Pods que estén configurados para usar la cuenta de servicio pueden acceder a cualquier Servicio de AWS al que el rol tenga permisos para acceder.

Para crear una asociación de Pod Identity de EKS, solo hay un paso: se crea la asociación en EKS a través de la AWS Management Console, la AWS CLI, los SDK de AWS, la AWS CloudFormation y otras herramientas. No existen datos ni metadatos sobre las asociaciones dentro del clúster en ningún objeto de Kubernetes y no se agrega ninguna anotación a las cuentas de servicio.

Requisitos previos

- Un clúster existente. Si no tiene uno, puede crearlo mediante una de las siguientes guías de Introducción a Amazon EKS.
- La entidad principal de IAM que va a crear la asociación debe tener `iam:PassRole`.
- La última versión de AWS CLI instalada y configurada en su dispositivo o AWS CloudShell. Puede comprobar su versión actual con `aws --version | cut -d / -f2 | cut -d ' ' -f1`. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con `aws configure` en la Guía del usuario de AWS Command Line Interface. La versión de la AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- Un archivo `config` de `kubectl` existente que contenga la configuración del clúster. Para crear un archivo `config` de `kubectl`, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

Creación de la asociación de Pod Identity de EKS

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento de agente de Pod Identity de EKS.
3. Elija la pestaña Acceder.
4. En las Asociaciones de Pod Identity, elija Crear.
5. Para el Rol de IAM, seleccione el rol de IAM con los permisos que quiere que tenga la carga de trabajo.

> **Note**
>
> La lista solo contiene roles que tienen la siguiente política de confianza, que permite a Pod Identity de EKS utilizarlas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEksAuthToAssumeRoleForPodIdentity",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
```

`sts:AssumeRole`

Pod Identity de EKS usa `AssumeRole` para asumir el rol de IAM antes de pasar las credenciales temporales a sus pods.

`sts:TagSession`

Pod Identity de EKS usa `TagSession` para incluir etiquetas de sesión en las solicitudes para AWS STS.

Puede usar estas etiquetas en condition keys en la política de confianza para restringir qué cuentas de servicio, espacios de nombres y clústeres pueden usar este rol.

Para obtener una lista de las claves de condición de Amazon EKS, consulte Condiciones de Amazon Elastic Kubernetes Service en la Referencia de autorizaciones de servicio. Para obtener más información sobre las acciones y los recursos con los que puede utilizar

una clave de condición, consulte Acciones definidas por Amazon Elastic Kubernetes Service.

6. Para el Espacio de nombres de Kubernetes, seleccione el espacio de nombres de Kubernetes que contiene la cuenta de servicio y la carga de trabajo. Si lo desea, puede especificar un espacio de nombres por nombre que no existe en el clúster.
7. Para la Cuenta de servicio de Kubernetes, seleccione la cuenta de servicio de Kubernetes que desee usar. El manifiesto de su carga de trabajo de Kubernetes debe especificar esta cuenta de servicio. Si lo desea, puede especificar una cuenta de servicio por nombre que no exista en el clúster.
8. (Opcional) Para las Etiquetas, elija Agregar etiqueta para agregar metadatos en un par clave-valor. Estas etiquetas se aplican a la asociación y se pueden utilizar en las políticas de IAM.

   Puede repetir este paso para agregar varias etiquetas.
9. Seleccione Crear.

AWS CLI

1. Si desea asociar una política de IAM existente a su rol de IAM, vaya al siguiente paso.

   Cree una política de IAM. Puede crear su propia política o copiar una política gestionada por AWS que ya conceda algunos de los permisos que necesita y personalizarla según sus requisitos específicos. Para obtener más información, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

   a. Cree un archivo que incluya los permisos para los Servicios de AWS a los que quiere que accedan sus Pods. Para obtener una lista de todas las acciones para todos los Servicios de AWS, consulte la referencia de autorizaciones de servicio.

      Puede ejecutar el siguiente comando para crear un archivo de política de ejemplo que permita el acceso de solo lectura a un bucket de Amazon S3. Opcionalmente, puede almacenar información de configuración o un script de arranque en este bucket, y los contenedores de su Pod pueden leer el archivo desde el bucket y cargarlo en su aplicación. Si desea crear esta política de ejemplo, copie el siguiente contenido en su dispositivo. Sustituya *my-pod-secrets-bucket* por el nombre de su bucket y ejecute el comando.

      ```
      cat >my-policy.json <<EOF
      ```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
        }
    ]
}
EOF
```

b. Cree la política de IAM.

```
aws iam create-policy --policy-name my-policy --policy-document file://my-policy.json
```

2. Cree un rol de IAM y asócielo a una cuenta de servicio de Kubernetes.

1. Si tiene una cuenta de servicio de Kubernetes existente que desea que asuma un rol de IAM, puede omitir este paso.

   Cree una cuenta de servicio de Kubernetes. Copie los siguientes contenidos en su dispositivo. Reemplace *my-service-account* por el nombre que desee y *default* por un espacio de nombres diferente, si es necesario. Si cambia *default*, el espacio de nombres debe existir previamente.

   ```
   cat >my-service-account.yaml <<EOF
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     name: my-service-account
     namespace: default
   EOF
   kubectl apply -f my-service-account.yaml
   ```

   Ejecute el siguiente comando de la .

   ```
   kubectl apply -f my-service-account.yaml
   ```

2. Ejecute el siguiente comando para crear un archivo de política de confianza para el rol de IAM.

```
cat >trust-relationship.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEksAuthToAssumeRoleForPodIdentity",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
EOF
```

3. Cree el rol. Reemplace *my-role* con un nombre para el rol de IAM, y *my-role-description* con una descripción de su función.

```
aws iam create-role --role-name my-role --assume-role-policy-document
 file://trust-relationship.json --description "my-role-description"
```

4. Adjunte la política de IAM al rol. Reemplace *my-role* por el nombre de su rol de IAM y *my-policy* por el nombre de una política existente que haya creado.

```
aws iam attach-role-policy --role-name my-role --policy-
arn=arn:aws:iam::111122223333:policy/my-policy
```

> **Note**
>
> A diferencia de los roles de IAM para cuentas de servicio, Pod Identity de EKS no utiliza ninguna anotación en la cuenta de servicio.

5. Ejecute el siguiente comando para crear la asociación. Reemplace `my-cluster` por el nombre del clúster, reemplace *my-service-account* por el nombre que desee y *default* por un espacio de nombres diferente, si es necesario.

```
aws eks create-pod-identity-association --cluster-name my-cluster --role-
arn arn:aws:iam::111122223333:role/my-role --namespace default --service-
account my-service-account
```

Un ejemplo de salida sería el siguiente.

```
{
    "association": {
        "clusterName": "my-cluster",
        "namespace": "default",
        "serviceAccount": "my-service-account",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws::111122223333:podidentityassociation/my-
cluster/a-abcdefghijklmnop1",
        "associationId": "a-abcdefghijklmnop1",
        "tags": {},
        "createdAt": 1700862734.922,
        "modifiedAt": 1700862734.922
    }
}
```

> **Note**
>
> Puede especificar un espacio de nombres y una cuenta de servicio por nombre que no existe en el clúster. Debe crear el espacio de nombres, la cuenta de servicio y la carga de trabajo que utiliza la cuenta de servicio para que funcione la asociación de Pod Identity de EKS.

3. Confirme que el rol y la cuenta de servicio se hayan configurado correctamente.

   a. Confirme que la política de confianza del rol de IAM se haya configurado correctamente.

   ```
   aws iam get-role --role-name my-role --query Role.AssumeRolePolicyDocument
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
   ```

```
        {
            "Sid": "Allow EKS Auth service to assume this role for Pod
 Identities",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
```

b. Confirme que la política que adjuntó a su rol en un paso anterior se encuentre adjunta al rol.

```
aws iam list-attached-role-policies --role-name my-role --query
 AttachedPolicies[].PolicyArn --output text
```

Un ejemplo de salida sería el siguiente.

```
arn:aws:iam::111122223333:policy/my-policy
```

c. Establezca una variable para almacenar el nombre de recurso de Amazon (ARN) de la política que quiera utilizar. Reemplace *my-policy* por el nombre de la política para la que desea confirmar los permisos.

```
export policy_arn=arn:aws:iam::111122223333:policy/my-policy
```

d. Vea la versión predeterminada de la política.

```
aws iam get-policy --policy-arn $policy_arn
```

Un ejemplo de salida sería el siguiente.

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "EXAMPLEBIOWGLDEXAMPLE",
```

```
        "Arn": "arn:aws:iam::111122223333:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        [...]
    }
}
```

e. Vea el contenido de la política para asegurarse de que incluye todos los permisos que su Pod necesita. Si es necesario, reemplace *1* en el siguiente comando por la versión devuelta en la salida anterior.

```
aws iam get-policy-version --policy-arn $policy_arn --version-id v1
```

Un ejemplo de salida sería el siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
        }
    ]
}
```

Si creó la política de ejemplo en un paso anterior, el resultado es el mismo. Si creó una política diferente, el contenido de *example* es diferente.

Siguiente paso

Configuración de Pods para usar una cuenta de servicio de Kubernetes

Configuración de Pods para usar una cuenta de servicio de Kubernetes

Si un Pod necesita acceder a los Servicios de AWS, debe configurarlo para que use una cuenta de servicio de Kubernetes. La cuenta de servicio debe estar asociada a un rol de AWS Identity and Access Management (IAM) que tenga permisos para acceder a Servicios de AWS.

Requisitos previos

- Un clúster existente. Si no tiene uno, puede crearlo mediante una de las guías de Introducción a Amazon EKS.
- Una cuenta de servicio de Kubernetes existente y una asociación de Pod Identity de EKS que asocia la cuenta de servicio a un rol de IAM. El rol debe tener una política de IAM asociada que contenga los permisos que desee que tengan sus Pods para usar Servicios de AWS. Para obtener más información acerca de cómo crear y configurar la cuenta de servicio y el rol, consulte Configuración de una cuenta de servicio de Kubernetes para asumir un rol de IAM con Pod Identity de EKS.
- La última versión de AWS CLI instalada y configurada en su dispositivo o AWS CloudShell. Puede comprobar su versión actual con `aws --version | cut -d / -f2 | cut -d ' ' -f1`. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con `aws configure` en la Guía del usuario de AWS Command Line Interface. La versión de la AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- Un archivo `config` de `kubectl` existente que contenga la configuración del clúster. Para crear un archivo `config` de `kubectl`, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

Cómo configurar un Pod para usar una cuenta de servicio

1. Use el siguiente comando para crear un manifiesto de implementación que puede implementar en un Pod con el que se puede confirmar la configuración. Sustituya *`example values`* con valores propios.

```
cat >my-deployment.yaml <<EOF
apiVersion: apps/v1
```

```
kind: Deployment
metadata:
  name: my-app
spec:
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      serviceAccountName: my-service-account
      containers:
      - name: my-app
        image: public.ecr.aws/nginx/nginx:X.XX
EOF
```

2. Implemente el manifiesto en el clúster.

```
kubectl apply -f my-deployment.yaml
```

3. Confirme que existan las variables de entorno necesarias para su Pod.

   a. Consulte los Pods que se implementaron en el paso anterior.

   ```
   kubectl get pods | grep my-app
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   my-app-6f4dfff6cb-76cv9   1/1     Running   0          3m28s
   ```

   b. Confirme que el Pod tiene un archivo de token de cuenta de servicio montado.

   ```
   kubectl describe pod my-app-6f4dfff6cb-76cv9 | grep
    AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE:
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE:  /var/run/secrets/
   pods.eks.amazonaws.com/serviceaccount/eks-pod-identity-token
   ```

4. Confirme que sus Pods puedan interactuar con los Servicios de AWS mediante los permisos que asignó en la política de IAM adjunta a su rol.

> **Note**
>
> Cuando un Pod utiliza credenciales de AWS de un rol de IAM asociado a una cuenta de servicio, la AWS CLI u otros SDK de los contenedores de ese Pod utilizan las credenciales proporcionadas por dicho rol. Si no restringe el acceso a las credenciales que se proporcionan al rol de IAM del nodo de Amazon EKS, el Pod sigue teniendo acceso a esas credenciales. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Si sus Pods no pueden interactuar con los servicios como esperaba, siga estos pasos para confirmar que todo esté configurado correctamente.

a. Confirme que los Pods usen una versión del SDK de AWS que admita asumir un rol de IAM a través de una asociación de Pod Identity de EKS. Para obtener más información, consulte Uso de un AWS SDK compatible.

b. Confirme que la implementación use la cuenta de servicio.

```
kubectl describe deployment my-app | grep "Service Account"
```

Un ejemplo de salida sería el siguiente.

```
Service Account:  my-service-account
```

Definición de permisos para que Pod Identities de EKS asuman roles basados en etiquetas

Pod Identity de EKS adjunta etiquetas a las credenciales temporales de cada pod con atributos como el nombre del clúster, el espacio de nombres y el nombre de la cuenta de servicio. Estas etiquetas de sesión de roles permiten a los administradores crear un único rol que funcione en todas las cuentas de servicio al permitir el acceso a los recursos de AWS en función de las etiquetas coincidentes. Al añadir la compatibilidad con las etiquetas de sesión de roles, los clientes pueden reforzar los límites

de seguridad entre los clústeres y las cargas de trabajo dentro de los clústeres, mientras reutilizan los mismos roles y políticas de IAM.

Por ejemplo, la siguiente política permite la acción `s3:GetObject` si el objeto está etiquetado con el nombre del clúster de EKS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/eks-cluster-name": "${aws:PrincipalTag/eks-
cluster-name}"
                }
            }
        }
    ]
}
```

Lista de etiquetas de sesión agregadas por Pod Identity de EKS

La siguiente lista contiene todas las claves de las etiquetas que se añaden a la solicitud `AssumeRole` realizada por Amazon EKS. Para usar estas etiquetas en las políticas, utilice `${aws:PrincipalTag/` seguido de la clave, por ejemplo `${aws:PrincipalTag/kubernetes-namespace}`.

- `eks-cluster-arn`
- `eks-cluster-name`

- `kubernetes-namespace`
- `kubernetes-service-account`
- `kubernetes-pod-name`
- `kubernetes-pod-uid`

Etiquetas entre cuentas

Todas las etiquetas de sesión que agrega Pod Identity de EKS son transitivas; las claves y los valores de las etiquetas se transfieren a cualquier acción `AssumeRole` que sus cargas de trabajo utilicen para cambiar los roles a otra cuenta. Puede utilizar estas etiquetas en las políticas de otras cuentas para limitar el acceso en situaciones entre cuentas. Para obtener más información, consulte Encadenar roles con etiquetas de sesión en la Guía del usuario de IAM.

Etiquetas personalizadas

Pod Identity de EKS no puede agregar etiquetas personalizadas adicionales a la acción `AssumeRole` que realiza. Sin embargo, las etiquetas que se aplican al rol de IAM siempre están disponibles en el mismo formato: `${aws:PrincipalTag/` seguido de la clave, por ejemplo `${aws:PrincipalTag/MyCustomTag}`.

> **Note**
>
> Las etiquetas agregadas a la sesión mediante la solicitud de `sts:AssumeRole` tienen prioridad en caso de conflicto. Por ejemplo, supongamos que Amazon EKS añade una clave `eks-cluster-name` y un valor `my-cluster` a la sesión cuando EKS asume el rol de cliente. También ha añadido una etiqueta `eks-cluster-name` con valor `my-own-cluster` al rol de IAM. En este caso, prevalecerá lo primero y el valor de la etiqueta `eks-cluster-name` será `my-cluster`.

Uso de un AWS SDK compatible

> **Important**
>
> Una versión anterior de la documentación era incorrecta. El SDK de AWS para Java v1 no es compatible con Pod Identity de EKS.

Para usar Pod Identities de EKS, los contenedores de los Pods deben utilizar una versión del SDK de AWS que admita asumir un rol de IAM del agente de Pod Identity de EKS. Asegúrese de usar las siguientes versiones, o posteriores, para el SDK de AWS:

- Java (Versión 2): 2.21.30
- Go v1: v1.47.11
- Go v2: lanzamiento-2023-11-14
- Python (Boto3): 1.34.41
- Python (botocore): 1.34.41
- AWS CLI: 1.30.0

  AWS CLI: 2.15.0
- JavaScript v2: 2.1550.0
- JavaScript v3: v3.458.0
- Kotlin: v1.0.1
- Ruby: 3.188.0
- Rust: lanzamiento-2024-03-13
- C++: 1.11.263
- .NET: 3.7.734.0
- PowerShell: 4.1.502
- PHP: 3.287.1

Para asegurarse de que esté utilizando un SDK compatible, siga las instrucciones de instalación para su SDK preferido en Herramientas para crear en AWS al crear los contenedores.

Para obtener una lista de los complementos compatibles con Pod Identity de EKS, consulte Versiones de complementos compatibles con Pod Identity de EKS.

Uso de las credenciales de Pod Identity de EKS

Para usar las credenciales de una asociación de Pod Identity de EKS, el código puede usar cualquier SDK de AWS para crear un cliente para un servicio de AWS con un SDK y, de forma predeterminada, el SDK busca en una cadena de ubicaciones las credenciales de AWS Identity and Access Management que desee utilizar. Las credenciales de Pod Identity de EKS se utilizarán si no especifica un proveedor de credenciales al crear el cliente o al iniciar el SDK de otro modo.

Esto funciona porque las Pod Identities de EKS se han agregado al proveedor de credenciales del contenedor, que se busca en un paso de la cadena de credenciales predeterminada. Si sus cargas de trabajo utilizan actualmente credenciales que se encuentran en una fase anterior de la cadena de credenciales, esas credenciales seguirán utilizándose aunque configure una asociación de Pod Identity de EKS para la misma carga de trabajo.

Para obtener más información sobre cómo funcionan las Pod Identities de EKS, consulte Cómo funciona Pod Identity de EKS.

Rol de Pod Identity de EKS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEksAuthToAssumeRoleForPodIdentity",
            "Effect": "Allow",
            "Principal": {
                "Service": "pods.eks.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:TagSession"
            ]
        }
    ]
}
```

`sts:AssumeRole`

Pod Identity de EKS usa `AssumeRole` para asumir el rol de IAM antes de pasar las credenciales temporales a sus pods.

`sts:TagSession`

Pod Identity de EKS usa `TagSession` para incluir etiquetas de sesión en las solicitudes para AWS STS.

Puede usar estas etiquetas en condition keys en la política de confianza para restringir qué cuentas de servicio, espacios de nombres y clústeres pueden usar este rol.

Para obtener una lista de las claves de condición de Amazon EKS, consulte Condiciones de Amazon Elastic Kubernetes Service en la Referencia de autorizaciones de servicio. Para obtener

más información sobre las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Elastic Kubernetes Service.

Roles de IAM para cuentas de servicio

Las aplicaciones de los contenedores de un Pod pueden usar un SDK de AWS o la AWS CLI para llevar a cabo solicitudes de API a Servicios de AWS mediante permisos de AWS Identity and Access Management (IAM). Las aplicaciones deben firmar sus solicitudes de API AWS con credenciales de AWS. Los roles de IAM para cuentas de servicio ofrecen la posibilidad de administrar las credenciales para las aplicaciones, de un modo similar a cómo los perfiles de instancia de Amazon EC2 proporcionan credenciales a instancias de Amazon EC2. En lugar de crear y distribuir las credenciales de AWS a los contenedores o de utilizar el rol de la instancia de Amazon EC2, puede asociar el rol de IAM con una cuenta de servicio de Kubernetes y configurar los Pods para usar la cuenta de servicio. No puede usar roles de IAM para cuentas de servicio con clústeres locales para Amazon EKS en AWS Outposts.

Los roles de IAM para cuentas de servicio ofrecen los siguientes beneficios:

- Privilegio mínimo: puede limitar los permisos de IAM a una cuenta de servicio y solo los Pods que utilizan esa cuenta de servicio tienen acceso a esos permisos. Esta característica también elimina la necesidad de soluciones de terceros como `kiam` o `kube2iam`.
- Aislamiento de credenciales: los contenedores de un Pod's solo pueden recuperar las credenciales para el rol de IAM asociado a la cuenta de servicio que usa el contenedor. Un contenedor nunca tiene acceso a credenciales que utilizan otros contenedores de otros Pods. Al utilizar roles de IAM para cuentas de servicio, los contenedores de Pod's también tienen los permisos asignados al rol de IAM del nodo de Amazon EKS, a menos que bloquee el acceso del Pod al servicio de metadatos de instancias (IMDS) de Amazon EC2. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.
- Auditabilidad: el acceso y el registro de eventos está disponible a través de AWS CloudTrail para garantizar una auditoría retrospectiva.

Siga estos procedimientos para habilitar los roles de IAM para cuentas de servicio:

1. Creación de un proveedor de OIDC de IAM para su clúster: solo complete este procedimiento una vez para cada clúster.

> **Note**
>
> Si habilita el punto de conexión de VPC de EKS, no se podrá acceder al punto de conexión del servicio OIDC de EKS desde el interior de esa VPC. Por lo tanto, no funcionarán operaciones tales como crear un proveedor de OIDC con `eksctl` en la VPC, y provocarán que se agote el tiempo de espera al intentar solicitar `https://oidc.eks.region.amazonaws.com`. A continuación se muestra un ejemplo de mensaje de error:
>
> ```
> ** server can't find oidc.eks.region.amazonaws.com: NXDOMAIN
> ```
>
> Para completar este paso, puede ejecutar el comando fuera de la VPC; por ejemplo, en AWS CloudShell o en un equipo conectado a Internet.

2. Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM: complete este procedimiento para cada conjunto único de permisos que desee que tenga una aplicación.
3. Configuración de Pods para usar una cuenta de servicio de Kubernetes: complete este procedimiento para cada Pod que necesite acceso a Servicios de AWS.
4. Uso de un SDK de AWS compatible— Confirme que la carga de trabajo utilice un SDK de AWS de una versión compatible y que utilice la cadena de credenciales predeterminada.

Información general de IAM, Kubernetes y OpenID Connect (OIDC)

En 2014, AWS Identity and Access Management agregó compatibilidad con identidades federadas mediante OpenID Connect (OIDC). Esta característica le permite autenticar llamadas a la API de AWS con proveedores de identidades compatibles y recibir un token web JSON (JWT) de OIDC válido. Puede transferir este token a la operación de API AWS STS`AssumeRoleWithWebIdentity` y recibir credenciales temporales del rol de IAM. Puede utilizar estas credenciales para interactuar con cualquier Servicio de AWS, como Amazon S3 y DynamoDB.

Cada token JWT está firmado por un par de claves de firma. Las claves se envían al proveedor de OIDC administrado por Amazon EKS y la clave privada cambia cada 7 días. Amazon EKS conserva las claves públicas hasta que caduquen. Si conecta clientes OIDC externos, tenga en cuenta que debe actualizar las claves de firma antes de que caduque la clave pública. Aprenda cómo the section called “Obtenga las claves de firma”.

Kubernetes ha usado durante mucho tiempo las cuentas de servicio como su propio sistema de identidad interno. Pods se puede autenticar con el servidor de la API de Kubernetes mediante un token montado automáticamente (un JWT que no era OIDC) que solo podía validar el servidor de la API de Kubernetes. Estos tokens de cuenta de servicio heredados no caducan, y rotar la clave de firma es un proceso difícil. En Kubernetes, versión `1.12`, se agregó compatibilidad para una nueva característica de `ProjectedServiceAccountToken`. Esta característica es un token web JSON de OIDC que también contiene la identidad de la cuenta de servicio y permite una audiencia configurable.

Amazon EKS aloja un punto de conexión de detección de OIDC público por clúster que contiene las claves de firma para los tokens web JSON `ProjectedServiceAccountToken` a fin de que los sistemas externos como IAM puedan validar y aceptar los tokens de OIDC que emite Kubernetes.

Creación de un proveedor de OIDC de IAM para su clúster

Su clúster tiene una URL de emisor de OpenID Connect (OIDC) asociada. Para utilizar roles de AWS Identity and Access Management (IAM) para cuentas de servicio, debe existir un proveedor de OIDC de IAM para la URL del emisor de OIDC de su clúster.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

- Un archivo `config` de `kubectl` existente que contenga la configuración del clúster. Para crear un archivo `config` de `kubectl`, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

Puede crear un proveedor IAM de OIDC para el clúster mediante `eksctl` o la AWS Management Console.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Para crear un proveedor de identidades OIDC de IAM para su clúster con **`eksctl`**

1. Determine el ID de emisor de OIDC correspondiente a su clúster.

   Recupere el ID de emisor de OIDC de su clúster y almacénelo en una variable. Reemplace *`my-cluster`* por su propio valor.

   ```
   cluster_name=my-cluster
   ```

   ```
   oidc_id=$(aws eks describe-cluster --name $cluster_name --query
    "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5)
   ```

   ```
   echo $oidc_id
   ```

2. Determine si ya hay un proveedor de OIDC de IAM con el ID de su clúster en su cuenta.

   ```
   aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
   ```

   Si el comando anterior devuelve la salida, significa que ya tiene un proveedor IAM de OIDC para su clúster y puede ir al siguiente paso. Si no se devuelve ninguna salida, debe crear un proveedor de OIDC de IAM para su clúster.

3. Cree un proveedor de identidades de OIDC de IAM para su clúster con el siguiente comando.

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name --approve
```

> **Note**
>
> Si habilita el punto de conexión de VPC de EKS, no se podrá acceder al punto de conexión del servicio OIDC de EKS desde el interior de esa VPC. Por lo tanto, no funcionarán operaciones tales como crear un proveedor de OIDC con `eksctl` en la VPC, y provocarán que se agote el tiempo de espera al intentar solicitar `https://oidc.eks.region.amazonaws.com`. A continuación se muestra un ejemplo de mensaje de error:
>
> ```
> ** server can't find oidc.eks.region.amazonaws.com: NXDOMAIN
> ```
>
> Para completar este paso, puede ejecutar el comando fuera de la VPC; por ejemplo, en AWS CloudShell o en un equipo conectado a Internet.

AWS Management Console

Para crear un proveedor de identidades de OIDC de IAM para su clúster con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de la izquierda, seleccione Clústeres y, a continuación, seleccione el nombre de su clúster en la página Clusters (Clústeres).
3. En la sección de Details (Detalles) en la pestaña Overview (Resumen), anote el valor de la OpenID Connect provider URL (URL del proveedor de OpenID Connect).
4. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
5. En el panel de navegación, elija Identity Providers (Proveedores de identidad) en Access management (Administración de acceso). Si aparece un Proveedor que coincide con la URL de su clúster, ya cuenta con un proveedor para su clúster. Si no aparece un proveedor en la lista que coincida con la URL del clúster, debe crear uno.
6. Para crear un proveedor, elija Add Provider (Agregar proveedor).
7. En Provider type (Tipo de proveedor), seleccione OpenID Connect.

8. En Provider URL (URL de proveedor), pegue la URL del proveedor de OIDC para su clúster y, a continuación, elija Get thumbprint (Obtener huella digital).
9. En Audience (Público), ingrese **`sts.amazonaws.com`** y elija Add provider (Agregar proveedor).

Siguiente paso

Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM

Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM

En este tema se explica cómo configurar una cuenta de servicio de Kubernetes para asumir un rol de AWS Identity and Access Management (IAM). Los Pods que estén configurados para usar la cuenta de servicio pueden acceder a cualquier Servicio de AWS al que el rol tenga permisos para acceder.

Requisitos previos

- Un clúster existente. Si no tiene uno, puede crearlo mediante una de las siguientes guías de Introducción a Amazon EKS.
- Un proveedor de OpenID Connect (OIDC) de IAM existente para el clúster. Para saber si ya tiene un proveedor o cómo crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

- Un archivo `config` de `kubectl` existente que contenga la configuración del clúster. Para crear un archivo con `config` de `kubectl`, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

Para asociar el rol de IAM con una cuenta de servicio de Kubernetes

1. Si desea asociar una política de IAM existente a su rol de IAM, vaya al siguiente paso.

   Cree una política de IAM. Puede crear su propia política o copiar una política gestionada por AWS que ya conceda algunos de los permisos que necesita y personalizarla según sus requisitos específicos. Para obtener más información, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

   a. Cree un archivo que incluya los permisos para los Servicios de AWS a los que quiere que accedan sus Pods. Para obtener una lista de todas las acciones para todos los Servicios de AWS, consulte la referencia de autorizaciones de servicio.

      Puede ejecutar el siguiente comando para crear un archivo de política de ejemplo que permita el acceso de solo lectura a un bucket de Amazon S3. Opcionalmente, puede almacenar información de configuración o un script de arranque en este bucket, y los contenedores de su Pod pueden leer el archivo desde el bucket y cargarlo en su aplicación. Si desea crear esta política de ejemplo, copie el siguiente contenido en su dispositivo. Sustituya *`my-pod-secrets-bucket`* por el nombre de su bucket y ejecute el comando.

      ```
      cat >my-policy.json <<EOF
      {
          "Version": "2012-10-17",
          "Statement": [
              {
                  "Effect": "Allow",
                  "Action": "s3:GetObject",
                  "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
              }
          ]
      }
      EOF
      ```

   b. Cree la política de IAM.

```
aws iam create-policy --policy-name my-policy --policy-document file://my-policy.json
```

2. Cree un rol de IAM y asócielo a una cuenta de servicio de Kubernetes. Puede utilizar `eksctl` o AWS CLI.

   eksctl

   Requisito previo

   La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

   Reemplace *my-service-account* por el nombre de la cuenta de servicio de Kubernetes que desea que `eksctl` cree y asóciela con un rol de IAM. Reemplace *default* por el espacio de nombres en el que desea que `eksctl` cree la cuenta de servicio. Reemplace *my-cluster* por el nombre de su clúster. Reemplace *my-role* por el nombre del rol al que desea asociar la cuenta de servicio. Si no existe, `eksctl` lo crea por usted. Reemplace *111122223333* por el ID de su cuenta y *my-policy* por el nombre de una política existente.

   ```
   eksctl create iamserviceaccount --name my-service-account --namespace default --cluster my-cluster --role-name my-role \
       --attach-policy-arn arn:aws:iam::111122223333:policy/my-policy --approve
   ```

   > **Important**
   >
   > Si el rol o la cuenta de servicio ya existen, podría producirse un error al ejecutar el comando anterior. `eksctl` tiene diferentes opciones que puede proporcionar en esas situaciones. Para obtener más información, ejecute **`eksctl create iamserviceaccount --help`**.

   AWS CLI

   1. Si tiene una cuenta de servicio de Kubernetes existente que desea que asuma un rol de IAM, puede omitir este paso.

Cree una cuenta de servicio de Kubernetes. Copie los siguientes contenidos en su dispositivo. Reemplace *my-service-account* por el nombre que desee y *default* por un espacio de nombres diferente, si es necesario. Si cambia *default*, el espacio de nombres debe existir previamente.

```
cat >my-service-account.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-service-account
  namespace: default
EOF
kubectl apply -f my-service-account.yaml
```

2. Establezca su ID de Cuenta de AWS en una variable de entorno con el siguiente comando.

```
account_id=$(aws sts get-caller-identity --query "Account" --output text)
```

3. Establezca el proveedor de identidades de OIDC en una variable de entorno con el siguiente comando. Reemplace *my-cluster* por el nombre del clúster.

```
oidc_provider=$(aws eks describe-cluster --name my-cluster --region
 $AWS_REGION --query "cluster.identity.oidc.issuer" --output text | sed -e "s/
^https:\/\///")
```

4. Establezca variables para el espacio de nombres y el nombre de la cuenta de servicio. Reemplace *my-service-account* por la cuenta de servicio de Kubernetes que desea que asuma el rol. Reemplace *default* por el espacio de nombres de la cuenta de servicio.

```
export namespace=default
export service_account=my-service-account
```

5. Ejecute el siguiente comando para crear un archivo de política de confianza para el rol de IAM. Si quiere permitir que todas las cuentas de servicio de un espacio de nombres utilicen el rol, copie el siguiente contenido en su dispositivo. Reemplace *StringEquals* por **StringLike** y reemplace *$service_account* por *****. Puede agregar varias entradas en las condiciones `StringEquals` y `StringLike` para permitir que varias cuentas

de servicio o espacios de nombres asuman el rol. Para permitir que los roles de una Cuenta de AWS diferente a la de su clúster asuman el rol, consulte Permisos de IAM entre cuentas para más información.

```
cat >trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::$account_id:oidc-provider/$oidc_provider"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "$oidc_provider:aud": "sts.amazonaws.com",
          "$oidc_provider:sub": "system:serviceaccount:
$namespace:$service_account"
        }
      }
    }
  ]
}
EOF
```

6. Cree el rol. Reemplace *my-role* con un nombre para el rol de IAM, y *my-role-description* con una descripción de su función.

```
aws iam create-role --role-name my-role --assume-role-policy-document
 file://trust-relationship.json --description "my-role-description"
```

7. Adjunte la política de IAM al rol. Reemplace *my-role* por el nombre de su rol de IAM y *my-policy* por el nombre de una política existente que haya creado.

```
aws iam attach-role-policy --role-name my-role --policy-arn=arn:aws:iam::
$account_id:policy/my-policy
```

8. Anote su cuenta de servicio con el nombre de recurso de Amazon (ARN) del rol de IAM que desea que asuma la cuenta de servicio. Reemplace *my-role* por el nombre de su rol de IAM existente. Supongamos que permitió que un rol de una Cuenta de AWS diferente a la de su clúster asumiera el rol en un paso anterior. A continuación, asegúrese

de especificar la Cuenta de AWS y el rol de la otra cuenta. Para obtener más información, consulte Permisos de IAM entre cuentas.

```
kubectl annotate serviceaccount -n $namespace $service_account
 eks.amazonaws.com/role-arn=arn:aws:iam::$account_id:role/my-role
```

3. Confirme que el rol y la cuenta de servicio se hayan configurado correctamente.

   a. Confirme que la política de confianza del rol de IAM se haya configurado correctamente.

```
aws iam get-role --role-name my-role --query Role.AssumeRolePolicyDocument
```

   Un ejemplo de salida sería el siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:default:my-
service-account",
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
                }
            }
        }
    ]
}
```

   b. Confirme que la política que adjuntó a su rol en un paso anterior se encuentre adjunta al rol.

```
aws iam list-attached-role-policies --role-name my-role --query
 AttachedPolicies[].PolicyArn --output text
```

Un ejemplo de salida sería el siguiente.

```
arn:aws:iam::111122223333:policy/my-policy
```

c. Establezca una variable para almacenar el nombre de recurso de Amazon (ARN) de la política que quiera utilizar. Reemplace *my-policy* por el nombre de la política para la que desea confirmar los permisos.

```
export policy_arn=arn:aws:iam::111122223333:policy/my-policy
```

d. Vea la versión predeterminada de la política.

```
aws iam get-policy --policy-arn $policy_arn
```

Un ejemplo de salida sería el siguiente.

```
{
    "Policy": {
        "PolicyName": "my-policy",
        "PolicyId": "EXAMPLEBIOWGLDEXAMPLE",
        "Arn": "arn:aws:iam::111122223333:policy/my-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        [...]
    }
}
```

e. Vea el contenido de la política para asegurarse de que incluye todos los permisos que su Pod necesita. Si es necesario, reemplace *1* en el siguiente comando por la versión devuelta en la salida anterior.

```
aws iam get-policy-version --policy-arn $policy_arn --version-id v1
```

Un ejemplo de salida sería el siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-pod-secrets-bucket"
        }
    ]
}
```

Si creó la política de ejemplo en un paso anterior, el resultado es el mismo. Si creó una política diferente, el contenido de *example* es diferente.

f. Confirme que la cuenta de servicio de Kubernetes se anota con el rol.

```
kubectl describe serviceaccount my-service-account -n default
```

Un ejemplo de salida sería el siguiente.

```
Name:                my-service-account
Namespace:           default
Annotations:         eks.amazonaws.com/role-arn:
 arn:aws:iam::111122223333:role/my-role
Image pull secrets:  <none>
Mountable secrets:   my-service-account-token-qqjfl
Tokens:              my-service-account-token-qqjfl
[...]
```

4. (Opcional) Configure el punto de conexión AWS Security Token Service de una cuenta de servicio. AWS recomienda el uso de un punto de conexión de AWS STS en lugar del global. Esto reduce la latencia, proporciona redundancia integrada y aumenta la validez de los tokens de sesión.

Siguiente paso

Configuración de Pods para usar una cuenta de servicio de Kubernetes

Configuración de Pods para usar una cuenta de servicio de Kubernetes

Si un Pod necesita acceder a los Servicios de AWS, debe configurarlo para que use una cuenta de servicio de Kubernetes. La cuenta de servicio debe estar asociada a un rol de AWS Identity and Access Management (IAM) que tenga permisos para acceder a Servicios de AWS.

Requisitos previos

- Un clúster existente. Si no tiene uno, puede crearlo mediante una de las guías de Introducción a Amazon EKS.
- Un proveedor de OpenID Connect (OIDC) de IAM existente para el clúster. Para saber si ya tiene un proveedor o cómo crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- Una cuenta de servicio de Kubernetes asociada a un rol de IAM. La cuenta de servicio debe estar anotada con el nombre de recurso de Amazon (ARN) del rol de IAM. El rol debe tener una política de IAM asociada que contenga los permisos que desee que tengan sus Pods para usar Servicios de AWS. Para obtener más información acerca de cómo crear y configurar la cuenta de servicio y el rol, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- Un archivo `config` de `kubectl` existente que contenga la configuración del clúster. Para crear un archivo `config` de `kubectl`, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

Cómo configurar un Pod para usar una cuenta de servicio

1. Use el siguiente comando para crear un manifiesto de implementación que puede implementar en un Pod con el que se puede confirmar la configuración. Sustituya *example values* con valores propios.

```
cat >my-deployment.yaml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      serviceAccountName: my-service-account
      containers:
      - name: my-app
        image: public.ecr.aws/nginx/nginx:X.XX
EOF
```

2. Implemente el manifiesto en el clúster.

```
kubectl apply -f my-deployment.yaml
```

3. Confirme que existan las variables de entorno necesarias para su Pod.

   a. Consulte los Pods que se implementaron en el paso anterior.

   ```
   kubectl get pods | grep my-app
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   my-app-6f4dfff6cb-76cv9   1/1     Running   0          3m28s
   ```

   b. Consulte el ARN del rol de IAM que esté utilizando el Pod.

```
kubectl describe pod my-app-6f4dfff6cb-76cv9 | grep AWS_ROLE_ARN:
```

Un ejemplo de salida sería el siguiente.

```
AWS_ROLE_ARN:                 arn:aws:iam::111122223333:role/my-role
```

El ARN del rol debe coincidir con el ARN del rol con el que anotó la cuenta de servicio existente. Para obtener más información sobre cómo anotar la cuenta de servicio, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

c. Confirme que el Pod tenga un montaje de archivo de token de identidad web.

```
kubectl describe pod my-app-6f4dfff6cb-76cv9 | grep
 AWS_WEB_IDENTITY_TOKEN_FILE:
```

Un ejemplo de salida sería el siguiente.

```
AWS_WEB_IDENTITY_TOKEN_FILE:  /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
```

El `kubelet` solicita y almacena el token en nombre del Pod. De forma predeterminada, el `kubelet` actualiza el token si tiene más del 80 % de su tiempo de vida total o más de 24 horas. Puede modificar la duración del vencimiento de cualquier cuenta, excepto la cuenta de servicio predeterminada, con la configuración en la especificación del Pod. Para obtener más información, consulte Proyección del volumen del token de la cuenta de servicio en la documentación de Kubernetes.

El webhook de Pod Identity de Amazon EKS del clúster observa los Pods que usan una cuenta de servicio con la siguiente anotación:

```
eks.amazonaws.com/role-arn: arn:aws:iam::111122223333:role/my-role
```

El webhook aplica las variables de entorno anteriores a esos Pods. El clúster no necesita utilizar el webhook para configurar las variables de entorno y los montajes de archivos de token. Puede configurar manualmente el Pods para que tenga estas variables de entorno. Las versiones compatibles del SDK de AWS buscan primero estas variables de entorno

en el proveedor de la cadena de credenciales. Las credenciales del rol se utilizan para los Pods que cumplen estos criterios.

4. Confirme que sus Pods puedan interactuar con los Servicios de AWS mediante los permisos que asignó en la política de IAM adjunta a su rol.

> **Note**
>
> Cuando un Pod utiliza credenciales de AWS de un rol de IAM asociado a una cuenta de servicio, la AWS CLI u otros SDK de los contenedores de ese Pod utilizan las credenciales proporcionadas por dicho rol. Si no restringe el acceso a las credenciales que se proporcionan al rol de IAM del nodo de Amazon EKS, el Pod sigue teniendo acceso a esas credenciales. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Si sus Pods no pueden interactuar con los servicios como esperaba, siga estos pasos para confirmar que todo esté configurado correctamente.

a. Confirme que los Pods usen una versión del SDK de AWS que admita asumir un rol de IAM a través de un archivo de token de identidad web de OpenID Connect. Para obtener más información, consulte Uso de un SDK de AWS compatible.

b. Confirme que la implementación use la cuenta de servicio.

```
kubectl describe deployment my-app | grep "Service Account"
```

Un ejemplo de salida sería el siguiente.

```
Service Account:  my-service-account
```

c. Si sus Pods siguen sin poder acceder a los servicios, revise los pasos que se describen en Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM para confirmar que el rol y la cuenta de servicio se hayan configurado correctamente.

Configure el punto de conexión AWS Security Token Service de una cuenta de servicio

Si utiliza una cuenta de servicio de Kubernetes con Roles de IAM para cuentas de servicio, puede configurar el tipo de punto de conexión de AWS Security Token Service utilizado por la cuenta de servicio si el clúster y la versión de la plataforma son iguales o posteriores a las enumeradas en la tabla siguiente. Si su versión de Kubernetes o plataforma son anteriores a las enumeradas en la tabla, sus cuentas de servicio solo pueden utilizar el punto de conexión global.

Versión de Kubernetes	Versión de la plataforma	Tipo de punto de conexión predeterminado
1.30	eks.2	Regional
1.29	eks.1	Regional
1.28	eks.1	Regional
1.27	eks.1	Regional
1.26	eks.1	Regional
1.25	eks.1	Regional
1.24	eks.2	Regional
1.23	eks.1	Regional

AWS recomienda usar el sistema regional de puntos de conexión de AWS STS en lugar del global. Esto reduce la latencia, proporciona redundancia integrada y aumenta la validez de los tokens de sesión. La AWS Security Token Service debe estar activa en la Región de AWS donde Pod se ejecuta. Además, su aplicación debe tener incorporada una redundancia para una Región de AWS diferente en caso de error del servicio en la Región de AWS. Para obtener más información, consulte Administración de AWS STS en una Región de AWS en la guía del usuario de IAM.

Requisitos previos

- Un clúster existente. Si no tiene uno, puede crearlo mediante una de las guías de Introducción a Amazon EKS.

- Un proveedor de OIDC de IAM existente para el clúster. Para obtener más información, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- Una cuenta de servicio de Kubernetes existente configurada para su uso con la característica IAM de Amazon EKS para cuentas de servicio.

Para configurar el tipo de punto de conexión utilizado por una cuenta de servicio de Kubernetes

Todos los siguientes ejemplos utilizan la cuenta de servicio de `aws-node` de Kubernetes utilizada por el Complemento CNI de Amazon VPC. Puede reemplazar los *`example values`* con sus propias cuentas de servicio, Pods, espacios de nombres y otros recursos.

1. Seleccione un Pod que utilice una cuenta de servicio para la que desee cambiar el punto de conexión. Determine en qué Región de AWS se ejecuta el Pod. Reemplace *`aws-node-6mfgv`* con su nombre de Pod y *`kube-system`* con el espacio de nombres de Pod.

   ```
   kubectl describe pod aws-node-6mfgv -n kube-system |grep Node:
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   ip-192-168-79-166.us-west-2/192.168.79.166
   ```

   En la salida anterior, el Pod se ejecuta en un nodo de la us-west-2 Región de AWS.
2. Determine el tipo de punto de conexión que utiliza la cuenta de servicio del Pod's.

   ```
   kubectl describe pod aws-node-6mfgv -n kube-system |grep AWS_STS_REGIONAL_ENDPOINTS
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   AWS_STS_REGIONAL_ENDPOINTS: regional
   ```

   Si el punto de conexión actual es global, `global` se devuelven en la salida. Si no se devuelve ningún resultado, el tipo de punto de conexión predeterminado está en uso y no se ha anulado.
3. Si la versión de clúster o plataforma es la misma o posterior a las enumeradas en la tabla, puede cambiar el tipo de punto de conexión utilizado por la cuenta de servicio del tipo predeterminado a otro con uno de los siguientes comandos. Reemplace *`aws-node`* con el nombre de su cuenta de servicio y *`kube-system`* con el espacio de nombres de su cuenta de servicio.

- Si el tipo de punto de conexión predeterminado o actual es global y desea cambiarlo a regional:

```
kubectl annotate serviceaccount -n kube-system aws-node eks.amazonaws.com/sts-regional-endpoints=true
```

Si utiliza Roles de IAM para cuentas de servicio para generar URL de S3 prefirmadas en la aplicación que se ejecuta en los contenedores de los Pods, el formato de la URL de los puntos de conexión regionales es similar al siguiente ejemplo:

```
https://bucket.s3.us-west-2.amazonaws.com/path?...&X-Amz-Credential=your-access-key-id/date/us-west-2/s3/aws4_request&...
```

- Si el tipo de punto de conexión predeterminado o actual es regional y desea cambiarlo a global:

```
kubectl annotate serviceaccount -n kube-system aws-node eks.amazonaws.com/sts-regional-endpoints=false
```

Si su aplicación realiza solicitudes explícitamente a puntos de enlace globales de AWS STS y no anula el comportamiento predeterminado de usar puntos de enlace regionales en clústeres de Amazon EKS, las solicitudes fallarán y mostrarán un error. Para obtener más información, consulte Los contenedores de pods muestran el siguiente error: `An error occurred (SignatureDoesNotMatch) when calling the GetCallerIdentity operation: Credential should be scoped to a valid region`.

Si utiliza Roles de IAM para cuentas de servicio para generar URL de S3 prefirmadas en la aplicación que se ejecuta en los contenedores de los Pods, el formato de la URL de los puntos de conexión globales es similar al siguiente ejemplo:

```
https://bucket.s3.amazonaws.com/path?...&X-Amz-Credential=your-access-key-id/date/us-west-2/s3/aws4_request&...
```

Si tiene una automatización que espera la URL prefirmada en un formato determinado o si su aplicación o dependencias posteriores que utilizan URL prefirmadas tienen expectativas para la Región de AWSsegmentada, luego realice los cambios necesarios para utilizar el punto de conexión AWS STS.

4. Elimine y vuelva a crear todos los Pods existentes asociados a la cuenta de servicio para aplicar las variables de entorno de credenciales. El enlace web que muta no se aplica a los Pods que ya están en ejecución. Puede reemplazar *Pods*, *kube-system* y *-l k8s-app=aws-node* con la información del Pods para el que configuró la anotación.

```
kubectl delete Pods -n kube-system -l k8s-app=aws-node
```

5. Confirme que todos los Pods se reiniciaron.

```
kubectl get Pods -n kube-system -l k8s-app=aws-node
```

6. Consulte las variables de entorno de uno de los Pods. Compruebe que el valor `AWS_STS_REGIONAL_ENDPOINTS` sea el que estableció en un paso anterior.

```
kubectl describe pod aws-node-kzbtr -n kube-system |grep AWS_STS_REGIONAL_ENDPOINTS
```

Un ejemplo de salida sería el siguiente.

```
AWS_STS_REGIONAL_ENDPOINTS=regional
```

Permisos de IAM entre cuentas

Puede configurar permisos de IAM entre cuentas mediante la creación de un proveedor de identidades a partir del clúster de otra cuenta o mediante operaciones `AssumeRole` encadenadas. En los siguientes ejemplos, la Cuenta A posee un clúster de Amazon EKS que admite roles de IAM para las cuentas de servicio. Los Pods que se ejecutan en ese clúster deben asumir permisos de IAM de la Cuenta B.

Example Creación de un proveedor de identidades a partir del clúster de otra cuenta

Example

En este ejemplo, la Cuenta A proporciona a la Cuenta B la URL del emisor de OpenID Connect (OIDC) desde su clúster. La cuenta B sigue las instrucciones de Creación de un proveedor de OIDC de IAM para su clúster y Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM utilizando la URL del emisor OIDC del clúster de la cuenta A. A continuación, un administrador del clúster anota la cuenta de servicio en el clúster de la Cuenta A para utilizar el rol de la Cuenta B (*444455556666*).

```
apiVersion: v1
kind: ServiceAccount
metadata:
  annotations:
    eks.amazonaws.com/role-arn: arn:aws:iam::444455556666:role/account-b-role
```

Example Uso de operaciones **`AssumeRole`** encadenadas

Example

En este ejemplo, la cuenta B crea una política de IAM con los permisos que debe otorgar a los Pods del clúster de la cuenta A. La cuenta B (*444455556666*) adjunta dicha política a un rol de IAM con una relación de confianza que concede permisos de `AssumeRole` a la cuenta A (*111122223333*).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

La cuenta A crea un rol con una política de confianza que obtiene las credenciales del proveedor de identidades creado con la dirección del emisor OIDC del clúster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.region-
code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity"
```

```
    }
  ]
}
```

La cuenta A asocia una política a ese rol con los siguientes permisos para asumir el rol que ha creado la cuenta B.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::444455556666:role/account-b-role"
        }
    ]
}
```

El código de aplicación para que los Pods asuman el rol de la cuenta B utiliza dos perfiles: `account_b_role` y `account_a_role`. El perfil `account_b_role` utiliza el perfil `account_a_role` como origen. Para la AWS CLI, el archivo `~/.aws/config` es similar al siguiente.

```
[profile account_b_role]
source_profile = account_a_role
role_arn=arn:aws:iam::444455556666:role/account-b-role

[profile account_a_role]
web_identity_token_file = /var/run/secrets/eks.amazonaws.com/serviceaccount/token
role_arn=arn:aws:iam::111122223333:role/account-a-role
```

Para especificar perfiles encadenados para otros SDK de AWS, consulte la documentación del SDK que utiliza. Para obtener más información, consulte herramientas para crear en AWS.

Uso de un SDK de AWS compatible

Para usar Roles de IAM para cuentas de servicio, los contenedores de los Pods deben utilizar una versión del SDK de AWS que admita asumir un rol de IAM a través de un archivo de token de identidad web de OpenID Connect. Asegúrese de usar las siguientes versiones, o posteriores, para el SDK de AWS:

- Java (Versión 2): 2.10.11
- Java: 1.11.704
- Go: 1.23.13
- Python (Boto3): 1.9.220
- Python (botocore): 1.12.200
- AWS CLI: 1.16.232
- Nodo: 2.525.0 y 3.27.0
- Ruby: 3.58.0
- C++: 1.7.174
- .NET: 3.3.659.1: también debe incluir `AWSSDK.SecurityToken`.
- PHP: 3.110.7

Muchos complementos populares de Kubernetes, como el escalador automático de clústeres y el Amazon VPC CNI plugin for Kubernetes permiten utilizar roles de IAM para cuentas de servicio.

Para asegurarse de que esté utilizando un SDK compatible, siga las instrucciones de instalación para su SDK preferido en Herramientas para crear en AWS al crear los contenedores.

Uso de las credenciales

Para usar las credenciales de los roles de IAM para cuentas de servicio, el código puede usar cualquier SDK de AWS para crear un cliente para un servicio de AWS con un SDK y, de forma predeterminada, el SDK busca en una cadena de ubicaciones las credenciales de AWS Identity and Access Management para usar. Las credenciales de los roles de IAM para cuentas de servicio se utilizarán si no especifica un proveedor de credenciales al crear el cliente o al iniciar el SDK de otro modo.

Esto funciona porque los roles de IAM para cuentas de servicio se han agregado como un paso en la cadena de credenciales predeterminada. Si sus cargas de trabajo utilizan actualmente credenciales que se encuentran en una fase anterior de la cadena de credenciales, esas credenciales seguirán utilizándose aunque configure un rol de IAM para cuentas de servicio de la misma carga de trabajo.

El SDK intercambia automáticamente el token OIDC de la cuenta de servicio por credenciales temporales de AWS Security Token Service mediante la acción `AssumeRoleWithWebIdentity`. Amazon EKS y esta acción de SDK siguen rotando las credenciales temporales y las renuevan antes de que caduquen.

Obtenga las claves de firma

Kubernetes emite un `ProjectedServiceAccountToken` para cada Kubernetes de Service Account. Este token es un token OIDC, que, además, es un tipo de JSON web token (JWT). Amazon EKS aloja un punto de conexión OIDC público por cada clúster que contiene las claves de firma para el token así los sistemas externos pueden validarlo.

Para validar un `ProjectedServiceAccountToken`, necesita buscar las claves de firma pública OIDC, también conocidas como JSON Web Key Set (JWKS). Utilice estas claves en su aplicación para validar el token. Por ejemplo, puede usar la biblioteca Python PyJWT para validar los tokens con estas claves. Para más información sobre `ProjectedServiceAccountToken`, consulte the section called “Información general de IAM, Kubernetes y OpenID Connect (OIDC)”.

Requisitos previos

- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- AWS CLI: una herramienta de línea de comandos para trabajar con servicios de AWS, incluido Amazon EKS. Para obtener más información, consulte Instalar, actualizar y desinstalar la AWS CLI en la guía del usuario de AWS Command Line Interface. Después de instalar la AWS CLI, recomendamos que también la configure. Para obtener más información, consulte Configuración rápida con `aws configure` en la guía del usuario de AWS Command Line Interface.

Obtenga las claves de firma pública OIDC (AWS CLI)

1. Recupere la URL de OIDC del clúster de Amazon EKS con AWS CLI.

```
$ aws eks describe-cluster --name my-cluster --query 'cluster.identity.oidc.issuer'
"https://oidc.eks.us-west-2.amazonaws.com/id/8EBDXXXX00BAE"
```

2. Recupere la clave de firma pública con curl o una herramienta similar. El resultado es un JSON Web Key Set (JWKS).

Important

Amazon EKS limita las llamadas al punto de conexión OIDC. Debe almacenar en caché la clave de firma pública. Respete el encabezado `cache-control` incluido en la respuesta.

Important

Amazon EKS rota la clave de firma OIDC cada siete días.

```
$ curl https://oidc.eks.us-west-2.amazonaws.com/id/8EBDXXXX00BAE/keys
{"keys":
[{"kty":"RSA","kid":"2284XXXX4a40","use":"sig","alg":"RS256","n":"wklbXXXXMVfQ","e":"AQAB"}
```

Nodos de Amazon EKS

Un nodo de Kubernetes es una máquina que ejecuta aplicaciones en contenedores. Cada nodo tiene los siguientes componentes:

- Tiempo de ejecución de contenedores: software responsable de ejecutar los contenedores.
- **`kubelet`**: se asegura de que los contenedores estén en buen estado y funcionando dentro de su Pod asociado.
- **`kube-proxy`**: mantiene las reglas de red que permiten la comunicación con sus Pods.

Para obtener más información, consulte Nodos en la documentación de Kubernetes.

Su clúster de Amazon EKS puede programar Pods en cualquier combinación de nodos autoadministrados, grupos de nodos administrados por Amazon EKS y AWS Fargate. Para obtener más información sobre los nodos implementados en el clúster, consulte Vea los recursos de Kubernetes.

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

> **Note**
>
> Los nodos deben estar en la misma VPC que las subredes que seleccionó al crear el clúster. Sin embargo, los nodos no tienen que estar en las mismas subredes.

La siguiente tabla proporciona varios criterios para evaluar a la hora de decidir qué opciones se ajustan mejor a sus requisitos. Esta tabla no incluye nodos conectados que se han creado fuera de Amazon EKS, que solo se pueden ver.

> **Note**
>
> Bottlerocket presenta algunas diferencias específicas con respecto a la información general de esta tabla. Para obtener más información, consulte la documentación de Bottlerocket en GitHub.

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Se puede implementar en AWS Outposts.	No	Sí	No
Se puede implementar en AWS Local Zones.	No	Sí. Para obtener más información, consulte Amazon EKS y AWS Local Zones.	No
Puede ejecutar contenedores que requieran Windows.	Sí	Sí, su clúster requiere al menos un nodo de Linux (se recomiendan dos para tener disponibilidad).	No
Puede ejecutar contenedores que requieran Linux.	Sí	Sí	Sí
Puede ejecutar cargas de trabajo que requieren el chip de inferencias.	Sí. Solo nodos de Amazon Linux.	Sí. Solo Amazon Linux.	No

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Puede ejecutar cargas de trabajo que requieren una GPU.	Sí. Solo nodos de Amazon Linux.	Sí. Solo Amazon Linux.	No
Puede ejecutar cargas de trabajo que requieren procesadores Arm.	Sí	Sí	No
Puede ejecutar AWS Bottlerocket.	Sí	Sí	No
Los pods comparten un entorno en tiempo de ejecución del kernel con otros Pods.	Sí, todos los Pods en cada uno de los nodos	Sí, todos los Pods en cada uno de los nodos	No, cada Pod tiene un kernel dedicado.
Los pods comparten CPU, memoria, almacenamiento y recursos de red con otros Pods.	Sí. Puede hacer que no se utilicen recursos en cada nodo.	Sí. Puede hacer que no se utilicen recursos en cada nodo.	No, cada Pod tiene recursos dedicados y su tamaño se puede adaptar de forma independiente para maximizar la utilización de los recursos.
Los pods pueden utilizar más hardware y memoria que lo que se indica en las especificaciones del Pod.	Sí, si el Pod requiere más recursos que los solicitados y los recursos están disponibles en el nodo, el Pod puede utilizar recursos adicionales.	Sí, si el Pod requiere más recursos que los solicitados y los recursos están disponibles en el nodo, el Pod puede utilizar recursos adicionales.	No, el Pod se puede volver a implementar mediante una vCPU y una configuración de memoria más grandes.

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Debe implementar y administrar instancias de Amazon EC2.	Sí. Automatizado a través de Amazon EKS si ha implementado una AMI optimizada para Amazon EKS. Si ha implementado una AMI personalizada, debe actualizar la instancia de forma manual.	Sí, configuración manual o con plantillas de AWS CloudFormation proporcionadas por Amazon EKS para implementar nodos de Linux (x86), Linux (Arm) o Windows.	No
Debe proteger, mantener y aplicar parches al sistema operativo de las instancias de Amazon EC2.	Sí	Sí	No
Puede proporcionar argumentos de arranque en la implementación de un nodo, como argumentos `kubelet` adicionales.	Sí. Mediante `eksctl` o una plantilla de lanzamiento con una AMI personalizada	Sí. Para obtener más información, consulte la información de uso del script de arranque en GitHub.	No

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Puede asignar direcciones IP a Pods desde un bloque de CIDR diferente de la dirección IP asignada al nodo.	Sí. Mediante una plantilla de lanzamiento con una AMI personalizada. Para obtener más información, consulte Personalización de nodos administrados con plantillas de lanzamiento.	Sí. Para obtener más información, consulte Redes personalizadas para los pods.	No
Se puede utilizar SSH en el nodo.	Sí	Sí	No. No hay ningún sistema operativo host de nodos para utilizar SSH.
Puede implementar su propia AMI personalizada en los nodos.	Sí. Mediante una plantilla de lanzamiento.	Sí	No
Puede implementar su propia CNI personalizada en los nodos.	Sí. Mediante una plantilla de lanzamiento con una AMI personalizada.	Sí	No

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Debe actualizar la AMI de nodos por su cuenta	Sí. Si ha implementado una AMI optimizada de Amazon EKS, recibirá una notificación en la consola de Amazon EKS cuando las actualizaciones estén disponibles. Puede realizar la actualización con un solo clic de en la consola. Si ha implementado una AMI personalizada, no recibirá una notificación en la consola de Amazon EKS cuando las actualizaciones estén disponibles. Debe realizar la actualización por su cuenta.	Sí. Uso de herramientas distintas de la consola de Amazon EKS. Esto se debe a que los nodos autoadministrados no se pueden administrar con la consola de Amazon EKS.	No

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Debe actualizar la versión de Kubernetes de los nodos por su cuenta	Sí. Si ha implementado una AMI optimizada de Amazon EKS, recibirá una notificación en la consola de Amazon EKS cuando las actualizaciones estén disponibles. Puede realizar la actualización con un solo clic de en la consola. Si ha implementado una AMI personalizada, no recibirá una notificación en la consola de Amazon EKS cuando las actualizaciones estén disponibles. Debe realizar la actualización por su cuenta.	Sí. Uso de herramientas distintas de la consola de Amazon EKS. Esto se debe a que los nodos autoadministrados no se pueden administrar con la consola de Amazon EKS.	No. No administra nodos.

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Puede utilizar el almacenamiento de Amazon EBS con Pods.	Sí	Sí	No
Puede utilizar el almacenamiento de Amazon EFS con Pods.	Sí	Sí	Sí
Puede utilizar el almacenamiento de Amazon FSx para Lustre con Pods.	Sí	Sí	No
Puede utilizar el Network Load Balancer para servicios.	Sí	Sí	Sí, cuando se utiliza el Crear un equilibrador de carga de red.
Los pods pueden ejecutarse en una subred pública.	Sí	Sí	No
Puede asignar diferentes grupos de seguridad de VPC a Pods individuales.	Sí, solo nodos Linux	Sí, solo nodos Linux	Sí
Puede ejecutar Kubernetes DaemonSets	Sí	Sí	No
Soporte de `HostPort` y `HostNetwork` en el manifiesto de Pod.	Sí	Sí	No
Disponibilidad de Región de AWS	Todas las regiones admitidas por Amazon EKS	Todas las regiones admitidas por Amazon EKS	Algunas regiones admitidas por Amazon EKS
Puede ejecutar contenedores en hosts dedicados de Amazon EC2	Sí	Sí	No

Criterios	Grupos de nodos administrados por EKS	Nodos autoadministrados	AWS Fargate
Precios	Costo de la instancia de Amazon EC2 que ejecuta varios Pods. Para obtener más información, consulte Precios de Amazon EC2.	Costo de la instancia de Amazon EC2 que ejecuta varios Pods. Para obtener más información, consulte Precios de Amazon EC2.	Costo de una memoria de Fargate individual y configuración de CPU. Cada Pod tiene su propio costo. Para obtener más información, consulte Precios de AWS Fargate.

Grupos de nodos administrados

Los grupos de nodos administrados por Amazon EKS automatizan el aprovisionamiento y la administración del ciclo de vida de nodos (instancias de Amazon EC2) para clústeres de Kubernetes de Amazon EKS.

Con los grupos de nodos administrados por Amazon EKS, no es necesario aprovisionar ni registrar por separado las instancias de Amazon EC2 que proporcionan capacidad de computación para ejecutar sus aplicaciones de Kubernetes. Puede crear, actualizar o terminar nodos para el clúster con una sola operación y de manera automática. Las actualizaciones y terminaciones de nodos drenan de forma automática los nodos para garantizar que sus aplicaciones permanezcan disponibles.

Todos los nodos administrados se aprovisionan como parte de un grupo de Amazon EC2 Auto Scaling administrado por usted a través de Amazon EKS. Todos los recursos, incluidas las instancias y los grupos de Auto Scaling, se ejecutan dentro de su cuenta de AWS. Cada grupo de nodos puede ejecutarse en varias zonas de disponibilidad que defina.

Puede agregar un grupo de nodos administrado a clústeres nuevos o existentes mediante la consola de Amazon EKS, `eksctl`, la AWS CLI, la API de AWS o herramientas de infraestructura como código que incluyen AWS CloudFormation. Los nodos lanzados como parte de un grupo de nodos administrado se etiquetan automáticamente para la detección automática por el autoescalador de

clústeres de Kubernetes. Puede utilizar el grupo de nodos para aplicar etiquetas de Kubernetes a los nodos y actualizarlos en cualquier momento.

No incurre en costos adicionales por usar grupos de nodos administrados por Amazon EKS, solo paga por los recursos de AWS que aprovisione. Estos incluyen instancias de Amazon EC2, volúmenes de Amazon EBS, horas de clúster de Amazon EKS y cualquier otra infraestructura de AWS. No se requieren pagos mínimos ni compromisos iniciales.

Para comenzar a utilizar un grupo de nodos administrado y un clúster de Amazon EKS nuevos, consulte Introducción a Amazon EKS: AWS Management Console y AWS CLI.

Para agregar un grupo de nodos administrados a un clúster existente, consulte Creación de un grupo de nodos administrados.

Conceptos de grupos de nodos administrados

- Los grupos de nodos administrados por Amazon EKS crean y administran instancias de Amazon EC2 para usted.
- Todos los nodos administrados se aprovisionan como parte de un grupo de Amazon EC2 Auto Scaling administrado por usted a través de Amazon EKS. Además, todos los recursos, incluidas las instancias de Amazon EC2 y los grupos de Auto Scaling, se ejecutan dentro de su cuenta de AWS.
- El grupo de Auto Scaling de un grupo de nodos administrados abarca todas las subredes que especifique al crear el grupo.
- Las etiquetas de Amazon EKS administran recursos de grupo de nodos para que estén configurados para usar el escalador automático de clústeres de Kubernetes.

> **Important**
>
> Si está ejecutando una aplicación con estado en varias zonas de disponibilidad respaldadas por volúmenes de Amazon EBS y utilizando el Escalado automático de Kubernetes, debe configurar varios grupos de nodos, cada uno enfocado a una sola zona de disponibilidad. Además, debe habilitar la característica `--balance-similar-node-groups`.

- Puede utilizar una plantilla de lanzamiento personalizada para obtener un mayor nivel de flexibilidad y personalización al implementar nodos administrados. Por ejemplo, puede especificar argumentos `kubelet` adicionales y utilizar una AMI personalizada. Para obtener más información,

consulte Personalización de nodos administrados con plantillas de lanzamiento. Si no usa una plantilla de lanzamiento personalizada al crear un grupo de nodos administrados, hay una plantilla de lanzamiento generada automáticamente. No modifique manualmente esta plantilla generada automáticamente o se producirán errores.

- Amazon EKS sigue el modelo de responsabilidad compartida para CVE y los parches de seguridad en grupos de nodos administrados. Cuando los nodos administrados ejecutan una AMI optimizada para Amazon EKS, Amazon EKS es responsable de crear versiones con parches de la AMI cuando se informa de errores o problemas. Podemos publicar una solución. Sin embargo, es responsable de implementar estas versiones de AMI con parches en los grupos de nodos administrados. Cuando los nodos administrados ejecutan una AMI personalizada, es responsable de crear versiones con parches de la AMI cuando se informa de errores o problemas y, a continuación, implementar la AMI. Para obtener más información, consulte Actualización de un grupo de nodos administrados.
- Los grupos de nodos administrados por Amazon EKS se pueden lanzar tanto en subredes públicas como privadas. Si lanza un grupo de nodos administrado en una subred pública el 22 de abril de 2020 o después, la subred debe tener la opción `MapPublicIpOnLaunch` establecida en verdadero para que las instancias puedan unirse a un clúster correctamente. Si la subred pública se creó con `eksctl` o las plantillas de AWS CloudFormation ofrecidas por Amazon EKS el 26 de marzo de 2020 o después, esta configuración ya está establecida en verdadero. Si las subredes públicas se crearon antes del 26 de marzo de 2020, debe cambiar el valor de forma manual. Para obtener más información, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred.
- Al implementar un grupo de nodos administrado en subredes privadas, debe asegurarse de que pueda acceder a Amazon ECR para extraer imágenes de contenedores. Para ello, conecte una puerta de enlace de NAT a la tabla de enrutamiento de la subred o agregue los siguientes puntos de conexión de VPC de AWS PrivateLink:
  - Interfaz de punto de conexión de la API de Amazon ECR: `com.amazonaws.`*`region-code`*`.ecr.api`
  - Interfaz de punto de conexión de la API de registro de Docker de Amazon ECR: `com.amazonaws.`*`region-code`*`.ecr.dkr`
  - Punto de conexión de puerta de enlace de Amazon S3: `com.amazonaws.`*`region-code`*`.s3`

  Para ver otros servicios y puntos de conexión de uso común, consulte Requisitos del clúster privado.

- Los grupos de nodos administrados no se pueden implementar en AWS Outposts, AWS Wavelength o AWS Local Zones.
- Puede crear varios grupos de nodos administrados dentro de un único clúster. Por ejemplo, puede crear un grupo de nodos con la AMI optimizada de Amazon Linux para Amazon EKS estándar para algunas cargas de trabajo y otro con la variante de GPU para las cargas de trabajo que requieren compatibilidad con GPU.
- Si el grupo de nodos administrado encuentra un error de comprobación de estado de instancia de Amazon EC2, Amazon EKS devuelve un código de error para ayudarlo a diagnosticar el problema. Para obtener más información, consulte Códigos de error del grupo de nodos administrado.
- Amazon EKS agrega etiquetas de Kubernetes a instancias de grupos de nodos administrados. Estas etiquetas proporcionadas por Amazon EKS tienen el prefijo `eks.amazonaws.com`.
- Amazon EKS drena nodos automáticamente a través de la API de Kubernetes durante las terminaciones o actualizaciones.
- Los presupuestos de interrupción del pod no se respetan al terminar un nodo con `AZRebalance` o al reducir el número de nodos deseado. Estas acciones intentan desalojar los Pods en el nodo. Con estas acciones, se intentan expulsar los del nodo, pero si tardan más de 15 minutos, el nodo se termina independientemente de si todos los Pods del nodo están terminados. Para ampliar el período hasta que finalice el nodo, agregue un enlace de ciclo de vida al grupo de escalado automático. Para obtener más información, consulte Agregar enlaces de enlace de ciclo de vida en la guía de hombre del usuario de Amazon EC2 Auto Scaling.
- Para ejecutar el proceso de drenaje correctamente después de recibir una notificación de interrupción puntual o una notificación de reequilibrio de capacidad, `CapacityRebalance` debe configurarse en `true`.
- La actualización de los grupos de nodos administrados respetan los presupuestos de interrupción Pod que ha establecido para los Pods. Para obtener más información, consulte Comportamiento de actualización de nodos administrados.
- No incurre en costos adicionales por usar grupos de nodos administrados de Amazon EKS. Solo pagará por los recursos de AWS que aprovisione.
- Si desea cifrar volúmenes de Amazon EBS para sus nodos, puede implementarlos mediante una plantilla de lanzamiento. Para implementar nodos administrados con volúmenes cifrados de Amazon EBS sin utilizar una plantilla de lanzamiento, cifre todos los nuevos volúmenes de Amazon EBS creados en su cuenta. Para obtener más información, consulte Cifrado de forma predeterminada en la Guía del usuario de Amazon EC2.

Tipos de capacidad de grupo de nodos administrado

Al crear un grupo de nodos administrado, puede elegir el tipo de capacidad bajo demanda o Spot. Amazon EKS implementa un grupo de nodos administrado con un grupo de Amazon EC2 Auto Scaling que solo contiene instancias de spot bajo demanda o solo instancias de spot de Amazon EC2. Puede programar Pods para aplicaciones con tolerancia a errores en grupos de nodos administrados por spot y aplicaciones intolerantes a errores a grupos de nodos bajo demanda dentro de un único clúster de Kubernetes. De forma predeterminada, un grupo de nodos administrado implementa instancias de Amazon EC2 bajo demanda.

Bajo demanda

Con instancias bajo demanda, paga la capacidad informática por segundo, sin compromisos a largo plazo.

Funcionamiento

De forma predeterminada, si no especifica un Tipo de capacidad, el grupo de nodos administrado se aprovisionará con instancias bajo demanda. En su nombre, un grupo de nodos administrado configura un grupo de Amazon EC2 Auto Scaling con la siguiente configuración aplicada:

- La estrategia de asignación para aprovisionar capacidad bajo demanda se establece en `prioritized`. El grupo de nodos administrado utiliza el orden de los tipos de instancia de la API para determinar qué tipo de instancia se utilizará en primer lugar al cumplir con la capacidad bajo demanda. Por ejemplo, puede especificar tres tipos de instancias en el siguiente orden: `c5.large`, `c4.large` y `c3.large`. Cuando se lanzan las instancias bajo demanda, el grupo de nodos administrado satisface la capacidad bajo demanda, primero en `c5.large`, luego en `c4.large` y luego en `c3.large`. Para obtener más información, consulte Grupo de Amazon EC2 Auto Scaling en la Guía del usuario de Amazon EC2 Auto Scaling.
- Amazon EKS agrega la siguiente etiqueta de Kubernetes a todos los nodos del grupo de nodos administrado que especifica el tipo de capacidad: `eks.amazonaws.com/capacityType: ON_DEMAND`. Puede utilizar esta etiqueta para programar aplicaciones con estado o intolerantes a errores en nodos bajo demanda.

Spot

Las instancias de spot de Amazon EC2 son una capacidad de Amazon EC2 de reemplazo que ofrece grandes descuentos con respecto a los precios bajo demanda. Las instancias de spot de

Amazon EC2 se pueden interrumpir con una notificación previa de dos minutos cuando EC2 necesita recuperar la capacidad. Para obtener más información, consulte Instancias de spot en Guía del usuario de Amazon EC2. Puede configurar un grupo de nodos administrado con instancias de spot de Amazon EC2 para optimizar los costos de los nodos informáticos que se ejecutan en su clúster de Amazon EKS.

Funcionamiento

Para utilizar instancias de spot dentro de un grupo de nodos administrados, cree uno y establezca el tipo de capacidad como `spot`. Un grupo de nodos administrado configura un grupo de Amazon EC2 Auto Scaling en su nombre con las siguientes prácticas recomendadas de spot aplicadas:

- Para garantizar que los nodos de spot se aprovisionen en los grupos de capacidad de spot óptima, la estrategia de asignación se establece en una de las siguientes:
  - `price-capacity-optimized` (PCO): Al crear nuevos grupos de nodos en un clúster con la versión `1.28` de Kubernetes o superior, la estrategia de asignación se establece en `price-capacity-optimized`. Sin embargo, la estrategia de asignación no cambiará para los grupos de nodos ya creados con `capacity-optimized` antes de que los grupos de nodos gestionados por Amazon EKS comenzaran a admitir PCO.
  - `capacity-optimized` (CO): Al crear nuevos grupos de nodos en un clúster con la versión `1.27` de Kubernetes o una versión anterior, la estrategia de asignación se establece en `capacity-optimized`.

  Para aumentar el número de grupos de capacidad de spot disponibles a fin de asignar su capacidad, configure un grupo de nodos administrados para utilizar varios tipos de instancias.
- El reequilibrio de capacidad de spot de Amazon EC2 está habilitado para que Amazon EKS pueda drenar y reequilibrar los nodos de spot de forma sencilla para minimizar la interrupción de la aplicación cuando un nodo de spot corre un riesgo elevado de interrupción. Para obtener más información, consulte Reequilibrio de capacidad de Amazon EC2 Auto Scaling en la Guía del usuario de Amazon EC2 Auto Scaling.
  - Cuando un nodo de Spot recibe una recomendación de reequilibrio, Amazon EKS automáticamente intenta lanzar un nuevo nodo de Spot de reemplazo.
  - Si llega un aviso de interrupción de spot de dos minutos antes de que el nodo de spot de reemplazo se encuentre en estado `Ready`, Amazon EKS comienza a vaciar el nodo de spot que recibió la recomendación de reequilibrio. Amazon EKS drena el nodo haciendo todo lo posible. En consecuencia, no hay garantía de que Amazon EKS espere a que el nodo de reemplazo se una al clúster antes de agotar el nodo existente.

  - Cuando se inicia un nodo spot de reemplazo con el estado `Ready` en Kubernetes, Amazon EKS acordona y drena el nodo de spot que recibió la recomendación de reequilibrio. El acordonamiento del nodo de spot garantiza que el controlador de servicio no envíe nuevas solicitudes a este nodo de spot. También lo elimina de su lista de nodos de spot activos y en buen estado. Drenar el nodo de spot garantiza que los Pods en funcionamiento se expulsen de manera sencilla.
- Amazon EKS agrega la siguiente etiqueta de Kubernetes a todos los nodos del grupo de nodos administrado que especifica el tipo de capacidad: `eks.amazonaws.com/capacityType: SPOT`. Puede utilizar esta etiqueta para programar aplicaciones con tolerancia a errores en nodos de spot.

Consideraciones para seleccionar un tipo de capacidad

Al decidir si desea implementar un grupo de nodos con capacidad bajo demanda o de spot, debe tener en cuenta las siguientes condiciones:

- Las instancias de spot son una buena opción para aplicaciones sin estado, tolerantes a errores y flexibles. Estos incluyen cargas de trabajo de formación de machine learning y por lotes, ETL de big data como Apache Spark, aplicaciones de procesamiento de colas y puntos de conexión de API sin estado. Dado que spot es capacidad de Amazon EC2 de reemplazo y puede cambiar con el tiempo, le recomendamos que utilice la capacidad de spot para cargas de trabajo tolerantes a interrupciones. Más concretamente, la capacidad de spot es adecuada para cargas de trabajo que pueden tolerar periodos en los que la capacidad requerida no está disponible.
- Recomendamos que utilice la opción bajo demanda para aplicaciones intolerantes a errores. Esto incluye herramientas de administración de clústeres como herramientas operativas y de supervisión, implementaciones que requieren `StatefulSets` y aplicaciones con estado, como bases de datos.
- Para maximizar la disponibilidad de las aplicaciones mientras se utilizan instancias de spot, se recomienda configurar un grupo de nodos administrado de spot para que utilice varios tipos de instancias. Se recomienda aplicar las siguientes reglas al utilizar varios tipos de instancias:
  - Dentro de un grupo de nodos administrado, si está utilizando el escalador automático de clústeres, se recomienda utilizar un conjunto flexible de tipos de instancias con la misma cantidad de vCPU y recursos de memoria. Esto es para garantizar que los nodos del clúster se escalen según lo esperado. Por ejemplo, si necesita cuatro vCPU y 8 GiB de memoria, utilice `c3.xlarge`, `c4.xlarge`, `c5.xlarge`, `c5d.xlarge`, `c5a.xlarge`, `c5n.xlarge` u otros tipos de instancias similares.

- Para mejorar la disponibilidad de las aplicaciones, recomendamos implementar varios grupos de nodos administrados por spot. Para ello, cada grupo debe utilizar un conjunto flexible de tipos de instancias que tengan los mismos recursos de memoria y vCPU. Por ejemplo, si necesita 4 vCPU y 8 GiB de memoria, le recomendamos que cree un grupo de nodos administrado con `c3.xlarge`, `c4.xlarge`, `c5.xlarge`, `c5d.xlarge`, `c5a.xlarge`, `c5n.xlarge` u otros tipos de instancias similares, y un segundo grupo de nodos administrado con `m3.xlarge`, `m4.xlarge`, `m5.xlarge`, `m5d.xlarge`, `m5a.xlarge`, `m5n.xlarge` u otros tipos de instancias similares.
- Al implementar el grupo de nodos con el tipo de capacidad spot que utiliza una plantilla de lanzamiento personalizada, utilice la API para pasar varios tipos de instancia. No pase ni un solo tipo de instancia a través de la plantilla de lanzamiento. Para obtener más información sobre cómo implementar un grupo de nodos con una plantilla de lanzamiento, consulte Personalización de nodos administrados con plantillas de lanzamiento.

Creación de un grupo de nodos administrados

En este tema, se describe cómo puede lanzar grupos de nodos administrados de Amazon EKS que se registra en el clúster de Amazon EKS. Una vez que los nodos se hayan unido al clúster, puede implementar aplicaciones de Kubernetes en ellos.

Si es la primera vez que lanza un grupo de nodos administrado de Amazon EKS, le recomendamos que siga una de nuestras guías Introducción a Amazon EKS en su lugar. Las guías proporcionan explicaciones para crear un clúster de Amazon EKS con nodos.

> **Important**
>
> - Los nodos de Amazon EKS son instancias estándar de Amazon EC2. Se le facturará en función de los precios normales de Amazon EC2. Para obtener más información, consulte Precios de Amazon EC2.
> - No puede crear nodos administrados en una Región de AWS en la que tiene AWS Outposts, AWS Wavelength o bien AWS Local Zones habilitados. Puede crear nodos autoadministrados en una Región de AWS en la que tenga habilitados AWS Outposts, AWS Wavelength, o bien AWS Local Zones. Para obtener más información, consulte Lanzar nodos autoadministrados de Amazon Linux,Lanzamiento de nodos de Windows autoadministrados y Lanzamiento de nodos de Bottlerocket autoadministrados. También puede crear un grupo de nodos autoadministrados de Amazon Linux en un Outpost. Para

obtener más información, consulte Lanzamiento de nodos autoadministrados de Amazon Linux en un Outpost.

- Si no especifica un ID de AMI para el archivo `bootstrap.sh` incluido en Amazon EKS optimizado para Linux o Bottlerocket, los grupos de nodos administrados imponen un número máximo al valor de `maxPods`. Para las instancias con menos de 30 vCPU, el número máximo es `110`. Para las instancias con más de 30 vCPU, el número máximo aumenta a `250`. Estas cifras se basan en los umbrales de escalabilidad de Kubernetes y en las configuraciones recomendadas mediante las pruebas internas del equipo de escalabilidad de Amazon EKS. Para obtener más información, consulte la entrada del blog Complemento CNI de Amazon VPC que aumenta los límites de pods por nodo.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Creación de un clúster de Amazon EKS.
- Un rol de IAM existente para que lo utilicen los nodos. Para crear uno, consulte Rol de IAM de nodo de Amazon EKS. Si este rol no tiene ninguna de las políticas de la CNI de la VPC, es necesario el rol independiente que se indica a continuación para los pods de la CNI de la VPC.
- (Opcional pero recomendado) El complemento Amazon VPC CNI plugin for Kubernetes configurado con su propio rol de IAM que tenga adjunta la política de IAM necesaria. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
- Conocimiento de las consideraciones enumeradas en Elección de un tipo de instancia de Amazon EC2. Según el tipo de instancia que elija, es posible que haya requisitos previos adicionales para su clúster y VPC.
- Para agregar un grupo de nodos administrado para Windows, primero debe habilitar la compatibilidad con Windows de su clúster. Para obtener más información, consulte Activación de la compatibilidad con Windows para su clúster de Amazon EKS.

Puede crear un grupo de nodos administrados con `eksctl` o la AWS Management Console.

eksctl

Crear un grupo de nodos administrados con **`eksctl`**

En este procedimiento, se requiere la versión `0.183.0` o posterior de la `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

1. (Opcional) Si la política administrada de IAM AmazonEKS_CNI_Policy se adjunta a su Rol de IAM de nodo de Amazon EKS, recomendamos asignarla a un rol de IAM asociado a la cuenta de servicios del `aws-node` de Kubernetes en su lugar. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
2. Cree un grupo de nodos administrados con una plantilla de lanzamiento personalizada o sin ella. La especificación manual de una plantilla de lanzamiento permite una mayor personalización de un grupo de nodos. Por ejemplo, puede permitir implementar una AMI personalizada o proporcionar argumentos al script `boostrap.sh` en una AMI optimizada para Amazon EKS. Para obtener una lista completa de todas las opciones y valores predeterminados disponibles, ingrese el siguiente comando.

   ```
   eksctl create nodegroup --help
   ```

   En el siguiente comando, reemplace *`my-cluster`* por el nombre del clúster y sustituya *`my-mng`* por el nombre del grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.

   > **Important**
   >
   > Si no utiliza una plantilla de lanzamiento personalizada al crear un grupo de nodos administrados, no utilice una en otro momento para el grupo de nodos. Si no especificó una plantilla de lanzamiento personalizada, el sistema genera automáticamente una plantilla de lanzamiento que no recomendamos que modifique

manualmente. Si se modifica manualmente esta plantilla de lanzamiento generada automáticamente, se podrían producir errores.

Sin una plantilla de lanzamiento

`eksctl` crea una plantilla de lanzamiento predeterminada de Amazon EC2 en su cuenta e implementa el grupo de nodos mediante una plantilla de lanzamiento que crea en función de las opciones que especifique. Antes de especificar un valor para `--node-type`, consulte Elección de un tipo de instancia de Amazon EC2.

Sustituya *`ami-family`* por una palabra clave permitida. Para obtener más información, consulte Configuración de la familia AMI del nodo en la documentación de `eksctl`. Reemplace *`my-key`* con el nombre de su par de claves de Amazon EC2 o la clave pública. Esta clave se utiliza para SSH en sus nodos después de que se lancen.

> **Note**
>
> Para Windows, este comando no habilita SSH. En su lugar, asocia el par de claves de Amazon EC2 con la instancia y le permite RDP en la instancia.

Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener información de Linux, consulte Pares de claves de Amazon EC2 e instancias de Linux en la Guía del usuario de Amazon EC2. Para obtener información de Windows, consulte Pares de claves de Amazon EC2 e instancias de Windows en la Guía del usuario de Amazon EC2.

Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

- Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
- Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Para bloquear el acceso del Pod a IMDS, agregue la opción **--disable-pod-imds** al siguiente comando.

```
eksctl create nodegroup \
  --cluster my-cluster \
  --region region-code \
  --name my-mng \
  --node-ami-family ami-family \
  --node-type m5.large \
  --nodes 3 \
  --nodes-min 2 \
  --nodes-max 4 \
  --ssh-access \
  --ssh-public-key my-key
```

Las instancias pueden asignar de manera opcional un número significativamente mayor de direcciones IP a los Pods, asignar direcciones IP a los Pods de un bloque de CIDR diferente al de la instancia e implementar en un clúster sin acceso a Internet. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2, Redes personalizadas para los pods y Requisitos del clúster privado para obtener opciones adicionales que se pueden agregar al comando anterior.

Los grupos de nodos administrados calculan y aplican un único valor para el número máximo de Pods que se pueden ejecutar en cada nodo del grupo de nodos, según el tipo de instancia. Si crea un grupo de nodos con distintos tipos de instancias, el valor más pequeño calculado en todos los tipos de instancias se aplica como el número máximo de Pods que se pueden ejecutar en cada tipo de instancia del grupo de nodos. Los grupos de nodos administrados calculan el valor mediante el script al que se hace referencia en Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2.

Con una plantilla de lanzamiento

La plantilla de lanzamiento ya debe existir y cumplir con los requisitos especificados en Conceptos básicos de configuración de plantillas de lanzamiento.

Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

- Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
- Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Si desea bloquear el acceso de Pod a IMDS, especifique la configuración necesaria en la plantilla de lanzamiento.

a. Copie los siguientes contenidos en su dispositivo. Reemplace los *example values* y, a continuación, ejecute el comando modificado para crear el archivo `eks-nodegroup.yaml`. Varias configuraciones que especifique al implementar sin una plantilla de lanzamiento se mueven a la plantilla de lanzamiento. Si no especifica una `version`, se usa la versión de plantilla predeterminada.

```
cat >eks-nodegroup.yaml <<EOF
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: my-cluster
  region: region-code
managedNodeGroups:
- name: my-mng
  launchTemplate:
    id: lt-id
    version: "1"
EOF
```

Para obtener una lista completa de configuraciones de archivo de config de `eksctl`, consulte Esquema de archivo de Config en la documentación de `eksctl`. Las instancias pueden asignar de manera opcional un número significativamente mayor de direcciones IP a los Pods, asignar direcciones IP a los Pods de un bloque de CIDR diferente al de la instancia, utilizar el tiempo de ejecución de `containerd` e implementar en un clúster sin acceso a Internet saliente. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2, Redes personalizadas para los pods, Prueba de la migración

de Docker a `containerd` y Requisitos del clúster privado para obtener opciones adicionales que se pueden agregar al archivo de config.

Si no especificó ningún ID de AMI en la plantilla de lanzamiento, los grupos de nodos administrados calculan y aplican un único valor para el número máximo de Pods que se pueden ejecutar en cada nodo del grupo de nodos, según el tipo de instancia. Si crea un grupo de nodos con distintos tipos de instancias, el valor más pequeño calculado en todos los tipos de instancias se aplica como el número máximo de Pods que se pueden ejecutar en cada tipo de instancia del grupo de nodos. Los grupos de nodos administrados calculan el valor mediante el script al que se hace referencia en Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2.

Si especificó un ID de AMI en la plantilla de lanzamiento, debe especificar el número máximo de Pods que se pueden ejecutar en cada nodo del grupo de nodos si usa redes personalizadas o quiere aumentar el número de direcciones IP asignadas a la instancia. Para obtener más información, consulte Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2.

b. Implemente el grupo de nodos con el siguiente comando.

```
eksctl create nodegroup --config-file eks-nodegroup.yaml
```

AWS Management Console

Para crear un grupo de nodos administrados con la AWS Management Console

1. Espere a que el estado del clúster sea `ACTIVE`. No se puede crear un grupo de nodos administrados para un clúster que aún no está `ACTIVE`.
2. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
3. Elija el nombre del clúster en el que desea crear un grupo de nodos administrados.
4. En la pestaña Informática.
5. Elija Agregar grupo de nodos.
6. En la página Configurar grupo de nodos rellene los parámetros en consecuencia y, a continuación, elija Siguiente.

- Nombre: Ingrese un nombre único para el grupo de nodos administrado. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.
- Rol de IAM de nodos: Elija el rol de instancia de nodo que se va a utilizar con su grupo de nodos. Para obtener más información, consulte Rol de IAM de nodo de Amazon EKS.

  > **Important**
  > - No se puede usar el mismo rol que se usa para crear clústeres.
  > - Es recomendable utilizar un rol que no se esté utilizando actualmente en ningún grupo de nodos autoadministrados. De lo contrario, se utiliza en un nuevo grupo de nodos autoadministrados. Para obtener más información, consulte Eliminación de un grupo de nodos administrados.

- Utilizar la plantilla de lanzamiento: (opcional) seleccione esta opción si desea utilizar una plantilla de lanzamiento existente. Seleccione un Nombre de plantilla de lanzamiento. A continuación, seleccione Versión de plantilla de lanzamiento. Si no selecciona una versión, Amazon EKS utiliza la versión predeterminada de la plantilla. Las plantillas de lanzamiento permiten una mayor personalización del grupo de nodos, como permitir implementar una AMI personalizada, asignar un número significativamente mayor de direcciones IP a los Pods, asignar direcciones IP a los Pods de un bloque de CIDR diferente al de la instancia, habilitar el tiempo de ejecución de `containerd` para sus instancias e implementar nodos en un clúster sin acceso a Internet saliente. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2, Redes personalizadas para los pods, Prueba de la migración de Docker a `containerd` y Requisitos del clúster privado.

  La plantilla de lanzamiento debe cumplir los requisitos de Personalización de nodos administrados con plantillas de lanzamiento. Si no utiliza su propia plantilla de lanzamiento, la API de Amazon EKS crea una plantilla de lanzamiento predeterminada de Amazon EC2 en su cuenta e implementa el grupo de nodos utilizando la plantilla de lanzamiento predeterminada.

  Si implementa roles de IAM para cuentas de servicio, asigna los permisos necesarios directamente a todos los Pod que requieren acceso a servicios de AWS y ningún Pods del clúster requiere acceso a IMDS por otros motivos (como recuperar la Región de AWS actual), también puede desactivar el acceso a IMDS para los Pods que no utilizan redes de

host en una plantilla de lanzamiento. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

- Etiquetas de Kubernetes: (opcional) puede optar por aplicar etiquetas de Kubernetes a los nodos del grupo de nodos administrado.
- Taints de Kubernetes: (opcional) puede optar por aplicar taints de Kubernetes de los nodos de su grupo de nodos administrados. Las opciones disponibles en el menú Efecto son **`NoSchedule`**, **`NoExecute`** y **`PreferNoSchedule`**. Para obtener más información, consulte Taints de nodos para grupos de nodos administrados.
- Etiquetas: (Opcional) puede elegir etiquetar su grupo de nodos administrado de Amazon EKS. Estas etiquetas no se propagan a otros recursos del grupo de nodos, como instancias o grupos de escalado automático. Para obtener más información, consulte Etiquetado de los recursos de Amazon EKS.

7. En la página Establecer configuración de informática y escalado, rellene los parámetros según corresponda y, a continuación, elija Siguiente.

   - Tipo de AMI: Seleccione un tipo de AMI. Si implementa instancias Arm, asegúrese de revisar las consideraciones de AMI de Amazon Linux optimizada para Amazon EKS Arm antes de implementarlas.

     Si especificó una plantilla de lanzamiento en la página anterior y especificó una AMI en la plantilla de lanzamiento, no podrá seleccionar un valor. Se muestra el valor de la plantilla. La AMI especificada en la plantilla debe cumplir los requisitos de Especificación de una AMI.
   - Tipo de capacidad: Seleccione un tipo de capacidad. Para obtener más información sobre cómo elegir un tipo de capacidad, consulte Tipos de capacidad de grupo de nodos administrado. No se pueden mezclar diferentes tipos de capacidad dentro del mismo grupo de nodos. Si desea utilizar ambos tipos de capacidad, cree grupos de nodos independientes, cada uno con su propia capacidad y tipos de instancias.
   - Tipos de instancia: Se especifican uno o más tipos de instancia de forma predeterminada. Para eliminar un tipo de instancia predeterminado, seleccione la casilla X en la parte derecha del tipo de instancia. Elija los tipos de instancia que se van a utilizar en el grupo de nodos administrado. Para obtener más información, consulte Elección de un tipo de instancia de Amazon EC2.

     La consola muestra un conjunto de tipos de instancia de uso frecuente. Si necesita crear un grupo de nodos administrados con un tipo de instancia que no figura en la lista, utilice

`eksctl`, la AWS CLI, AWS CloudFormation o un SDK para crear el grupo de nodos. Si especificó una plantilla de lanzamiento en la página anterior, no podrá seleccionar un valor porque el tipo de instancia debe especificarse en la plantilla de lanzamiento. Se muestra el valor de la plantilla de lanzamiento. Si seleccionó Spot para Tipo de capacidad, recomendamos especificar varios tipos de instancia para mejorar la disponibilidad.

- Tamaño del disco: Ingrese el tamaño del disco (en GiB) que se va a utilizar para el volumen raíz de su nodo.

  Si especificó una plantilla de lanzamiento en la página anterior, no podrá seleccionar un valor porque debe especificarse en la plantilla de lanzamiento.

- Tamaño deseado: Especifique el número actual de nodos que debe mantener el grupo de nodos administrado durante el lanzamiento.

  > **Note**
  >
  > Amazon EKS no escala automáticamente el grupo de nodos de entrada o salida. Sin embargo, puede configurar el escalador automático de clústeres de Kubernetes para que lo haga por usted.

- Tamaño mínimo: Especifica la cantidad mínima de nodos a los que puede escalar el grupo de nodos administrado.
- Tamaño máximo: Especifica el número máximo de nodos a los que puede escalar el grupo de nodos administrado.
- Configuración de la actualización del grupo de nodos: (Opcional) puede seleccionar el número o el porcentaje de nodos que se actualizarán en paralelo. Estos nodos no estarán disponibles durante la actualización. En Máximo no disponible, seleccione una de las siguientes opciones y especifique un valor:
  - Número: Seleccione y especifique el número de nodos del grupo de nodos que se pueden actualizar en paralelo.
  - Porcentaje: Seleccione y especifique el porcentaje de nodos del grupo de nodos que se pueden actualizar en paralelo. Esto es útil si tiene un gran número de nodos en su grupo de nodos.

8. En la página Especificar redes, rellene los parámetros como corresponda y, a continuación, elija Siguiente.

   - Subredes: Elija las subredes en las que lanzar los nodos administrados.

> **Important**
>
> Si está ejecutando una aplicación con estado en varias zonas de disponibilidad respaldadas por volúmenes de Amazon EBS y utilizando el Escalado automático de Kubernetes, debe configurar varios grupos de nodos, cada uno enfocado a una sola zona de disponibilidad. Además, debe habilitar la característica `--balance-similar-node-groups`.

> **Important**
>
> - Si elige una subred pública y el clúster solo tiene habilitado el punto de conexión del servidor API público, la subred debe tener `MapPublicIPOnLaunch` establecido en `true` para que las instancias se unan correctamente a un clúster. Si la subred se creó con `eksctl` o las plantillas de AWS CloudFormation ofrecidas por Amazon EKS el 26 de marzo de 2020 o después, este valor ya está establecido en `true`. Si las subredes se crearon con `eksctl` o las plantillas de AWS CloudFormation antes del 26 de marzo de 2020, debe cambiar el valor de forma manual. Para obtener más información, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred.
> - Si utiliza una plantilla de lanzamiento y especifica varias interfaces de red, Amazon EC2 no asignará automáticamente una dirección `IPv4` pública, incluso si `MapPublicIpOnLaunch` se establece en `true`. Para que los nodos se unan al clúster en este escenario, debe habilitar el punto de conexión del servidor API privado del clúster o lanzar nodos en una subred privada con acceso a Internet saliente proporcionado a través de un método alternativo, como una puerta de enlace NAT. A fin de obtener más información, consulte Direcciones IP de instancias de Amazon EC2 en la Guía del usuario de Amazon EC2.

- Configure el acceso SSH a los nodos (opcional). El acceso SSH le permite conectarse a sus instancias y recopilar información de diagnóstico si hay algún problema. Le recomendamos que habilite el acceso remoto cuando cree un grupo de nodos. No puede habilitar el acceso remoto después de crear el grupo de nodos.

  Si eligió usar una plantilla de lanzamiento, esta opción no se muestra. Para habilitar el acceso remoto a los nodos, especifique un par de claves en la plantilla de lanzamiento

y asegúrese de que el puerto adecuado esté abierto para los nodos de los grupos de seguridad que especifique en la plantilla de lanzamiento. Para obtener más información, consulte Uso de grupos de seguridad personalizados.

> **Note**
>
> Para Windows, este comando no habilita SSH. En su lugar, asocia el par de claves de Amazon EC2 con la instancia y le permite RDP en la instancia.

- En Par de claves de SSH seleccione una clave SSH de Amazon EC2 para utilizar. Para obtener información de Linux, consulte Pares de claves de Amazon EC2 e instancias de Linux en la Guía del usuario de Amazon EC2. Para obtener información de Windows, consulte Pares de claves de Amazon EC2 e instancias de Windows en la Guía del usuario de Amazon EC2. Si eligió usar una plantilla de lanzamiento, no puede seleccionar una. Cuando se proporciona una clave SSH de Amazon EC2 para grupos de nodos que utilizan AMI de Bottlerocket, el contenedor administrativo también se habilita. Para obtener más información, consulte Contenedor de administración en GitHub.
- En Permitir acceso remoto de SSH desde, si desea limitar el acceso a instancias específicas, seleccione los grupos de seguridad asociados a dichas instancias. Si no selecciona grupos de seguridad específicos, se permite el acceso SSH desde cualquier lugar de Internet (`0.0.0.0/0`).

9. En la página Revisar y crear, revise la configuración del grupo de nodos administrados y elija Crear.

   Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en la Guía de solución de problemas.

10. Observe el estado de los nodos y espere a que aparezca el estado `Ready`.

    ```
    kubectl get nodes --watch
    ```

11. (Solo para nodos de GPU) Si ha elegido un tipo de instancia de GPU y la AMI acelerada optimizada para Amazon EKS, debe aplicar el complemento de dispositivo NVIDIA para Kubernetes como un DaemonSet en su clúster. Reemplace *vX.X.X* con la versión Plugin de dispositivo NVidia/K8Sdeseada antes de ejecutar el siguiente comando.

    ```
    kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
    ```

Ahora que tiene un clúster de Amazon EKS en funcionamiento con nodos, está listo para comenzar a instalar los complementos de Kubernetes e implementar aplicaciones en su clúster. Los siguientes temas de documentación lo ayudarán a ampliar la funcionalidad de su clúster.

- La entidad principal de IAM que creó el clúster es la única entidad principal que puede realizar llamadas al servidor de la API de Kubernetes con `kubectl` o laAWS Management Console. Si desea que otras entidades principales de IAM tengan acceso al clúster, debe agregarlas. Para obtener más información, consulte Concesión de acceso a las API de Kubernetes y Permisos necesarios.
- Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:
  - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
  - Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

  Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.
- Escalado automático: Configure el escalador automático de clústeres de Kubernetes para ajustar de forma automática el número de nodos en sus grupos de nodos.
- Implemente una aplicación de muestra en su clúster.
- Administración de clústeres: Conozca cómo utilizar herramientas importantes para administrar el clúster.

Actualización de un grupo de nodos administrados

Cuando inicia una actualización de grupo de nodos administrados, Amazon EKS actualiza los nodos de forma automática al completar los pasos que se indican en Comportamiento de actualización de nodos administrados. Si utiliza una AMI optimizada para Amazon EKS, Amazon EKS aplica automáticamente los últimos parches de seguridad y actualizaciones del sistema operativo a los nodos como parte de la versión más reciente de la AMI.

Existen varios escenarios en los que resulta útil actualizar la versión o configuración del grupo de nodos administrado de Amazon EKS:

- Ha actualizado la versión de Kubernetes para su clúster de Amazon EKS y desea actualizar los nodos para que utilicen la misma versión de Kubernetes.

- Hay disponible una nueva versión de la AMI para el grupo de nodos administrados. Para obtener más información acerca de las versiones de AMI, consulte estas secciones:
  - Versiones de la AMI de Amazon Linux optimizada para Amazon EKS
  - AMI de Bottlerocket optimizadas para Amazon EKS
  - Versiones de AMI optimizadas para Amazon EKS de Windows
- Desea ajustar el número mínimo, máximo o deseado de las instancias del grupo de nodos administrados.
- Desea agregar o quitar etiquetas Kubernetes de las instancias del grupo de nodos administrados.
- Desea agregar o quitar etiquetas de AWS del grupo de nodos administrados.
- Debe implementar una nueva versión de una plantilla de lanzamiento con cambios de configuración, como una AMI personalizada actualizada.
- Ha implementado la versión `1.9.0` o posterior del complemento CNI de Amazon VPC, ha habilitado el complemento para la delegación de prefijos y desea nuevas instancias AWS de Nitro System en un grupo de nodos para admitir un número significativamente mayor de Pods. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.
- Ha habilitado la delegación de prefijos IP para los nodos de Windows y quiere que las nuevas instancias de Nitro System de AWS en un grupo de nodos admitan un número significativamente mayor de Pods. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.

Si hay una versión de lanzamiento de AMI más reciente para la versión de Kubernetes del grupo de nodos administrado, puede actualizar la versión de su grupo de nodos para utilizar esa nueva versión de la AMI. De manera similar, si su clúster está ejecutando una versión de Kubernetes más reciente que su grupo de nodos, puede actualizar el grupo de nodos para que utilice la última versión de la AMI que coincida con la versión de Kubernetes del clúster.

Cuando se termina un nodo de un grupo de nodos administrados debido a una operación de escalado o actualización, los Pods de ese nodo se drenan primero. Para obtener más información, consulte Comportamiento de actualización de nodos administrados.

Actualizar una versión de grupo de nodos

Puede actualizar una versión del grupo de nodos con `eksctl` o la AWS Management Console. La versión a la que se actualiza no puede ser superior a la versión del plano de control.

eksctl

Para actualizar una versión de grupo de nodos con **`eksctl`**

- Actualice un grupo de nodos administrado a la última versión de AMI de la misma versión de Kubernetes implementada actualmente en los nodos de trabajo con el siguiente comando. Sustituya cada *`example value`* con valores propios.

```
eksctl upgrade nodegroup \
  --name=node-group-name \
  --cluster=my-cluster \
  --region=region-code
```

> **Note**
>
> Si va a actualizar un grupo de nodos que se implementa con una plantilla de lanzamiento a una nueva versión de plantilla de lanzamiento, agregue `--launch-template-version` *`version-number`* en el comando anterior. La plantilla de lanzamiento debe cumplir los requisitos descritos en Personalización de nodos administrados con plantillas de lanzamiento. Si la plantilla de lanzamiento incluye una AMI personalizada, la AMI debe cumplir los requisitos de Especificación de una AMI. Cuando actualiza el grupo de nodos a una versión más reciente de la plantilla de lanzamiento, todos los nodos se reciclan para que coincidan con la nueva configuración de la versión de la plantilla de lanzamiento especificada.
> No puede actualizar directamente un grupo de nodos que se implementa sin una plantilla de lanzamiento a una nueva versión de la plantilla de lanzamiento. En su lugar, debe implementar un nuevo grupo de nodos mediante la plantilla de lanzamiento para actualizar el grupo de nodos a una nueva versión de la plantilla de lanzamiento.

Puede actualizar un grupo de nodos a la misma versión que la versión de Kubernetes del plano de control. Por ejemplo, si tiene un clúster que ejecuta Kubernetes `1.29`, puede actualizar los procesos que ejecutan Kubernetes `1.28` actualmente a la versión `1.29` con el siguiente comando.

```
eksctl upgrade nodegroup \
  --name=node-group-name \
```

```
--cluster=my-cluster \
--region=region-code \
--kubernetes-version=1.29
```

AWS Management Console

Para actualizar una versión de grupo de nodos con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el clúster que contiene el grupo de nodos que desea actualizar.
3. Si al menos un grupo de nodos tiene una actualización disponible, aparece un cuadro en la parte superior de la página con una notificación sobre la actualización disponible. Si selecciona la pestaña Compute (Computación), verá Update now (Actualizar ahora) en la columna AMI release version (Versión de lanzamiento de la AMI) de la tabla Node groups (Grupos de nodos) para el grupo de nodos que tenga una actualización disponible. Para actualizar el grupo de nodos, elija Update now (Actualizar ahora).

   No verá una notificación para los grupos de nodos que se implementaron con una AMI personalizada. Si los nodos se implementan con una AMI personalizada, complete los siguientes pasos para implementar una nueva AMI personalizada actualizada.

   a. Cree una nueva versión de su AMI.

   b. Cree una nueva versión de la plantilla de lanzamiento con el nuevo ID de AMI.

   c. Actualice los nodos a la nueva versión de la plantilla de lanzamiento.
4. En el cuadro de diálogo Update node group version (Actualizar la versión del grupo de nodos), active o desactive las siguientes opciones:

   - Update node group version (Actualizar la versión del grupo de nodos): esta opción no está disponible si ha implementado una AMI personalizada o su AMI optimizada para Amazon EKS está actualmente en la versión más reciente del clúster.
   - Change launch template version (Cambiar la versión de la plantilla de lanzamiento): esta opción no está disponible si el grupo de nodos se implementa sin una plantilla de lanzamiento personalizada. Solo puede actualizar la versión de la plantilla de lanzamiento para un grupo de nodos que se haya implementado con una plantilla de lanzamiento personalizada. Seleccione la versión de la plantilla de lanzamiento a la que desea actualizar el grupo de nodos. Si el grupo de nodos está configurado con una AMI personalizada, la versión que seleccione también debe especificar una AMI. Al actualizar

a una versión más reciente de la plantilla de lanzamiento, todos los nodos se reciclan para que coincidan con la nueva configuración de la versión de la plantilla de lanzamiento especificada.

5. En Actualizar estrategia, seleccione una de las siguientes opciones:

   - Actualización continua: esta opción respeta los presupuestos de interrupción del Pod para el clúster. Se produce un error en las actualizaciones si hay un problema de presupuesto de interrupción de Pod que hace que Amazon EKS no pueda vaciar correctamente los Pods que se están ejecutando en este grupo de nodos.
   - Actualización forzada: esta opción no respeta los presupuestos de interrupción del Pod. Las actualizaciones se producen independientemente de los problemas presupuestarios de la interrupción del Pod al forzar el reinicio de los nodos.

6. Elija Actualizar.

Editar una configuración de grupo de nodos

Puede modificar algunas de las opciones de configuración de un grupo de nodos administrado.

Para editar una configuración de grupo de nodos

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el clúster que contiene el grupo de nodos que desea editar.
3. Seleccione la pestaña Compute (Informática).
4. Seleccione el grupo de nodos que desea editar y elija Edit (Editar).
5. (Opcional) En la página Editar grupo de nodos, haga lo siguiente:

   a. Edite la configuración de escalado del grupo de nodos.

      - Tamaño deseado: especifica el número actual de nodos que debe mantener el grupo de nodos administrado.
      - Tamaño mínimo: Especifica la cantidad mínima de nodos a los que puede escalar el grupo de nodos administrado.
      - Tamaño máximo: especifica el número máximo de nodos a los que puede escalar el grupo de nodos administrado. Para obtener el número máximo de nodos admitidos en un grupo de nodos, consulte Cuotas de servicio de Amazon EKS.

b. (Opcional) Agregue o elimine etiquetas de Kubernetes para los nodos de su grupo de nodos. Las etiquetas que se muestran aquí son solo las que se han aplicado con Amazon EKS. Pueden existir otras etiquetas en los nodos que no se muestran aquí.

c. (Opcional) Agregue o elimine taints de Kubernetes para los nodos de su grupo de nodos. Las taints agregadas pueden tener el efecto de **`NoSchedule`**, **`NoExecute`** o **`PreferNoSchedule`**. Para obtener más información, consulte Taints de nodos para grupos de nodos administrados.

d. (Opcional) Agregue o elimine etiquetas del recurso de su grupo de nodos. Estas etiquetas solo se aplican al grupo de nodos de Amazon EKS. No se propagan a ningún otro recurso, como las subredes o instancias de Amazon EC2 en el grupo de nodos.

e. (Opcional) Edite la Configuración de la actualización del grupo de nodos. Seleccione el Number (Número) o el Percentage (Porcentaje).

- Número: seleccione y especifique el número de nodos del grupo de nodos que se pueden actualizar en paralelo. Estos nodos no estarán disponibles durante la actualización.
- Porcentaje: seleccione y especifique el porcentaje de nodos del grupo de nodos que se pueden actualizar en paralelo. Estos nodos no estarán disponibles durante la actualización. Esto es útil si tiene varios nodos en su grupo de nodos.

f. Cuando haya terminado de editar, elija Save changes (Guardar cambios).

Comportamiento de actualización de nodos administrados

La estrategia de actualización del nodo de trabajo administrado de Amazon EKS tiene cuatro fases diferentes descritas en las siguientes secciones.

Fase de configuración

La fase de configuración incluye los siguientes pasos:

1. Crea una nueva versión de plantilla de lanzamiento de Amazon EC2 para el grupo de escalado automático asociado al grupo de nodos. La nueva versión de la plantilla de lanzamiento utiliza la AMI de destino o la versión de plantilla de lanzamiento personalizada para la actualización.
2. El grupo de escalado automático se actualiza para utilizar la versión más reciente de la plantilla de lanzamiento.
3. Determina la cantidad máxima de nodos que se van a actualizar en paralelo mediante la propiedad de `updateConfig` para el grupo de nodos. El máximo no disponible tiene una cuota

de 100 nodos. El valor predeterminado es un nodo. Para obtener más información, consulte la propiedad `updateConfig` en la Referencia de la API de Amazon EKS.

Fase de escalado

Al actualizar los nodos de un grupo de nodos administrados, los nodos actualizados se lanzan en la misma zona de disponibilidad que los que se están actualizando. Para garantizar esta ubicación, utilizamos el reequilibrio de la zona de disponibilidad de Amazon EC2. Para obtener más información, consulte Reequilibrio de la zona de disponibilidad en la Guía del usuario de Amazon EC2 Auto Scaling. Para cumplir este requisito, es posible que lancemos hasta dos instancias por zona de disponibilidad en su grupo de nodos administrado.

La fase de escalado incluye los siguientes pasos:

1. Aumenta el tamaño máximo del grupo de escalado automático y el tamaño deseado en el mayor:
   - Hasta el doble del número de zonas de disponibilidad en la que se implementa el grupo de Auto Scaling.
   - El máximo no disponible de actualización.

     Por ejemplo, si el grupo de nodos tiene cinco zonas de disponibilidad y `maxUnavailable` como uno solo, el proceso de actualización puede lanzar un máximo de 10 nodos. Sin embargo, cuando `maxUnavailable` es 20 (o cualquier número superior a 10), el proceso puede lanzar 20 nuevos nodos.
2. Después de escalar el grupo de escalado automático, compruebe si los nodos que utilizan la configuración más reciente están presentes en el grupo de nodos. Este paso solo se efectúa correctamente cuando cumple estos criterios:
   - Se lanza al menos un nuevo nodo en cada zona de disponibilidad en la que existe el nodo.
   - Todos los nuevos nodos deberían estar en estado `Ready`.
   - Los nuevos nodos deben tener etiquetas aplicadas de Amazon EKS.

     Estas son las etiquetas aplicadas de Amazon EKS en los nodos de trabajo de un grupo de nodos normal:

     - `eks.amazonaws.com/nodegroup-image=`*`$amiName`*
     - `eks.amazonaws.com/nodegroup=`*`$nodeGroupName`*

     Estas son las etiquetas aplicadas de Amazon EKS en los nodos de trabajo en una plantilla de lanzamiento personalizado o grupo de nodos de AMI:

- `eks.amazonaws.com/nodegroup-image=$amiName`
- `eks.amazonaws.com/nodegroup=$nodeGroupName`
- `eks.amazonaws.com/sourceLaunchTemplateId=$launchTemplateId`
- `eks.amazonaws.com/sourceLaunchTemplateVersion=$launchTemplateVersion`

3. Marca los nodos como no programables para evitar programar nuevos Pods. También etiqueta los nodos con el `node.kubernetes.io/exclude-from-external-load-balancers=true` para eliminarlos de los equilibradores de carga antes de terminar los nodes.

Las siguientes son las razones conocidas que llevan a un error de `NodeCreationFailure` en esta fase:

Capacidad insuficiente en la zona de disponibilidad

Existe la posibilidad de que la zona de disponibilidad no tenga la capacidad de los tipos de instancias solicitados. Se recomienda configurar varios tipos de instancias al crear un grupo de nodos administrados.

Límites de instancias de EC2 en su cuenta

Es posible que tenga que aumentar el número de instancias de Amazon EC2 que su cuenta puede ejecutar simultáneamente mediante Service Quotas. Para obtener más información, consulte EC2 Service Quotas en la Guía del usuario de Amazon Elastic Compute Cloud para instancias de Linux.

Datos de usuario personalizados

Los datos de usuario personalizados a veces pueden interrumpir el proceso de arranque. Este escenario puede llevar a que la `kubelet` no se inicie en el nodo o que los nodos no reciban etiquetas de Amazon EKS esperadas en ellos. Para obtener más información, consulte Especificación de una AMI.

Cualquier cambio que haga que un nodo no esté en buen estado o no esté preparado

La presión del disco del nodo, la presión de la memoria y condiciones similares pueden provocar que un nodo no funcione en estado `Ready`.

Fase de actualización

La fase de actualización incluye los siguientes pasos:

1. Selecciona aleatoriamente un nodo que necesita una actualización hasta el máximo no disponible configurado para el grupo de nodos.
2. Drena los Pods del nodo. Si los Pods no salen del nodo en 15 minutos y no hay un indicador de fuerza, la fase de actualización falla con un error `PodEvictionFailure`. Para este escenario, puede aplicar el indicador de fuerza con la solicitud `update-nodegroup-version` para eliminar los Pods.
3. Acordona el nodo después de expulsar todos los Pod y esperar 60 segundos. Esto se hace para que el controlador del servicio no envíe ninguna solicitud nueva a este nodo y elimine este nodo de su lista de nodos activos.
4. Envía una solicitud de terminación al grupo de escalado automático para el nodo acordonado.
5. Repite los pasos anteriores de actualización hasta que no haya nodos en el grupo de nodos que se implementen con la versión anterior de la plantilla de lanzamiento.

Las siguientes son las razones conocidas que llevan a un error de `PodEvictionFailure` en esta fase:

PDB agresivo

El PDB agresivo se define en el Pod o hay varios PDB que apuntan al mismo Pod.

Implementación que tolera todas las taints

Una vez expulsado cada Pod, se espera que el nodo esté vacío porque el nodo se marcó como taint en los pasos anteriores. Sin embargo, si la implementación tolera todas las taints, es más probable que el nodo no esté vacío, lo que provoca un error en la expulsión del Pod.

Fase de reducción vertical

La fase de reducción vertical disminuye el tamaño máximo del grupo de Auto Scaling y el tamaño deseado en uno para volver a los valores antes de que se inicie la actualización.

Si el flujo de trabajo de actualización determina que el escalador automático del clúster realiza el escalado vertical del grupo de nodos durante la fase de reducción vertical del flujo de trabajo, se cierra inmediatamente sin que el grupo de nodos vuelva a su tamaño original.

Taints de nodos para grupos de nodos administrados

Amazon EKS admite la configuración de taints de Kubernetes a través de grupos de nodos administrados. Las taints y toleraciones funcionan juntas para garantizar que los Pods no se

programen en nodos inapropiados. Se pueden aplicar una o más taints a un nodo. Esto indica que el nodo no debe aceptar ningún Pods que no tolere las taints. Las toleraciones se aplican a los Pods y permiten, pero no requieren, que los Pods se programen en nodos con taints coincidentes. Para obtener más información, consulte Taints y toleraciones en la documentación de Kubernetes.

Las taints de nodos de Kubernetes se pueden aplicar a grupos de nodos administrados nuevos y existentes mediante la AWS Management Console o a través de la API de Amazon EKS.

- Para obtener información sobre la creación de un grupo de nodos con una taint mediante la AWS Management Console, consulte Creación de un grupo de nodos administrados.
- A continuación, se muestra un ejemplo de creación de un grupo de nodos con una taint mediante la AWS CLI:

```
aws eks create-nodegroup \
 --cli-input-json '
{
  "clusterName": "my-cluster",
  "nodegroupName": "node-taints-example",
  "subnets": [
     "subnet-1234567890abcdef0",
     "subnet-abcdef01234567890",
     "subnet-021345abcdef67890"
   ],
  "nodeRole": "arn:aws:iam::111122223333:role/AmazonEKSNodeRole",
  "taints": [
     {
         "key": "dedicated",
         "value": "gpuGroup",
         "effect": "NO_SCHEDULE"
     }
   ]
}'
```

Para obtener más información y ejemplos de uso, consulte taint en la documentación de referencia de Kubernetes.

Note

- Las taints se pueden actualizar después de crear el grupo de nodos mediante la API `UpdateNodegroupConfig`.
- La clave de la taint debe comenzar con una letra o un número. Puede contener letras, números, guiones (**-**), puntos (**.**) y guiones bajos (**_**). Puede tener hasta 63 caracteres.
- De manera opcional, la clave de la taint puede comenzar con un prefijo de subdominio DNS y una única **/**. Si comienza con un prefijo de subdominio DNS, puede tener 253 caracteres de longitud.
- El valor es opcional y debe comenzar por una letra o un número. Puede contener letras, números, guiones (**-**), puntos (**.**) y guiones bajos (**_**). Puede tener hasta 63 caracteres.
- Cuando se usa Kubernetes directamente o la AWS Management Console, el efecto de la taint debe ser **`NoSchedule`**, **`PreferNoSchedule`** o **`NoExecute`**. Sin embargo, cuando se usa la AWS CLI o la API, el efecto de la taint debe ser **`NO_SCHEDULE`**, **`PREFER_NO_SCHEDULE`** o **`NO_EXECUTE`**.
- Se permite un máximo de 50 taints por grupo de nodos.
- Si las taint que se crearon mediante un grupo de nodos administrado se eliminan manualmente de un nodo, Amazon EKS no volverá a añadir las taint al nodo. Esto es cierto incluso si las taint se especifican en la configuración del grupo de nodos administrado.

Puede utilizar el comando `aws eks update-nodegroup-config` AWS CLIpara añadir, eliminar o reemplazar elementos taint en los grupos de nodos gestionados.

Personalización de nodos administrados con plantillas de lanzamiento

Para obtener el nivel más alto de personalización, puede implementar nodos administrados mediante el uso de su propia plantilla de lanzamiento. El uso de una plantilla de lanzamiento le permite hacer lo siguiente:

- Proporcionar argumentos de arranque en la implementación de un nodo, como argumentos `kubelet` adicionales.
- Asignar direcciones IP a Pods desde un bloque de CIDR diferente de la dirección IP asignada al nodo.
- Implementar su propia AMI personalizada en los nodos.

- Implementar su propia CNI personalizada en los nodos.

Si proporciona su propia plantilla de lanzamiento al crear por primera vez un grupo de nodos administrado, también tendrá mayor flexibilidad más adelante. Siempre que implemente un grupo de nodos administrado con su propia plantilla de lanzamiento, puede actualizarlo de forma iterativa con una versión diferente de la misma plantilla de lanzamiento. Cuando actualiza el grupo de nodos a una versión diferente de la plantilla de lanzamiento, todos los nodos del grupo se reciclan para que coincidan con la nueva configuración de la versión de la plantilla de lanzamiento especificada.

Los grupos de nodos administrados se implementan siempre con una plantilla de lanzamiento para utilizar con el grupo de Amazon EC2 Auto Scaling. Cuando no proporciona una plantilla de lanzamiento, la API de Amazon EKS crea una en su cuenta de forma automática con los valores predeterminados. Sin embargo, no le recomendamos que modifique las plantillas de lanzamiento generadas automáticamente. Además, los grupos de nodos existentes que no utilizan una plantilla de lanzamiento personalizada no se pueden actualizar directamente. En su lugar, debe crear un nuevo grupo de nodos con una plantilla de lanzamiento personalizada para hacerlo.

Conceptos básicos de configuración de plantillas de lanzamiento

Puede crear una plantilla de lanzamiento de Amazon EC2 Auto Scaling con la AWS Management Console, la AWS CLI o un SDK de AWS. Para obtener más información, consulte Creación de una plantilla de lanzamiento para un grupo de Auto Scaling en la guía del usuario de Amazon EC2 Auto Scaling. Algunas de las opciones de configuración de una plantilla de lanzamiento son similares a las que se utilizan para la configuración de nodos administrados. Al implementar o actualizar un grupo de nodos con una plantilla de lanzamiento, se deben especificar algunas opciones en la configuración del grupo de nodos o en la plantilla de lanzamiento. No especifique un ajuste en ambos lugares. Si existe una configuración donde no debería, las operaciones como la creación o actualización de un grupo de nodos fallarán.

En la siguiente tabla, se enumeran los ajustes prohibidos en una plantilla de lanzamiento. También se enumeran ajustes similares, si hay alguno disponible, que se requieren en la configuración del grupo de nodos administrados. La configuración de la lista es la configuración que aparece en la consola. Pueden tener nombres similares, pero diferentes en la AWS CLI y el SDK.

Plantilla de lanzamiento: opciones prohibidas	Configuración del grupo de nodos de Amazon EKS
Subred en Interfaces de red (Agregar interfaz de red)	Subredes en Configuración de red del grupo de nodos en la página Especificar red
Perfil de instancia de IAM en Detalles avanzados	Rol de IAM del nodo en Configuración del grupo de nodos en la página Configurar grupo de nodos.
Comportamiento de apagado y Detener: comportamiento de hibernación en Detalles avanzados. Mantenga la opción predeterm inada No incluir en la configuración de la plantilla de lanzamiento en la plantilla de lanzamiento para ambas configuraciones.	Sin equivalente. Amazon EKS debe controlar el ciclo de vida de la instancia, no el grupo de escalado automático.

En la siguiente tabla, se enumeran los ajustes prohibidos de una configuración de grupo de nodos administrados. También se enumeran configuraciones similares, si hay alguna disponible, que son necesarias en una plantilla de lanzamiento. La configuración de la lista es la configuración que aparece en la consola. Es posible que tengan nombres similares en la AWS CLI y el SDK.

Configuración del grupo de nodos de Amazon EKS: opciones prohibidas	Plantilla de inicialización
(Solo si especificó una AMI personalizada en una plantilla de lanzamiento) Tipo de AMI en Configuración de computación del grupo de nodos en la página Establecer la configuración de informática y escalado: la consola muestra Especificada en la plantilla de lanzamiento y el ID de la AMI que se especificó. Si no se especificó nada en Imágenes de aplicaciones y sistema operativo (Imagen de máquina de Amazon) en la plantilla de	Imágenes de aplicaciones y sistema operativo (Imagen de máquina de Amazon) en Contenido de la plantilla de lanzamiento: debe especificar un ID si tiene alguno de los siguientes requisito s: • Uso de una AMI personalizada. Si especific a una AMI que no cumple los requisito s enumerados en Especificación de una AMI, la implementación del grupo de nodos producirá un error.

Configuración del grupo de nodos de Amazon EKS: opciones prohibidas	Plantilla de inicialización
lanzamiento, puede seleccionar una AMI en la configuración del grupo de nodos.	• Desea facilitar datos de usuario para proporcionar argumentos al archivo `bootstrap.sh` incluido con una AMI optimizada para Amazon EKS. Puede habilitar sus instancias para asignar un número significativamente mayor de direcciones IP a los Pods, asignar direccion es IP a los Pods de un bloque de CIDR diferente al de la instancia, habilitar el tiempo de ejecución o implementar un clúster privado sin acceso a Internet saliente. Para obtener más información, consulte los temas siguientes: • Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2 • Redes personalizadas para los pods • Requisitos del clúster privado • Especificación de una AMI
Tamaño del disco en Configuración de computación del grupo de nodos en la página Establecer la configuración de informática y escalado: la consola muestra Especificado en la plantilla de lanzamiento.	Tamaño en Almacenamiento (Volúmenes) (Agregar nuevo volumen). Debe especificarlo en la plantilla de lanzamiento.
Par de claves de SSH en Configuración del grupo de nodos en la página Especificar redes: la consola muestra la clave especificada en la plantilla de lanzamiento o muestra No especific ado en la plantilla de lanzamiento.	Nombre del par de claves en Par de claves (inicio de sesión).

Configuración del grupo de nodos de Amazon EKS: opciones prohibidas	Plantilla de inicialización
No se pueden especificar grupos de seguridad fuente a los que se permita el acceso remoto cuando se utiliza una plantilla de lanzamiento.	Grupos de seguridad en Configuración de red para la instancia o Grupos de seguridad en Interfaces de red (Agregar interfaz de red), pero no ambos. Para obtener más información, consulte Uso de grupos de seguridad personalizados.

Note

- Si implementa un grupo de nodos mediante una plantilla de lanzamiento, especifique un tipo de instancia o ninguno en Contenido de la plantilla de lanzamiento en una plantilla de lanzamiento. Si lo desea, puede especificar entre 0 y 20 tipos de instancia para Tipos de instancias en la página Establecer la configuración de informática y escalado de la consola. O bien, puede hacerlo mediante otras herramientas que utilizan la API de Amazon EKS. Si especifica un tipo de instancia en una plantilla de lanzamiento y utiliza esa plantilla de lanzamiento para implementar el grupo de nodos, no podrá especificar ningún tipo de instancia en la consola ni utilizar otras herramientas que utilicen la API de Amazon EKS. Si no especifica un tipo de instancia en una plantilla de lanzamiento, en la consola o si utiliza otras herramientas que utilizan la API de Amazon EKS, se utiliza el tipo de instancia `t3.medium`. Si el grupo de nodos utiliza el tipo de capacidad spot, se recomienda especificar varios tipos de instancias mediante la consola. Para obtener más información, consulte Tipos de capacidad de grupo de nodos administrado.
- Si alguno de los contenedores que implementa en el grupo de nodos utiliza el servicio de metadatos de instancia versión 2, asegúrese de establecer la propiedad Límite del salto de respuesta de metadatos en 2 en la plantilla de lanzamiento. Para obtener más información, consulte Metadatos de instancia y datos de usuario en la Guía del usuario de Amazon EC2. Si implementa un grupo de nodos administrado sin utilizar una plantilla de lanzamiento personalizada, este valor se establece automáticamente para el grupo de nodos en la plantilla de lanzamiento predeterminada.

Etiquetado de instancias de Amazon EC2

Puede utilizar el parámetro `TagSpecification` de una plantilla de lanzamiento para especificar qué etiquetas se aplicarán a las instancias de Amazon EC2 del grupo de nodos. La entidad IAM que llama a las API `CreateNodegroup` o `UpdateNodegroupVersion` debe tener permisos para `ec2:RunInstances` y `ec2:CreateTags`, y las etiquetas deben agregarse a la plantilla de lanzamiento.

Uso de grupos de seguridad personalizados

Puede utilizar una plantilla de lanzamiento para especificar grupos de seguridad de Amazon EC2 personalizados para aplicar a instancias del grupo de nodos. Esto puede estar en el parámetro de grupos de seguridad de nivel de instancia o como parte de los parámetros de configuración de la interfaz de red. Sin embargo, no se puede crear una plantilla de lanzamiento que especifique el nivel de la instancia y los grupos de seguridad de una interfaz de red. Tenga en cuenta las siguientes condiciones que se aplican al uso de grupos de seguridad personalizados con grupos de nodos administrados:

- Amazon EKS solo permite plantillas de lanzamiento con una única especificación de interfaz de red.
- De forma predeterminada, Amazon EKS aplica el grupo de seguridad de clúster a las instancias del grupo de nodos para facilitar la comunicación entre nodos y el plano de control. Si especifica grupos de seguridad personalizados en la plantilla de lanzamiento mediante cualquiera de las opciones mencionadas anteriormente, Amazon EKS no agrega el grupo de seguridad del clúster. Debe asegurarse de que las reglas entrantes y salientes de los grupos de seguridad habiliten la comunicación con el punto de conexión del clúster. Si las reglas del grupo de seguridad son incorrectas, los nodos de trabajo no pueden unirse al clúster. Para obtener más información acerca de las reglas de los grupos de seguridad, consulte Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS.
- Si necesita acceso SSH a las instancias del grupo de nodos, incluya un grupo de seguridad que permita ese acceso.

Datos de usuario de Amazon EC2

La plantilla de lanzamiento incluye una sección para datos de usuario personalizados. Puede especificar la configuración de su grupo de nodos en esta sección sin crear manualmente AMI

personalizadas individuales. Para obtener más información sobre la configuración disponible para Bottlerocket, consulte Utilización de datos de usuario en GitHub.

Puede proporcionar datos de usuario de Amazon EC2 en su plantilla de lanzamiento mediante `cloud-init` al iniciar sus instancias. Para obtener más información, consulte la documentación de cloud-init. Los datos de usuario se pueden utilizar para realizar operaciones de configuración comunes. Esto incluye las operaciones siguientes:

- Inclusión de usuarios o grupos
- Instalación de paquetes

Los datos de usuario de Amazon EC2 en las plantillas de lanzamiento que se utilizan con grupos de nodos administrados deben estar en el formato de archivo multiparte MIME para las AMI de Amazon Linux y en el formato TOML para las AMI de Bottlerocket. Esto se debe a que los datos de usuario se combinan con los datos de usuario de Amazon EKS necesarios para que los nodos se unan al clúster. No especifique ningún comando en los datos de usuario que inicie o modifique `kubelet`. Esto se realiza como parte de los datos de usuario fusionados por Amazon EKS. Ciertos parámetros de `kubelet`, como establecer etiquetas en nodos, se pueden configurar directamente a través de la API de grupos de nodos administrados.

> **Note**
>
> Para obtener más información sobre la personalización de `kubelet` avanzada, lo que incluye un inicio manual o pasar parámetros de configuración personalizados, consulte Especificación de una AMI. Si se especifica un ID de AMI personalizado en una plantilla de lanzamiento, Amazon EKS no fusiona los datos de usuario.

Los siguientes detalles proporcionan más información sobre la sección de datos de usuario.

Amazon Linux 2 user data

Puede combinar varios bloques de datos de usuario en un único archivo multiparte MIME. Por ejemplo, puede combinar un boothook de nube que configure el daemon de Docker con un script de shell de datos de usuario que instala un paquete personalizado. Un archivo multiparte MIME consta de los siguientes componentes:

- El tipo de contenido y declaración de límite de partes: `Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="`
- La declaración de versión de MIME: `MIME-Version: 1.0`
- Uno o más bloques de datos de usuario, que contienen los siguientes componentes:
  - El límite de apertura, que señala el inicio de un bloque de datos de usuario: `--==MYBOUNDARY==`
  - La declaración de tipo de contenido para el bloque: `Content-Type: text/cloud-config; charset="us-ascii"`. Para obtener más información sobre los tipos de contenido, consulte la documentación de cloud-init.
  - El contenido de los datos de usuario, por ejemplo, una lista de comandos de shell o políticas de `cloud-init`.
  - El límite de cierre, que señala el final del archivo multiparte MIME: `--==MYBOUNDARY==--`

  A continuación, se muestra un ejemplo de un archivo multiparte MIME que puede utilizar para crear el suyo propio.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo "Running custom user data script"

--==MYBOUNDARY==--
```

Amazon Linux 2023 user data

Amazon Linux 2023 (AL2023) introduce un nuevo proceso de inicialización de nodos `nodeadm` que utiliza un esquema de configuración YAML. Si utiliza grupos de nodos autoadministrados o una AMI con una plantilla de lanzamiento, ahora tendrá que proporcionar metadatos del clúster adicionales de forma explícita cuando cree un nuevo grupo de nodos. A continuación, se muestra un ejemplo de los parámetros mínimos necesarios, en los que ahora se necesitan `apiServerEndpoint`, `certificateAuthority` y el servicio de `cidr`:

```
---
```

```
apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  cluster:
    name: my-cluster
    apiServerEndpoint: https://example.com
    certificateAuthority: Y2VydGlmaWNhdGVBdXRob3JpdHk=
    cidr: 10.100.0.0/16
```

Por lo general, establecerá esta configuración en los datos de usuario, tal y como están o incrustados en un documento MIME de varias partes:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="BOUNDARY"

--BOUNDARY
Content-Type: application/node.eks.aws

---
apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig spec: [...]

--BOUNDARY--
```

En AL2, los metadatos de estos parámetros se descubrieron a partir de la llamada a la API `DescribeCluster` de Amazon EKS. Con AL2023, este comportamiento ha cambiado, ya que la llamada a la API adicional corre el riesgo de limitarse durante los escalados verticales de nodos a gran escala. Este cambio no le afecta si utiliza grupos de nodos administrados sin una plantilla de lanzamiento o si utiliza Karpenter. Para obtener más información sobre `certificateAuthority` y el servicio de `cidr`, consulte `DescribeCluster` en la Referencia de la API de Amazon EKS.

Bottlerocket user data

Bottlerocket estructura los datos de usuario en formato TOML. Puede facilitar datos de usuario para fusionarlos con los datos de usuario proporcionados por Amazon EKS. Por ejemplo, puede especificar un configuración adicional de `kubelet`.

```
[settings.kubernetes.system-reserved]
cpu = "10m"
memory = "100Mi"
```

```
ephemeral-storage= "1Gi"
```

Para obtener más información acerca de la configuración admitida, consulte la documentación de Bottlerocket. Puede configurar etiquetas de nodos y taints en los datos de usuario. Sin embargo, recomendamos que las configure en su grupo de nodos. Amazon EKS aplica estas configuraciones cuando lo hace de este modo.

Cuando se fusionan los datos de usuario, el formato no se conserva, pero el contenido sigue siendo el mismo. La configuración que proporciona en los datos de usuario anula cualquier configuración configurada por Amazon EKS. Entonces, si configura `settings.kubernetes.max-pods` o `settings.kubernetes.cluster-dns-ip`, estos valores de los datos de usuario se aplican a los nodos.

Amazon EKS no admite todos los TOML válidos. A continuación, se muestra una lista de formatos conocidos no compatibles:

- Comillas dentro de las claves citadas: `'quoted "value"' = "value"`
- Comillas escapadas en valores: `str = "I'm a string. \"You can quote me\""`
- Flotantes y enteros mixtos: `numbers = [ 0.1, 0.2, 0.5, 1, 2, 5 ]`
- Tipos mixtos en matrices: `contributors = ["foo@example.com", { name = "Baz", email = "baz@example.com" }]`
- Encabezados entre corchetes con claves citadas: `[foo."bar.baz"]`

Windows user data

Los datos de usuario de Windows utilizan comandos de PowerShell. Al crear un grupo de nodos administrado, los datos de usuario personalizados se combinan con los datos de usuario administrados de Amazon EKS. Sus comandos de PowerShell son lo primero, seguidos de los comandos de datos de usuario administrados, todo dentro de una etiqueta **<powershell></powershell>**.

> **Note**
>
> Si no se especifica ningún ID de AMI en la plantilla de lanzamiento, no utilice el script Amazon EKS Bootstrap de Windows en los datos de usuario para configurar Amazon EKS.

Los datos de usuario de ejemplo son los siguientes.

```
<powershell>
Write-Host "Running custom user data script"
</powershell>
```

Especificación de una AMI

Si tiene alguno de los siguientes requisitos, especifique un ID de AMI en el campo `ImageId` de la plantilla de lanzamiento. Seleccione el requisito que tiene para obtener información adicional.

Proporcione datos de usuario a fin de pasar argumentos al archivo **`bootstrap.sh`** incluido con una AMI optimizada Linux/Bottlerocket para Amazon EKS

Bootstrapping es un término que se utiliza para describir la adición de comandos que se pueden ejecutar cuando se inicia una instancia. Por ejemplo, el arranque permite usar argumentos `kubelet` adicionales. Puede pasar los argumentos al script `bootstrap.sh` mediante `eksctl` sin especificar una configuración de lanzamiento. O puede hacerlo al especificar la información en la sección de datos de usuario de una plantilla de lanzamiento.

eksctl without specifying a launch template

Cree un archivo denominado *`my-nodegroup.yaml`* con el siguiente contenido. Sustituya cada *`example value`* con valores propios. Los argumentos `--apiserver-endpoint`, `--b64-cluster-ca` y `--dns-cluster-ip` son opcionales. Sin embargo, definirlos permite que el script `bootstrap.sh` evite crear una llamada `describeCluster`. Esto resulta útil en configuraciones de clústeres privados o clústeres en los que los nodos se reducen y escalan horizontalmente con frecuencia. Para obtener más información sobre el script `bootstrap.sh`, consulte el archivo `bootstrap.sh` en GitHub.

- El único argumento requerido en el nombre del clúster (*`my-cluster`*).
- Para recuperar el ID de una AMI optimizada para `ami-`*`1234567890abcdef0`*, puede utilizar las tablas en las secciones siguientes:
  - Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS
  - Recuperación de los ID de la AMI de Bottlerocket optimizada para Amazon EKS
  - Recuperación de los ID de la AMI de Windows optimizada para Amazon EKS
- Para recuperar el valor de *`certificate-authority`* de su clúster, ejecute el siguiente comando.

```
aws eks describe-cluster --query "cluster.certificateAuthority.data" --output text
 --name my-cluster --region region-code
```

- Para recuperar el *api-server-endpoint* de su clúster, ejecute el siguiente comando.

```
aws eks describe-cluster --query "cluster.endpoint" --output text --name my-
cluster --region region-code
```

- El valor de `--dns-cluster-ip` es su servicio de CIDR con `.10` al final. Para recuperar el *service-cidr* de su clúster, ejecute el siguiente comando. Por ejemplo, si el valor devuelto es `ipv4 10.100.0.0/16`, su valor es *10.100.0.10*.

```
aws eks describe-cluster --query "cluster.kubernetesNetworkConfig.serviceIpv4Cidr"
 --output text --name my-cluster --region region-code
```

- En este ejemplo proporciona un argumento `kubelet` para establecer un valor de `max-pods` personalizado mediante el script `bootstrap.sh` incluido con la AMI optimizada para Amazon EKS. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Para obtener ayuda con la selección de *my-max-pods-value*, consulte Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2.

```
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code

managedNodeGroups:
  - name: my-nodegroup
    ami: ami-1234567890abcdef0
    instanceType: m5.large
    privateNetworking: true
    disableIMDSv1: true
    labels: { x86-al2-specified-mng }
    overrideBootstrapCommand: |
      #!/bin/bash
      /etc/eks/bootstrap.sh my-cluster \
```

```
        --b64-cluster-ca certificate-authority \
        --apiserver-endpoint api-server-endpoint \
        --dns-cluster-ip service-cidr.10 \
        --kubelet-extra-args '--max-pods=my-max-pods-value' \
        --use-max-pods false
```

Para cada opción de archivo `eksctl config` disponible, consulte Config file schema (Esquema de archivo de configuración) en la documentación de `eksctl`. La utilidad `eksctl` sigue creando una plantilla de lanzamiento para usted y rellena sus datos de usuario con los datos que proporciona en el archivo `config`.

Cree un grupo de nodos con el siguiente comando.

```
eksctl create nodegroup --config-file=my-nodegroup.yaml
```

User data in a launch template

Especifique la siguiente información en la sección de datos de usuario de la plantilla de lanzamiento. Sustituya cada *example value* con valores propios. Los argumentos `--apiserver-endpoint`, `--b64-cluster-ca` y `--dns-cluster-ip` son opcionales. Sin embargo, definirlos permite que el script `bootstrap.sh` evite crear una llamada `describeCluster`. Esto resulta útil en configuraciones de clústeres privados o clústeres en los que los nodos se reducen y escalan horizontalmente con frecuencia. Para obtener más información sobre el script `bootstrap.sh`, consulte el archivo bootstrap.sh en GitHub.

- El único argumento requerido en el nombre del clúster (*my-cluster*).
- Para recuperar el valor de *certificate-authority* de su clúster, ejecute el siguiente comando.

  ```
  aws eks describe-cluster --query "cluster.certificateAuthority.data" --output text
   --name my-cluster --region region-code
  ```

- Para recuperar el *api-server-endpoint* de su clúster, ejecute el siguiente comando.

  ```
  aws eks describe-cluster --query "cluster.endpoint" --output text --name my-
  cluster --region region-code
  ```

- El valor de `--dns-cluster-ip` es su servicio de CIDR con `.10` al final. Para recuperar el *service-cidr* de su clúster, ejecute el siguiente comando. Por ejemplo, si el valor devuelto es `ipv4 10.100.0.0/16`, su valor es *10.100.0.10*.

```
aws eks describe-cluster --query "cluster.kubernetesNetworkConfig.serviceIpv4Cidr"
 --output text --name my-cluster --region region-code
```

- En este ejemplo proporciona un argumento `kubelet` para establecer un valor de `max-pods` personalizado mediante el script `bootstrap.sh` incluido con la AMI optimizada para Amazon EKS. Para obtener ayuda con la selección de *`my-max-pods-value`*, consulte Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
set -ex
/etc/eks/bootstrap.sh my-cluster \
  --b64-cluster-ca certificate-authority \
  --apiserver-endpoint api-server-endpoint \
  --dns-cluster-ip service-cidr.10 \
  --kubelet-extra-args '--max-pods=my-max-pods-value' \
  --use-max-pods false

--==MYBOUNDARY==--
```

Proporcione datos de usuario a fin de pasar argumentos al archivo **`Start-EKSBootstrap.ps1`** incluido con una AMI optimizada Windows para Amazon EKS

Bootstrapping es un término que se utiliza para describir la adición de comandos que se pueden ejecutar cuando se inicia una instancia. Puede pasar los argumentos al script `Start-EKSBootstrap.ps1` mediante `eksctl` sin especificar una configuración de lanzamiento. O puede hacerlo al especificar la información en la sección de datos de usuario de una plantilla de lanzamiento.

Si desea especificar un ID de AMI de Windows personalizado, tenga en cuenta las siguientes consideraciones:

- Debe utilizar una plantilla de lanzamiento y proporcionar los comandos de arranque necesarios en la sección de datos de usuario. Para recuperar el identificador deseado de Windows, puede utilizar la tabla de AMI de Windows optimizadas para Amazon EKS.
- Hay varios límites y condiciones. Por ejemplo, debe agregar `eks:kube-proxy-windows` a su mapa de configuración de IAM Authenticator de AWS. Para obtener más información, consulte Límites y condiciones al especificar un ID de AMI.

Especifique la siguiente información en la sección de datos de usuario de la plantilla de lanzamiento. Sustituya cada *example value* con valores propios. Los argumentos `-APIServerEndpoint`, `-Base64ClusterCA` y `-DNSClusterIP` son opcionales. Sin embargo, definirlos permite que el script `Start-EKSBootstrap.ps1` evite crear una llamada `describeCluster`.

- El único argumento requerido en el nombre del clúster (*my-cluster*).
- Para recuperar el valor de *certificate-authority* de su clúster, ejecute el siguiente comando.

```
aws eks describe-cluster --query "cluster.certificateAuthority.data" --output text --
name my-cluster --region region-code
```

- Para recuperar el *api-server-endpoint* de su clúster, ejecute el siguiente comando.

```
aws eks describe-cluster --query "cluster.endpoint" --output text --name my-cluster
 --region region-code
```

- El valor de `--dns-cluster-ip` es su servicio de CIDR con `.10` al final. Para recuperar el *service-cidr* de su clúster, ejecute el siguiente comando. Por ejemplo, si el valor devuelto es `ipv4 10.100.0.0/16`, su valor es *10.100.0.10*.

```
aws eks describe-cluster --query "cluster.kubernetesNetworkConfig.serviceIpv4Cidr" --
output text --name my-cluster --region region-code
```

- Para obtener argumentos adicionales, consulte Parámetros de configuración del script de arranque.

> **Note**
>
> Si utiliza un CIDR de servicio personalizado, debe especificarlo con el parámetro `-ServiceCIDR`. De lo contrario, se producirá un error en la resolución de DNS del clúster para Pods.

```
<powershell>
[string]$EKSBootstrapScriptFile = "$env:ProgramFiles\Amazon\EKS\Start-EKSBootstrap.ps1"
& $EKSBootstrapScriptFile -EKSClusterName my-cluster `
  -Base64ClusterCA certificate-authority `
  -APIServerEndpoint api-server-endpoint `
  -DNSClusterIP service-cidr.10
</powershell>
```

Ejecute una AMI personalizada debido a requisitos específicos de seguridad, conformidad o políticas internas

Para obtener más información, consulte Imágenes de máquina de Amazon (AMI) en la Guía del usuario de Amazon EC2. La especificación de compilación de la AMI de Amazon EKS contiene recursos y scripts de configuración para crear una AMI personalizada de Amazon EKS basada en Amazon Linux. Para obtener más información, consulte Especificación de compilación de AMI de Amazon EKS en GitHub. Para crear AMI personalizadas instaladas con otros sistemas operativos, consulte AMI personalizadas de ejemplo de Amazon EKS en GitHub.

> **Important**
>
> Al especificar una AMI, Amazon EKS no combina ningún dato de usuario. Más bien, usted es responsable de suministrar el comando `bootstrap` requerido para que los nodos se unan al clúster. Si los nodos no se unen al clúster, las acciones de Amazon EKS `CreateNodegroup` y `UpdateNodegroupVersion` también fallan.

Límites y condiciones al especificar un ID de AMI

A continuación se detallan los límites y las condiciones que implica especificar un ID de AMI con grupos de nodos administrados:

- Debe crear un nuevo grupo de nodos para cambiar entre especificar un ID de AMI en una plantilla de lanzamiento y no especificar un ID de AMI.
- No se le notifica en la consola cuando hay disponible una versión más reciente de AMI. Para actualizar el grupo de nodos a una versión de AMI más reciente, debe crear una nueva versión de la plantilla de lanzamiento con un ID de AMI actualizado. A continuación, debe actualizar el grupo de nodos con la nueva versión de plantilla de lanzamiento.
- Los siguientes campos no se pueden establecer en la API si especifica un ID de AMI:
  - `amiType`
  - `releaseVersion`
  - `version`
- Todas las `taints` configuradas en la API se aplican de manera asíncrona si se especifica un ID de AMI. Para aplicar taints antes de que un nodo se una al clúster, se deben transferir las taints a `kubelet` en sus datos de usuario mediante la marca `--register-with-taints` de la línea de comandos. Para obtener más información, consulte `kubelet` en la documentación del Kubernetes.
- Al especificar un ID de AMI personalizado para los grupos de nodos administrados de Windows, agregue `eks:kube-proxy-windows` al mapa de configuración de AWS IAM Authenticator. Esto es necesario para que el DNS funcione correctamente.

  1. Abra el mapa de configuración de IAM Authenticator de AWS para editarlo.

     ```
     kubectl edit -n kube-system cm aws-auth
     ```

  2. Agregue esta entrada a la lista de `groups` debajo de cada uno de los `rolearn` asociados con nodos de Windows. El mapa de configuración debería tener un aspecto similar a `aws-auth-cm-windows.yaml`.

     ```
     - eks:kube-proxy-windows
     ```

  3. Guarde el archivo y salga del editor de texto.

Eliminación de un grupo de nodos administrados

En este tema se describe cómo puede eliminar un grupo de nodos administrado de Amazon EKS. Al eliminar un grupo de nodos administrados, Amazon EKS establece primero el tamaño mínimo,

máximo y deseado del grupo de Auto Scaling en cero. Esto hace que el grupo de nodos se reduzca verticalmente.

Antes de finalizar cada instancia, Amazon EKS envía una señal para vaciar los Pods de ese nodo. Si los Pods no se han drenado después de unos minutos, Amazon EKS permite que Auto Scaling continúe con la finalización de la instancia. Una vez terminadas todas las instancias, se elimina el grupo de Auto Scaling.

> **Important**
>
> Si elimina un grupo de nodos administrado que utiliza un rol de IAM de un nodo que no se emplea en ningún otro grupo de nodos administrado en el clúster, el rol se quitará del `ConfigMap` de `aws-auth`. Si algún grupo de nodos autoadministrados del clúster utiliza el mismo rol de IAM del nodo, los nodos autoadministrados adoptarán el estado `NotReady`. Además, la operación del clúster también se ve interrumpida. Para añadir una asignación para el rol que está utilizando solo para los grupos de nodos autoadministrados, consulte Creación de entradas de acceso, si la versión de la plataforma de su clúster es al menos la versión mínima que aparece en la sección de requisitos previos de Administración de entradas de acceso. Si la versión de la plataforma es anterior a la versión mínima requerida para las entradas de acceso, puede volver a añadir la entrada al `ConfigMap` de `aws-auth`. Para obtener más información, ingrese `eksctl create iamidentitymapping --help` en su terminal.

Puede eliminar un grupo de nodos administrados con `eksctl` o la AWS Management Console.

eksctl

Para eliminar un grupo de nodos administrados con **`eksctl`**

Escriba el siguiente comando. Reemplace cada *`example value`* con valores propios.

```
eksctl delete nodegroup \
  --cluster my-cluster \
  --name my-mng \
  --region region-code
```

Para obtener más opciones, consulte Eliminar y drenar grupos de nodos en la documentación `eksctl`.

AWS Management Console

Para eliminar un grupo de nodos administrados mediante la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En la página Clústeres, elija el clúster que contiene el grupo de nodos que desea eliminar.
3. En la página del clúster, seleccione la pestaña Computar.
4. En la sección de Node Groups (Grupos de nodos), elija el grupo de nodos que desea eliminar. A continuación, elija Eliminar.
5. En el cuadro de diálogo de confirmación Eliminar grupo de nodos, introduzca el nombre del grupo de nodos. A continuación, elija Eliminar.

AWS CLI

Para eliminar un grupo de nodos administrados mediante la AWS CLI

1. Escriba el siguiente comando. Reemplace cada *example value* con valores propios.

```
aws eks delete-nodegroup \
  --cluster-name my-cluster \
  --nodegroup-name my-mng \
  --region region-code
```

2. Use las teclas de flecha del teclado para desplazarse por el resultado de la respuesta. Pulse la tecla **q** cuando termine.

Para obtener más opciones, consulte el comando `delete-nodegroup` en la referencia de comandos de la AWS CLI.

Nodos autoadministrados

Un clúster contiene uno o varios nodos de Amazon EC2 en los que están programados los Pods. Los nodos de Amazon EKS se ejecutan en su cuenta de AWS y se conectan con el plano de control del clúster a través del punto de conexión del servidor de la API del clúster. Se le factura por ellos en función de los precios de Amazon EC2. Para obtener más información, consulte Precios de Amazon EC2.

Un clúster puede contener varios grupos de nodos. Cada grupo de nodos contiene uno o varios nodos que se implementan en un grupo de Amazon EC2 Auto Scaling. El tipo de instancia de los nodos del grupo puede variar; por ejemplo, cuando se utiliza la selección del tipo de instancia basada en atributos con Karpenter. Todas las instancias de un grupo de nodos deben utilizar el rol de IAM del nodo de Amazon EKS.

Amazon EKS proporciona imágenes de máquina de Amazon (AMI) especializadas que se denominan AMI optimizadas para Amazon EKS. Las AMI están configuradas para funcionar con Amazon EKS. Sus componentes incluyen `containerd`, `kubelet` y el Autenticador de AWS IAM. Las AMI también contienen un script de arranque especializado que permite detectar el plano de control del clúster y conectarse automáticamente a él.

Si restringe el acceso al punto de conexión público del clúster mediante bloques de CIDR, recomendamos habilitar también el acceso al punto de conexión privado. Esto permite que los nodos se puedan comunicar con el clúster. Si el punto de conexión privado no está habilitado, los bloques de CIDR que especifique para el acceso público deben incluir los orígenes de salida de su VPC. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS.

Para agregar nodos autoadministrados a su clúster de Amazon EKS, consulte los temas siguientes. Si lanza nodos autoadministrados de forma manual, debe agregar la siguiente etiqueta a cada nodo. Para obtener más información, consulte Cómo agregar etiquetas a un recurso individual y eliminarlas de él. Si sigue los pasos de las siguientes guías, se agregará automáticamente la etiqueta necesaria a los nodos.

Clave	Valor
`kubernetes.io/cluster/` *`my-cluster`*	`owned`

Para obtener más información sobre los nodos desde un punto de vista general de Kubernetes, consulte Nodos en la documentación de Kubernetes.

Temas

- Lanzar nodos autoadministrados de Amazon Linux
- Lanzamiento de nodos de Bottlerocket autoadministrados
- Lanzamiento de nodos de Windows autoadministrados

- Lanzamiento de nodos de Ubuntu autoadministrados
- Actualizaciones de nodos autoadministrados

Lanzar nodos autoadministrados de Amazon Linux

En este tema, se describe cómo puede lanzar grupos de escalado automático de nodos de Linux que se registrará con el clúster de Amazon EKS. Una vez que los nodos se hayan unido al clúster, puede implementar aplicaciones de Kubernetes en ellos. También puede lanzar nodos de Amazon Linux autoadministrados con `eksctl` o la AWS Management Console. Si necesita lanzar nodos en AWS Outposts, consulte Lanzamiento de nodos autoadministrados de Amazon Linux en un Outpost.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Creación de un clúster de Amazon EKS. Si tiene subredes en la Región de AWS en la que está habilitado AWS Outposts, AWS Wavelength o AWS Local Zones, esas subredes no deben haberse presentado cuando se creó el clúster.
- Un rol de IAM existente para que lo utilicen los nodos. Para crear uno, consulte Rol de IAM de nodo de Amazon EKS. Si este rol no tiene ninguna de las políticas de la CNI de la VPC, es necesario el rol independiente que se indica a continuación para los pods de la CNI de la VPC.
- (Opcional pero recomendado) El complemento Amazon VPC CNI plugin for Kubernetes configurado con su propio rol de IAM que tenga adjunta la política de IAM necesaria. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
- Conocimiento de las consideraciones enumeradas en Elección de un tipo de instancia de Amazon EC2. Según el tipo de instancia que elija, es posible que haya requisitos previos adicionales para su clúster y VPC.

eksctl

> **Note**
>
> `eksctl` no es compatible con Amazon Linux 2023 en este momento.

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para Instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Cómo lanzar nodos de Linux autoadministrados mediante **`eksctl`**

1. (Opcional) Si la política administrada de IAM AmazonEKS_CNI_Policy se adjunta a su Rol de IAM de nodo de Amazon EKS, recomendamos asignarla a un rol de IAM asociado a la cuenta de servicios del `aws-node` de Kubernetes en su lugar. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
2. El siguiente comando crea un grupo de nodos en un clúster existente. Reemplace *`al-nodes`* por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Reemplace *`my-cluster`* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. Reemplace los *`example value`* restantes por sus propios valores. Los nodos se crean de forma predeterminada con la misma versión de Kubernetes que el plano de control.

   Antes de elegir un valor para `--node-type`, revise Elección de un tipo de instancia de Amazon EC2.

   Reemplace *`my-key`* con el nombre de su par de claves de Amazon EC2 o la clave pública. Esta clave se utiliza para SSH en sus nodos después de que se lancen. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2.

   Cree el grupo de nodos con el siguiente comando.

   > **Important**
   >
   > Si desea implementar un grupo de nodos en las subredes de AWS Outposts, Wavelength o bien Local Zones, existen consideraciones adicionales:
   >
   > - Las subredes no deben haberse transferido al crear el clúster.

- Debe crear el grupo de nodos con un archivo de configuración, que especifique las subredes y `volumeType: gp2`. Para obtener más información, consulte Crear un grupo de nodos a partir de un archivo de Config y el Esquema de archivo de configuración en la documentación de `eksctl`.

```
eksctl create nodegroup \
  --cluster my-cluster \
  --name al-nodes \
  --node-type t3.medium \
  --nodes 3 \
  --nodes-min 1 \
  --nodes-max 4 \
  --ssh-access \
  --managed=false \
  --ssh-public-key my-key
```

Para implementar un grupo de nodos que:

- pueda asignar un número significativamente mayor de direcciones IP a Pods que la configuración predeterminada, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.
- pueda asignar direcciones `IPv4` a Pods de un bloque de CIDR diferente que el de la instancia, consulte Redes personalizadas para los pods.
- pueda asignar direcciones `IPv6` a Pods y servicios, consulte Direcciones `IPv6` de clústeres, Pods y services.
- use el tiempo de ejecución `containerd`, debe implementar el grupo de nodos mediante un archivo de `config`. Para obtener más información, consulte Prueba de la migración de Docker a `containerd`.
- no tenga acceso saliente a internet, consulte Requisitos del clúster privado.

Para obtener una lista completa de todas las opciones y valores predeterminados disponibles, ingrese el siguiente comando.

```
eksctl create nodegroup --help
```

Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en la Guía de solución de problemas.

Un ejemplo de salida sería el siguiente. Se generan varias líneas mientras se crean los nodos. Una de las últimas líneas de salida es la siguiente línea de ejemplo.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Opcional) Implemente una aplicación de muestra para probar el clúster y los nodos de Linux.
4. Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

   - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
   - Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

   Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

AWS Management Console

Paso 1: lanzar nodos de Linux autoadministrados mediante la AWS Management Console

1. Descargue la versión más reciente de la plantilla de AWS CloudFormation.

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2022-12-23/amazon-eks-nodegroup.yaml
   ```

2. Espere a que el estado del clúster sea `ACTIVE`. Si lanza los nodos antes de que el clúster esté activo, los nodos no pueden registrarse con el clúster y tendrá que volver a lanzarlos.
3. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
4. Seleccione Create stack (Crear pila) y, a continuación, seleccione With new resources (standard) (Con nuevos recursos [estándar]).
5. Para Especificar plantilla, seleccione Cargar un archivo de plantilla y, a continuación, elija Elegir archivo.

6. Edite el archivo `amazon-eks-nodegroup.yaml` que ha descargado.
7. Seleccione Siguiente.
8. En la página Especificar detalles de la pila, ingrese los siguientes parámetros según corresponda y luego seleccione Siguiente:

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation. Por ejemplo, puede llamarla ***`my-cluster-nodes`***. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
   - ClusterName: ingrese el nombre que usó al crear el clúster de Amazon EKS. Este nombre debe coincidir con el nombre del clúster o los nodos no se podrán unir al clúster.
   - ClusterControlPlaneSecurityGroup: elija el valor de SecurityGroups en la salida de AWS CloudFormation que generó al crear la VPC.

     En los siguientes pasos, se muestra una operación para recuperar el grupo aplicable.

     1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
     2. Elija el nombre del clúster.
     3. Elija la pestaña Redes.
     4. Use el valor de Grupo de seguridad adicional como referencia al realizar una selección en la lista desplegable ClusterControlPlaneSecurityGroup.
   - NodeGroupName: escriba un nombre para el grupo de nodos. Este nombre se puede utilizar más adelante para identificar el grupo de nodos de escalado automático que se crea para los nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.
   - NodeAutoScalingGroupMinSize: ingrese el número mínimo de nodos al que se puede reducir horizontalmente el grupo de escalado automático de nodos.
   - NodeAutoScalingGroupDesiredCapacity: escriba el número deseado de nodos que desea escalar cuando se crea la pila.
   - NodeAutoScalingGroupMaxSize: ingrese el número máximo de nodos que pueda alcanzar el grupo de Auto Scaling de nodos.
   - NodeInstanceType: elija un tipo de instancia para los nodos. Para obtener más información, consulte Elección de un tipo de instancia de Amazon EC2.

- NodeImageIdSSMParam: rellenado previamente con el parámetro de Amazon EC2 Systems Manager de una AMI optimizada recientemente para Amazon EKS para una versión de Kubernetes variable. Para utilizar otra versión secundaria de Kubernetes compatible con Amazon EKS, reemplace *1.XX* por una versión admitida diferente. Recomendamos especificar la misma versión de Kubernetes que el clúster.

  También se puede sustituir *amazon-linux-2* por un tipo de AMI diferente. Para obtener más información, consulte Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS.

  > **Note**
  >
  > La AMI del nodo de Amazon EKS se basa en Amazon Linux. Puede realizar un seguimiento de los eventos de seguridad o privacidad de Amazon Linux 2 en el Centro de seguridad de Amazon Linux o suscribirse a la fuente RSS asociada. Los eventos de seguridad y privacidad incluyen información general del problema, qué paquetes están afectados y cómo actualizar las instancias para corregir el problema.

- NodeImageId: (opcional) si utiliza su propia AMI personalizada (en lugar de la AMI optimizada para Amazon EKS), ingrese un ID de AMI de nodo para su Región de AWS. Si especifica un valor aquí, anula cualquier valor del campo NodeImageIdSSMParam.
- NodeVolumeSize: especifique un tamaño de volumen raíz para los nodos en GiB.
- NodeVolumeType: especifique un tipo de volumen raíz para sus nodos.
- KeyName: ingrese el nombre de un par de claves SSH de Amazon EC2 que pueda utilizar para conectar mediante SSH con los nodos después de haberlos lanzado. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2.

  > **Note**
  >
  > Si no proporciona un par de claves aquí, se produce un error al crear la pila de AWS CloudFormation.

- BootstrapArguments: especifique los argumentos opcionales que se van a pasar al script de arranque del nodo, como los argumentos de `kubelet` adicionales. Para obtener más información, consulte la información de uso del script de arranque en GitHub.

  Para implementar un grupo de nodos que:

  - pueda asignar un número significativamente mayor de direcciones IP a Pods que la configuración predeterminada, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.
  - pueda asignar direcciones `IPv4` a Pods de un bloque de CIDR diferente que el de la instancia, consulte Redes personalizadas para los pods.
  - pueda asignar direcciones `IPv6` a Pods y servicios, consulte Direcciones `IPv6` de clústeres, Pods y services.
  - use el tiempo de ejecución `containerd`, debe implementar el grupo de nodos mediante un archivo de `config`. Para obtener más información, consulte Prueba de la migración de Docker a `containerd`.
  - no tenga acceso saliente a internet, consulte Requisitos del clúster privado.
- DisableIMDSv1: cada nodo admite de forma predeterminada la versión 1 (IMDSv1) e IMDSv2 del servicio de metadatos de la instancia. Puede desactivar IMDSv1. Para evitar que los nodos y Pods futuros del grupo de nodos utilicen MDSv1, defina DisableIMDSv1 en true (verdadero). Para obtener más información, consulte Configuración del servicio de metadatos de instancia. Para obtener más información sobre cómo restringir el acceso en sus nodos, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.
- VpcId: ingrese el ID de la VPC que ha creado.
- Subredes: elija las subredes que creó para la VPC. Si creó la VPC siguiendo los pasos que se describen en Creación de una VPC para su clúster de Amazon EKS, especifique solo las subredes privadas en la VPC en las que desea lanzar los nodos. Puede ver qué subredes son privadas abriendo cada enlace de subred desde la pestaña Redes de su clúster.

  > **Important**
  >
  > - Si alguna de las subredes es pública, debe tener habilitada la configuración de asignación automática de direcciones IP públicas. Si la configuración no está habilitada para la subred pública, los nodos que implemente en dicha subred pública no tendrán asignada una dirección IP pública y no podrán comunicarse

con el clúster u otros servicios de AWS. Si la subred se implementó antes del 26 de marzo de 2020 mediante cualquiera de las plantillas de VPC de AWS CloudFormation de Amazon EKS o mediante `eksctl`, la asignación automática de direcciones IP públicas se deshabilitará en las subredes públicas. Para obtener información acerca de cómo habilitar la asignación de direcciones IP públicas en una subred, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred. Si el nodo se implementa en una subred privada, podrá comunicarse con el clúster y otros servicios de AWS a través de una puerta de enlace NAT.

- Si las subredes no tienen acceso a Internet, asegúrese de que conoce las consideraciones y los pasos adicionales en Requisitos del clúster privado.
- Si selecciona las subredes de AWS Outposts, Wavelength o Local Zones, las subredes no se deben haber pasado cuando creó el clúster.

9. Seleccione las opciones que desee en la página Configurar las opciones de pila y, a continuación, elija Siguiente.
10. Seleccione la casilla de verificación situada a la izquierda de I acknowledge that AWS CloudFormation might create IAM resources. (Reconozco que podría crear recursos de IAM) y luego seleccione Create stack (Crear pila).
11. Una vez completada la creación de la pila, selecciónela en la consola y elija Salidas.
12. Anote el valor de NodeInstanceRoles correspondiente al grupo de nodos creado. Lo necesitará al configurar los nodos de Amazon EKS de .

Paso 2: permitir a los nodos unirse al clúster

> **Note**
>
> Si ha lanzado nodos dentro de una VPC privada sin acceso a Internet saliente, asegúrese de habilitar los nodos para que se unan al clúster desde dentro de la VPC.

1. Verifique si ya tiene el `ConfigMap` de `aws-auth`.

```
kubectl describe configmap -n kube-system aws-auth
```

2. Si se le muestra un `ConfigMap` de `aws-auth`, actualícelo según sea necesario.

a. Abra el icono `ConfigMap` para editar.

```
kubectl edit -n kube-system configmap/aws-auth
```

b. Añada una nueva entrada de `mapRoles` según sea necesario. Establezca el valor de `rolearn` en el valor de NodeInstanceRole que registró en el procedimiento anterior.

```
[...]
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
[...]
```

c. Guarde el archivo y salga del editor de texto.

3. Si recibe un error que indica "`Error from server (NotFound): configmaps "aws-auth" not found`, aplique el `ConfigMap` bursátil.

a. Descargue el mapa de configuración.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/aws-auth-cm.yaml
```

b. En el archivo `aws-auth-cm.yaml`, establezca el valor de `rolearn` al valor NodeInstanceRole que ha registrado en el procedimiento anterior. Puede hacerlo con un editor de texto o reemplazando *`my-node-instance-role`* y ejecute el siguiente comando:

```
sed -i.bak -e 's|<ARN of instance role (not instance profile)>|my-node-
instance-role|' aws-auth-cm.yaml
```

c. Aplique la configuración. Este comando puede tardar varios minutos en finalizar.

```
kubectl apply -f aws-auth-cm.yaml
```

4. Observe el estado de los nodos y espere a que aparezca el estado `Ready`.

```
kubectl get nodes --watch
```

Ingrese `Ctrl+C` para obtener un símbolo del intérprete de comandos.

> **Note**
>
> Si recibe cualquier error de tipo de recurso o autorización, consulte Acceso denegado o no autorizado (`kubectl`) en el tema de solución de problemas.

Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en la guía de solución de problemas.

5. (Solo para nodos de GPU) Si ha elegido un tipo de instancia de GPU y la AMI acelerada optimizada para Amazon EKS, debe aplicar el complemento de dispositivo NVIDIA para Kubernetes como un DaemonSet en su clúster. Reemplace *vX.X.X* con la versión Plugin de dispositivo NVidia/K8S deseada antes de ejecutar el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
```

Paso 3: acciones adicionales

1. (Opcional) Implemente una aplicación de muestra para probar el clúster y los nodos de Linux.
2. (Opcional) Si la política de IAM administrada AmazonEKS_CNI_Policy (si tiene un clúster `IPv4`) o la *AmazonEKS_CNI_IPv6_Policy* (que usted haya creado si tiene un clúster `IPv6`) están adjuntas a su the section called “Rol de IAM de nodo”, le recomendamos asignarlas, en cambio, a un rol de IAM que asocie a la cuenta de servicio de `aws-node` de Kubernetes. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
3. Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

   - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.

- Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Bloques de capacidad para ML

> **Important**
>
> - Los bloques de capacidad solo están disponibles para determinados tipos de instancias de Amazon EC2 y Regiones de AWS. Para obtener información sobre compatibilidad, consulte los requisitos previos para trabajar con bloques de capacidad en la Guía del usuario de Amazon EC2 para instancias de Linux.
> - Los bloques de capacidad actualmente no se pueden usar con grupos de nodos administrados por Amazon EKS o Karpenter.

Los bloques de capacidad para machine learning (ML) permiten reservar instancias de GPU en una fecha futura para respaldar cargas de trabajo de ML de corta duración. Las instancias que se ejecutan dentro de un bloque de capacidad se ubican juntas automáticamente dentro de ultraclústeres de Amazon EC2, de modo que no es necesario utilizar un grupo con ubicación en clúster. Para obtener más información, consulte Bloques de capacidad para ML en la Guía del usuario de Amazon EC2 para instancias de Linux.

Puede utilizar bloques de capacidad con Amazon EKS para aprovisionar y escalar nodos autoadministrados. En los siguientes pasos se ofrece un ejemplo general.

1. Cree una plantilla de lanzamiento en la AWS Management Console. Para obtener más información, consulte Use Capacity Blocks for machine learning workloads en la Guía del usuario de Amazon EC2 Auto Scaling.

   Asegúrese de incluir la configuración del tipo de instancia e Imagen de máquina de Amazon (AMI).

2. Vincule el bloque de capacidad a una plantilla de lanzamiento mediante el ID de reserva de capacidad.

   A continuación se muestra un ejemplo de plantilla de AWS CloudFormation para crear una plantilla de lanzamiento destinada a un bloque de capacidad:

   ```
   NodeLaunchTemplate:
       Type: "AWS::EC2::LaunchTemplate"
       Properties:
         LaunchTemplateData:
           InstanceMarketOptions:
             MarketType: "capacity-block"
           CapacityReservationSpecification:
             CapacityReservationTarget:
               CapacityReservationId: "cr-02168da1478b509e0"
           IamInstanceProfile:
             Arn: iam-instance-profile-arn
           ImageId: image-id
           InstanceType: p5.48xlarge
           KeyName: key-name
           SecurityGroupIds:
           - sg-05b1d815d1EXAMPLE
           UserData: user-data
   ```

   Debe pasar la subred de la zona de disponibilidad en la que se realiza la reserva, ya que los bloques de capacidad son zonales.

3. Si va a crear el grupo de nodos autoadministrados antes de que se active la reserva de capacidad, defina la capacidad deseada como `0`. Al crear el grupo de nodos, asegúrese de especificar únicamente la subred correspondiente a la zona de disponibilidad en la que se va a reservar la capacidad.

   A continuación se muestra un ejemplo de plantilla de CloudFormation que se puede utilizar. En este ejemplo se obtienen `LaunchTemplateId` y `Version` del recurso `AWS::Amazon EC2::LaunchTemplate` que se muestra en el ejemplo anterior. También se obtienen los valores de `DesiredCapacity`, `MaxSize`, `MinSize` y `VPCZoneIdentifier` que se declaran en otras partes de la misma plantilla.

   ```
     NodeGroup:
       Type: "AWS::AutoScaling::AutoScalingGroup"
       Properties:
   ```

```
      DesiredCapacity: !Ref NodeAutoScalingGroupDesiredCapacity
      LaunchTemplate:
        LaunchTemplateId: !Ref NodeLaunchTemplate
        Version: !GetAtt NodeLaunchTemplate.LatestVersionNumber
      MaxSize: !Ref NodeAutoScalingGroupMaxSize
      MinSize: !Ref NodeAutoScalingGroupMinSize
      VPCZoneIdentifier: !Ref Subnets
      Tags:
        - Key: Name
          PropagateAtLaunch: true
          Value: !Sub ${ClusterName}-${NodeGroupName}-Node
        - Key: !Sub kubernetes.io/cluster/${ClusterName}
          PropagateAtLaunch: true
          Value: owned
```

4. Una vez que el grupo de nodos se ha creado correctamente, asegúrese de registrar el valor de `NodeInstanceRole` correspondiente al grupo de nodos que se ha creado. Esto es necesario para asegurarse de que, al escalar el grupo de nodos, los nuevos nodos se unan al clúster y Kubernetes pueda reconocerlos. Para obtener más información, consulte las instrucciones de la AWS Management Console en Lanzar nodos autoadministrados de Amazon Linux.
5. Se recomienda crear una política de escalado programado para el grupo de escalado automático que se ajuste a los tiempos de reserva del bloque de capacidad. Para obtener más información, consulte Scheduled scaling for Amazon EC2 Auto Scaling en la Guía del usuario de Amazon EC2 Auto Scaling.

   Puede utilizar todas las instancias que haya reservado hasta 30 minutos antes de la hora de finalización del bloque de capacidad. Las instancias que aún estén en ejecución en ese momento comenzarán a finalizar. Para que los nodos dispongan de tiempo suficiente para vaciarse correctamente, le sugerimos que programe el escalado para que llegue a cero más de 30 minutos antes de la hora de finalización de la reserva del bloque de capacidad.

   Si en lugar de esto desea escalar verticalmente cada vez que el estado de la reserva de capacidad sea `Active`, debe actualizar la capacidad deseada para el grupo de escalado automático a la hora de inicio de la reserva del bloque de capacidad. En ese caso, también tendría que reducir verticalmente y manualmente más de 30 minutos antes de la hora de finalización de la reserva del bloque de capacidad.
6. El grupo de nodos ya está listo para la programación de cargas de trabajo y Pods.
7. Para vaciar los Pods correctamente, se recomienda configurar AWS Node Termination Handler. Este controlador podrá observar eventos del ciclo de vida de “reducción horizontal de grupos

de escalado automático" de Amazon EC2 Auto Scaling mediante EventBridge y permitir que el plano de control de Kubernetes tome las medidas necesarias antes de que la instancia deje de estar disponible. De lo contrario, los objetos de Pods y Kubernetes quedarán bloqueados en estado pendiente. Para obtener más información, consulte AWS Node Termination Handler en GitHub.

Si no configura un controlador de finalización de nodos, le recomendamos que comience a vaciar los Pods manualmente antes de la ventana de 30 minutos, con el fin de que tengan tiempo suficiente para vaciarse correctamente.

Lanzamiento de nodos de Bottlerocket autoadministrados

> **Note**
>
> Los grupos de nodos administrados podrían ofrecer algunas ventajas para su caso de uso. Para obtener más información, consulte Grupos de nodos administrados.

En este tema se describe cómo iniciar grupos de escalado automático de nodos Bottlerocket que se registrará en el clúster de Amazon EKS. Bottlerocket es un sistema operativo de código abierto basado en Linux de AWS que puede utilizar para ejecutar contenedores en máquinas virtuales o anfitriones bare metal. Una vez que los nodos se hayan unido al clúster, puede implementar aplicaciones de Kubernetes en ellos. Para obtener más información acerca de Bottlerocket, consulte Uso de una AMI de Bottlerocket con Amazon EKS en GitHub y Compatibilidad con AMI personalizada en la documentación de `eksctl`.

Para obtener información sobre actualizaciones en contexto, consulte Operador de actualización de Bottlerocket en GitHub.

> **Important**
>
> - Los nodos de Amazon EKS son instancias estándar de Amazon EC2 y se les facturarán conforme a los precios ordinarios de las instancias de Amazon EC2. Para obtener más información, consulte Precios de Amazon EC2.
> - Puede lanzar nodos de Bottlerocket en clústeres extendidos de Amazon EKS en AWS Outposts, pero no puede lanzarlos en clústeres locales en AWS Outposts. Para obtener más información, consulte Amazon EKS en AWS Outposts.

- Puede implementar en instancias de Amazon EC2 con procesadores x86 o Arm. Sin embargo, no puede implementar en instancias que tienen chips Inferentia.
- Bottlerocket es compatible con AWS CloudFormation. Sin embargo, no existe ninguna plantilla oficial de CloudFormation que pueda copiarse para implementar nodos Bottlerocket para Amazon EKS.
- Las imágenes de Bottlerocket no vienen con un servidor SSH ni un intérprete de comandos. Puede usar métodos de acceso fuera de banda para permitir que SSH habilite el contenedor de administrador y superar algunos pasos de configuración de arranque con datos de usuario. Para obtener más información, consulte estas secciones en bottlerocket README.md en GitHub:
  - Exploration (Exploración)
  - Contenedor de administrador
  - Configuración de Kubernetes

Para lanzar nodos de Bottlerocket con **`eksctl`**

En este procedimiento, se requiere la versión `0.183.0` o posterior de `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

> **Note**
>
> Este procedimiento solo es válido para los clústeres que se crearon con `eksctl`.

1. Copie los siguientes contenidos en su dispositivo. Reemplace *`my-cluster`* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. Reemplace *`ng-bottlerocket`* por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como

caracteres no iniciales. Para implementar en instancias Arm, reemplace *m5.large* por un tipo de instancia Arm. Sustituya *my-ec2-keypair-name* por el nombre de un par de claves SSH de Amazon EC2 que pueda utilizar para conectar mediante SSH con los nodos después de haberlos lanzado. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2. Sustituya todos los *example values* restantes por sus propios valores. Una vez que haya llevado a cabo las sustituciones, ejecute el comando modificado para crear el archivo `bottlerocket.yaml`.

Si especifica un tipo de instancia Arm de Amazon EC2, revise las consideraciones en AMI de Amazon Linux optimizada para Amazon EKS Arm antes de llevar a cabo la implementación. Para ver instrucciones sobre cómo implementar mediante una AMI personalizada, consulte Creación de Bottlerocket en GitHub y Compatibilidad con AMI personalizada en la documentación de `eksctl`. Para implementar un grupo de nodos administrados, implemente una AMI personalizada mediante el uso de una plantilla de lanzamiento. Para obtener más información, consulte Personalización de nodos administrados con plantillas de lanzamiento.

> **Important**
>
> Para implementar un grupo de nodos en las subredes de AWS Outposts, AWS Wavelength o AWS Local Zones, no pase las subredes de AWS Outposts, AWS Wavelength o AWS Lozal Zones al crear el clúster. Debe especificar las subredes en el siguiente ejemplo. Para obtener más información, consulte Crear un grupo de nodos a partir de un archivo de Config y el Esquema de archivo de configuración en la documentación de `eksctl`. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster.

```
cat >bottlerocket.yaml <<EOF
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: '1.30'
```

```
iam:
  withOIDC: true

nodeGroups:
  - name: ng-bottlerocket
    instanceType: m5.large
    desiredCapacity: 3
    amiFamily: Bottlerocket
    ami: auto-ssm
    iam:
       attachPolicyARNs:
          - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
          - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
          - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
          - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
    ssh:
        allow: true
        publicKeyName: my-ec2-keypair-name
EOF
```

2. Implemente los nodos con el siguiente comando.

```
eksctl create nodegroup --config-file=bottlerocket.yaml
```

Un ejemplo de salida sería el siguiente.

Se generan varias líneas mientras se crean los nodos. Una de las últimas líneas de salida es la siguiente línea de ejemplo.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Opcional) Cree un volumen persistente de Kubernetes en un nodo de Bottlerocket mediante el Complemento CSI de Amazon EBS. El controlador predeterminado de Amazon EBS se basa en herramientas del sistema de archivos que no están incluidas con Bottlerocket. Para obtener información adicional acerca de cómo crear una clase de almacenamiento mediante un controlador, consulte Controlador CSI de Amazon EBS.
4. (Opcional) De forma predeterminada, `kube-proxy` establece el parámetro del kernel `nf_conntrack_max` en un valor predeterminado que puede diferir de lo que Bottlerocket establece inicialmente en el arranque. Para mantener la configuración predeterminada de Bottlerocket, edite la configuración `kube-proxy` con el siguiente comando.

```
kubectl edit -n kube-system daemonset kube-proxy
```

Agregue `--conntrack-max-per-core` y `--conntrack-min` a los argumentos `kube-proxy` que se encuentran en el siguiente ejemplo. Una configuración de `0` implica que no hay cambios.

```
      containers:
      - command:
        - kube-proxy
        - --v=2
        - --config=/var/lib/kube-proxy-config/config
        - --conntrack-max-per-core=0
        - --conntrack-min=0
```

5. (Opcional) Implemente una aplicación de muestra para probar los nodos de Bottlerocket.
6. Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

   - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
   - Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

   Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Lanzamiento de nodos de Windows autoadministrados

En este tema, se describe cómo puede lanzar un grupo de escalado automático de nodos de Windows que se registrará con el clúster de Amazon EKS. Una vez que los nodos se hayan unido al clúster, puede implementar aplicaciones de Kubernetes en ellos.

> **Important**
>
> - Los nodos de Amazon EKS son instancias estándar de Amazon EC2 y se les facturarán conforme a los precios ordinarios de las instancias de Amazon EC2. Para obtener más información, consulte Precios de Amazon EC2.

- Puede lanzar nodos de Windows en clústeres extendidos de Amazon EKS en AWS Outposts, pero no puede lanzarlos en clústeres locales en AWS Outposts. Para obtener más información, consulte Amazon EKS en AWS Outposts.

Habilite la compatibilidad con Windows para su clúster. Recomendamos que revise las consideraciones importantes antes de lanzar un grupo de nodos de Windows. Para obtener más información, consulte Habilitación de la compatibilidad con Windows.

Puede lanzar nodos de Windows autoadministrados con `eksctl` o la AWS Management Console.

eksctl

Para lanzar nodos de Windows autoadministrados mediante **`eksctl`**

En este procedimiento, se presupone que ha instalado `eksctl` y que su versión de `eksctl` es al menos `0.183.0`. Puede verificar la versión con el siguiente comando.

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

> **Note**
>
> Este procedimiento solo es válido para los clústeres que se crearon con `eksctl`.

1. (Opcional) Si la política de IAM administrada AmazonEKS_CNI_Policy (si tiene un clúster `IPv4`) o la *`AmazonEKS_CNI_IPv6_Policy`* (que usted haya creado si tiene un clúster `IPv6`) están adjuntas a su the section called “Rol de IAM de nodo”, le recomendamos asignarlas, en cambio, a un rol de IAM que asocie a la cuenta de servicio de `aws-node` de Kubernetes. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
2. En este procedimiento, se presupone que dispone de un clúster existente. Si aún no tiene un clúster de Amazon EKS ni un grupo de nodos de Amazon Linux al cual añadirle un grupo de nodos de Windows, le recomendamos que siga la guía Introducción a Amazon EKS: `eksctl`. En esta se proporciona una explicación completa para crear un clúster de Amazon EKS con nodos de Amazon Linux.

Cree el grupo de nodos con el siguiente comando. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Reemplace *my-cluster* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. Reemplace *ng-windows* por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Para Kubernetes versión `1.24` o posterior, puede reemplazar *2019* por `2022` para usar Windows Server 2022. Sustituya el resto de *example values* por sus propios valores.

> **Important**
>
> Para implementar un grupo de nodos en las subredes de AWS Outposts, AWS Wavelength o AWS Local Zones, no pase las subredes de AWS Outposts, Wavelength o Lozal Zones al crear el clúster. Cree el grupo de nodos con un archivo de configuración, que especifique las subredes de AWS Outposts, Wavelength o Local Zone. Para obtener más información, consulte Crear un grupo de nodos a partir de un archivo de Config y el Esquema de archivo de configuración en la documentación de `eksctl`.

```
eksctl create nodegroup \
    --region region-code \
    --cluster my-cluster \
    --name ng-windows \
    --node-type t2.large \
    --nodes 3 \
    --nodes-min 1 \
    --nodes-max 4 \
    --managed=false \
    --node-ami-family WindowsServer2019FullContainer
```

> **Note**
>
> - Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en la Guía de solución de problemas.
> - Para ver las opciones disponibles para los comandos `eksctl`, ingrese el siguiente comando.
>
> ```
> eksctl command -help
> ```

Un ejemplo de salida sería el siguiente. Se generan varias líneas mientras se crean los nodos. Una de las últimas líneas de salida es la siguiente línea de ejemplo.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Opcional) Implemente una aplicación de muestra para probar el clúster y los nodos de Windows.
4. Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:
   - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
   - Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

   Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

AWS Management Console

Requisitos previos

- Un clúster de Amazon EKS existente y un grupo de nodos de Linux. Si no tiene estos recursos, recomendamos que siga una de nuestras guías Introducción a Amazon EKS para crearlos. En las guías se describe cómo crear un clúster de Amazon EKS con nodos de Linux.

- Una VPC y un grupo de seguridad existentes que cumplen los requisitos para un clúster de Amazon EKS. Para obtener más información, consulte Requisitos y consideraciones de Amazon EKS VPC y subred y Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS. En la guía Introducción a Amazon EKS, se crea una VPC que cumple los requisitos. Si lo desea, también puede seguir Creación de una VPC para su clúster de Amazon EKS para crear una nueva manualmente.
- Un clúster de Amazon EKS existente que utiliza una VPC y un grupo de seguridad que cumplen los requisitos de un clúster de Amazon EKS. Para obtener más información, consulte Creación de un clúster de Amazon EKS. Si tiene subredes en la Región de AWS donde están habilitados AWS Outposts, AWS Wavelength o AWS Local Zones, las subredes no se deben haber pasado al crear el clúster.

Paso 1: lanzar nodos de Windows autoadministrados mediante la AWS Management Console

1. Espere a que el estado del clúster sea `ACTIVE`. Si lanza los nodos antes de que el clúster esté activo, los nodos no pueden registrarse con el clúster y debe volver a lanzarlos.
2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. Elija Crear pila.
4. En Especificar plantilla, seleccione URL de Amazon S3.
5. Copie la siguiente URL y péguela en la URL de Amazon S3.

   ```
   https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2023-02-09/amazon-eks-windows-nodegroup.yaml
   ```

6. Seleccione Siguiente dos veces.
7. En la página Creación rápida de pila, ingrese los siguientes parámetros según corresponda:

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation. Por ejemplo, puede llamarla ***my-cluster*-nodes**.
   - ClusterName: ingrese el nombre que usó al crear el clúster de Amazon EKS.

     > **Important**
     >
     > Este nombre debe coincidir exactamente con el nombre que utilizó en Paso 1: crear el clúster de Amazon EKS. De lo contrario, los nodos no podrán unirse al clúster.

- ClusterControlPlaneSecurityGroup: elija el grupo de seguridad de la salida de AWS CloudFormation que generó al crear la VPC.

  En los pasos siguientes se muestra un método para recuperar el grupo aplicable.

  1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
  2. Elija el nombre del clúster.
  3. Elija la pestaña Redes.
  4. Use el valor de Grupo de seguridad adicional como referencia al realizar una selección en la lista desplegable ClusterControlPlaneSecurityGroup.
- NodeGroupName: escriba un nombre para el grupo de nodos. Este nombre se puede utilizar más adelante para identificar el grupo de nodos de escalado automático que se crea para los nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.
- NodeAutoScalingGroupMinSize: ingrese el número mínimo de nodos al que se puede reducir horizontalmente el grupo de escalado automático de nodos.
- NodeAutoScalingGroupDesiredCapacity: escriba el número deseado de nodos que desea escalar cuando se crea la pila.
- NodeAutoScalingGroupMaxSize: ingrese el número máximo de nodos que pueda alcanzar el grupo de Auto Scaling de nodos.
- NodeInstanceType: elija un tipo de instancia para los nodos. Para obtener más información, consulte Elección de un tipo de instancia de Amazon EC2.

  > **Note**
  >
  > Los tipos de instancias compatibles con la versión más reciente del Amazon VPC CNI plugin for Kubernetes se muestran en vpc_ip_resource_limit.go en GitHub. Es posible que tenga que actualizar la versión de CNI para utilizar los tipos de instancia admitidos más recientes. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.
- NodeImageIdSSMParam: contiene un valor especificado previamente, que es el parámetro de Amazon EC2 Systems Manager del ID de la AMI de Windows Core optimizada para

Amazon EKS más reciente recomendada. Para utilizar la versión completa de Windows, sustituya *Core* por `Full`.

- NodeImageId: (opcional) si utiliza su propia AMI personalizada (en lugar de la AMI optimizada para Amazon EKS), ingrese un ID de AMI de nodo para su Región de AWS. Si especifica un valor para este campo, anula cualquier valor del campo NodeImageIdSSMParam.
- NodeVolumeSize: especifique un tamaño de volumen raíz para los nodos en GiB.
- KeyName: ingrese el nombre de un par de claves SSH de Amazon EC2 que pueda utilizar para conectar mediante SSH con los nodos después de haberlos lanzado. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2.

  > **Note**
  >
  > Si no proporciona un par de claves aquí, se produce un error al crear la pila de AWS CloudFormation.

- BootstrapArguments: especifique los argumentos opcionales que se van a pasar al script de arranque del nodo, como los argumentos de `kubelet` adicionales mediante `-KubeletExtraArgs`.
- DisableIMDSv1: cada nodo admite de forma predeterminada la versión 1 (IMDSv1) e IMDSv2 del servicio de metadatos de la instancia. Puede desactivar IMDSv1. Para evitar que los nodos y Pods futuros del grupo de nodos utilicen MDSv1, defina DisableIMDSv1 en true (verdadero). Para obtener más información, consulte Configurar el servicio de metadatos de instancia.
- VpcId: seleccione el ID de la VPC que creó.
- NodeSecurityGroups: seleccione el grupo de seguridad que se creó para su grupo de nodos de Linux cuando creó su VPC. Si sus nodos de Linux tienen más de un grupo de seguridad adjuntado a ellos, especifíquelos a todos aquí. Esto, por ejemplo, si el grupo de nodos Linux se creó con `eksctl`.
- Subredes: elija las subredes que creó. Si creó la VPC siguiendo los pasos que se describen en Creación de una VPC para su clúster de Amazon EKS, especifique solo las subredes privadas en la VPC en las que desea lanzar los nodos.

> **Important**
>
> - Si alguna de las subredes es pública, debe tener habilitada la configuración de asignación automática de direcciones IP públicas. Si la configuración no está habilitada para la subred pública, los nodos que implemente en dicha subred pública no tendrán asignada una dirección IP pública y no podrán comunicarse con el clúster u otros servicios de AWS. Si la subred se implementó antes del 26 de marzo de 2020 mediante cualquiera de las plantillas de VPC de AWS CloudFormation de Amazon EKS o mediante `eksctl`, la asignación automática de direcciones IP públicas se deshabilitará en las subredes públicas. Para obtener información acerca de cómo habilitar la asignación de direcciones IP públicas en una subred, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred. Si el nodo se implementa en una subred privada, podrá comunicarse con el clúster y otros servicios de AWS a través de una puerta de enlace NAT.
> - Si las subredes no tienen acceso a Internet, asegúrese de que conoce las consideraciones y los pasos adicionales en Requisitos del clúster privado.
> - Si selecciona las subredes de AWS Outposts, Wavelength o Local Zones, las subredes no se deben haber pasado cuando creó el clúster.

8. Confirme que la pila pueda crear recursos de IAM y, a continuación, seleccione Crear pila.
9. Una vez completada la creación de la pila, selecciónela en la consola y elija Salidas.
10. Anote el valor de NodeInstanceRoles correspondiente al grupo de nodos creado. Lo necesitará al configurar los nodos de Amazon EKS de Windows.

Paso 2: permitir a los nodos unirse al clúster

1. Verifique si ya tiene el `ConfigMap` de `aws-auth`.

```
kubectl describe configmap -n kube-system aws-auth
```

2. Si se le muestra un `ConfigMap` de `aws-auth`, actualícelo según sea necesario.

   a. Abra el icono `ConfigMap` para editar.

```
kubectl edit -n kube-system configmap/aws-auth
```

b. Añada nuevas entradas de `mapRoles` según sea necesario. Establezca los valores de `rolearn` en los valores de NodeInstanceRole que registró en los procedimientos anteriores.

```
[...]
data:
  mapRoles: |
- rolearn: <ARN of linux instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
    - rolearn: <ARN of windows instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
        - eks:kube-proxy-windows
[...]
```

c. Guarde el archivo y salga del editor de texto.

3. Si recibe un error que indica "`Error from server (NotFound): configmaps "aws-auth" not found`, aplique el `ConfigMap` bursátil.

a. Descargue el mapa de configuración.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/aws-auth-cm-windows.yaml
```

b. En el archivo `aws-auth-cm-windows.yaml`, establezca los valores de `rolearn` a los valores NodeInstanceRole que ha registrado en el procedimiento anterior. Puede hacerlo con un editor de texto o reemplazando *`example values`* y ejecutando el siguiente comando:

```
sed -i.bak -e 's|<ARN of linux instance role (not instance profile)>|my-
node-linux-instance-role|' \
    -e 's|<ARN of windows instance role (not instance profile)>|my-node-
windows-instance-role|' aws-auth-cm-windows.yaml
```

> **Important**
>
> - No modifique ninguna otra línea de este archivo.
> - No utilice el mismo rol de IAM para los nodos de Windows y Linux.

c. Aplique la configuración. Este comando podría tardar varios minutos en finalizar.

```
kubectl apply -f aws-auth-cm-windows.yaml
```

4. Observe el estado de los nodos y espere a que aparezca el estado `Ready`.

```
kubectl get nodes --watch
```

Ingrese `Ctrl+C` para obtener un símbolo del intérprete de comandos.

> **Note**
>
> Si recibe cualquier error de tipo de recurso o autorización, consulte Acceso denegado o no autorizado (`kubectl`) en el tema de solución de problemas.

Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en la guía de solución de problemas.

Paso 3: acciones adicionales

1. (Opcional) Implemente una aplicación de muestra para probar el clúster y los nodos de Windows.
2. (Opcional) Si la política de IAM administrada AmazonEKS_CNI_Policy (si tiene un clúster `IPv4`) o la *`AmazonEKS_CNI_IPv6_Policy`* (que usted haya creado si tiene un clúster `IPv6`) están adjuntas a su the section called “Rol de IAM de nodo”, le recomendamos asignarlas, en cambio, a un rol de IAM que asocie a la cuenta de servicio de `aws-node` de Kubernetes. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
3. Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

- Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
- Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Lanzamiento de nodos de Ubuntu autoadministrados

> **Note**
>
> Los grupos de nodos administrados podrían ofrecer algunas ventajas para su caso de uso. Para obtener más información, consulte Grupos de nodos administrados.

En este tema, se describe cómo puede lanzar un grupo de escalado automático de Ubuntu en Amazon Elastic Kubernetes Service (EKS) o Ubuntu Pro en Amazon Elastic Kubernetes Service (EKS) que se registrará con el clúster de Amazon EKS. Ubuntu y Ubuntu Pro para EKS se basan en el Ubuntu Minimal LTS oficial, incluyen el núcleo AWS personalizado desarrollado conjuntamente con AWS y se han creado específicamente para EKS. Ubuntu Pro añade una cobertura de seguridad adicional al admitir los períodos de soporte ampliados de EKS, el núcleo livepatch, el cumplimiento de las normas FIPS y la capacidad de ejecutar un número ilimitado de contenedores Pro.

Una vez que los nodos se hayan unido al clúster, puede implementar aplicaciones de contenedores en ellos. Para obtener más información, consulte la documentación sobre Ubuntu en AWS y la compatibilidad con AMI personalizada en la documentación `eksctl`.

> **Important**
>
> - Los nodos de Amazon EKS son instancias estándar de Amazon EC2 y se les facturarán conforme a los precios ordinarios de las instancias de Amazon EC2. Para obtener más información, consulte Precios de Amazon EC2.

- Puede lanzar nodos de Ubuntu en clústeres extendidos de Amazon EKS en AWS Outposts, pero no puede lanzarlos en clústeres locales en AWS Outposts. Para obtener más información, consulte Amazon EKS en AWS Outposts.
- Puede implementar en instancias de Amazon EC2 con procesadores `x86` o Arm. Sin embargo, es posible que las instancias que tienen chips Inferentia deban instalar primero el Neuron SDK.

Para lanzar Ubuntu para EKS o Ubuntu Pro para nodos EKS mediante **`eksctl`**

En este procedimiento, se requiere la versión `0.183.0` o posterior de `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

> **Note**
>
> Este procedimiento solo es válido para los clústeres que se crearon con `eksctl`.

1. Copie los siguientes contenidos en su dispositivo. Reemplace `my-cluster` por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfabético y no puede tener más de 100 caracteres. Reemplace `ng-ubuntu` por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Para implementar en instancias Arm, reemplace `m5.large` por un tipo de instancia Arm. Sustituya `my-ec2-keypair-name` por el nombre de un par de claves SSH de Amazon EC2 que pueda utilizar para conectar mediante SSH con los nodos después de haberlos lanzado. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2. Sustituya todos los *`example values`* restantes por sus propios valores. Una vez que haya llevado a cabo las sustituciones, ejecute el comando modificado para crear el archivo `ubuntu.yaml`.

> **Important**
>
> Para implementar un grupo de nodos en las subredes de AWS Outposts, AWS Wavelength o AWS Local Zones, no pase las subredes de AWS Outposts, AWS Wavelength o AWS Lozal Zones al crear el clúster. Debe especificar las subredes en el siguiente ejemplo. Para obtener más información, consulte Crear un grupo de nodos a partir de un archivo de Config y el Esquema de archivo de configuración en la documentación de `eksctl`. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster.

```
cat >ubuntu.yaml <<EOF
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: '1.30'

iam:
  withOIDC: true

nodeGroups:
  - name: ng-ubuntu
    instanceType: m5.large
    desiredCapacity: 3
    amiFamily: Ubuntu22.04
    ami: auto-ssm
    iam:
       attachPolicyARNs:
          - arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
          - arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
          - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
          - arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
    ssh:
        allow: true
        publicKeyName: my-ec2-keypair-name
EOF
```

Para crear un grupo de nodos Ubuntu Pro, simplemente cambie el valor `amiFamily` a `UbuntuPro2204`.

2. Implemente los nodos con el siguiente comando.

```
eksctl create nodegroup --config-file=ubuntu.yaml
```

Un ejemplo de salida sería el siguiente.

Se generan varias líneas mientras se crean los nodos. Una de las últimas líneas de salida es la siguiente línea de ejemplo.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Opcional) Implemente una aplicación de muestra para probar los nodos de Ubuntu.
4. Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

   - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
   - Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

   Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Actualizaciones de nodos autoadministrados

Cuando se lanza una nueva AMI optimizada para Amazon EKS, considere la posibilidad de reemplazar los nodos del grupo de nodos autoadministrados con la nueva AMI. Asimismo, si ha actualizado la versión de Kubernetes del clúster de Amazon EKS, actualice los nodos para utilizarlos con la misma versión de Kubernetes.

> **Important**
>
> En este tema, se explican las actualizaciones de los nodos autoadministrados. Si está usando Grupos de nodos administrados, consulte Actualización de un grupo de nodos administrados.

Existen dos formas básicas de actualizar grupos de nodos autoadministrados en los clústeres para utilizar una nueva AMI:

Migración a un nuevo grupo de nodos

Cree un nuevo grupo de nodos y migre los Pods a ese grupo. La migración a un nuevo grupo de nodos es más sencilla que simplemente actualizar el ID de la AMI en una pila de AWS CloudFormation existente. Esto se debe a que el proceso de migración marca al antiguo grupo de nodos como `NoSchedule` y drena los nodos después de que una nueva pila esté lista para aceptar la carga de trabajo del Pod existente.

Actualización de un grupo de nodos autoadministrados existente

Actualice la pila AWS CloudFormation para que un grupo de nodos existente utilice la nueva AMI. Este método no es compatible con los grupos de nodos creados con `eksctl`.

Migración a un nuevo grupo de nodos

En este tema, se describe cómo crear un nuevo grupo de nodos, migrar de forma correcta las aplicaciones existentes al nuevo grupo y eliminar el antiguo grupo de nodos del clúster. Puede migrar a un nuevo grupo de nodos mediante `eksctl` o la AWS Management Console.

eksctl

Para migrar sus aplicaciones a un nuevo grupo de nodos con **`eksctl`**

Para obtener más información sobre el uso de eksctl para la migración, consulte Nodos no administrados en la documentación de `eksctl`.

En este procedimiento, se requiere la versión `0.183.0` o posterior de `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

> **Note**
>
> Este procedimiento solo funciona para los clústeres y grupos de nodos que se crearon con `eksctl`.

1. Recupere el nombre de los grupos de nodos existentes al sustituir *`my-cluster`* por el nombre del clúster.

   ```
   eksctl get nodegroups --cluster=my-cluster
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   CLUSTER      NODEGROUP          CREATED               MIN SIZE      MAX SIZE
    DESIRED CAPACITY     INSTANCE TYPE     IMAGE ID
   default      standard-nodes   2019-05-01T22:26:58Z  1             4             3
                       t3.medium         ami-05a71d034119ffc12
   ```

2. Lance un nuevo grupo de nodos con `eksctl` mediante el siguiente comando. En el comando, sustituya cada *`example value`* por valores propios. El número de versión no puede ser posterior a la versión de Kubernetes del plano de control. Además, no puede ser más de dos versiones secundarias anteriores a la versión de Kubernetes para el plano de control. Recomendamos que utilice la misma versión que el plano de control.

   Se recomienda bloquear el acceso del Pod al IMDS si se cumplen las siguientes condiciones:

   - Tiene previsto asignar roles de IAM a todas sus cuentas de servicio de Kubernetes para que los Pods solo tengan los permisos mínimos que necesitan.
   - Ninguno de los Pods del clúster requiere acceso al servicio de metadatos de la instancia de Amazon EC2 (IMDS) por otros motivos, como la recuperación de la Región de AWS actual.

   Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

   Si desea bloquear el acceso del Pod a IMDS, agregue la opción `--disable-pod-imds` al siguiente comando.

> **Note**
>
> Para obtener más marcadores disponibles y sus descripciones, consulte https://eksctl.io/.

```
eksctl create nodegroup \
  --cluster my-cluster \
  --version 1.30 \
  --name standard-nodes-new \
  --node-type t3.medium \
  --nodes 3 \
  --nodes-min 1 \
  --nodes-max 4 \
  --managed=false
```

3. Cuando se complete el comando anterior, verifique con el siguiente comando que todos los nodos tengan el estado `Ready` (Listo):

```
kubectl get nodes
```

4. Elimine el grupo de nodos original con el siguiente comando. En el comando, sustituya cada *example value* con los nombres del clúster y grupo de nodos:

```
eksctl delete nodegroup --cluster my-cluster --name standard-nodes-old
```

AWS Management Console and AWS CLI

Para migrar sus aplicaciones a un nuevo grupo de nodos con la AWS Management Console y la AWS CLI

1. Lance un nuevo grupo de nodos mediante los pasos que se indican en Lanzar nodos autoadministrados de Amazon Linux.
2. Una vez completada la creación de la pila, selecciónela en la consola y elija Salidas.
3. Anote el valor de NodeInstanceRoles correspondiente al grupo de nodos creado. Lo necesita para agregar los nuevos nodos de Amazon EKS al clúster.

> **Note**
>
> Si ha adjuntado políticas de IAM adicionales al rol de IAM del grupo de nodos anterior, debe adjuntar esas mismas políticas al rol de IAM del nuevo grupo de nodos para mantener esa funcionalidad en el nuevo grupo. Esto se aplica si ha agregado permisos para el Autoescalador del clúster de Kubernetes, por ejemplo.

4. Actualice los grupos de seguridad de ambos grupos de nodos para que puedan comunicarse entre sí. Para obtener más información, consulte Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS.

   a. Anote los ID de grupo de seguridad de ambos grupos de nodos. Esto se muestra como el valor NodeSecurityGroup en los resultados de la pila de AWS CloudFormation.

      Puede utilizar los siguientes comandos de la AWS CLI para obtener los ID de grupo de seguridad de los nombres de pilas. En estos comandos, `oldNodes` es el nombre de pila de AWS CloudFormation de la pila de nodos antigua y `newNodes` es el nombre de la pila a la que migrará. Reemplace cada *`example value`* con valores propios.

      ```
      oldNodes="old_node_CFN_stack_name"
      newNodes="new_node_CFN_stack_name"

      oldSecGroup=$(aws cloudformation describe-stack-resources --stack-name
       $oldNodes \
      --query 'StackResources[?
      ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
      --output text)
      newSecGroup=$(aws cloudformation describe-stack-resources --stack-name
       $newNodes \
      --query 'StackResources[?
      ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
      --output text)
      ```

   b. Agregue reglas de entrada a cada grupo de seguridad de nodos para que acepten tráfico de los otros grupos.

      Los siguientes comandos de la AWS CLI agregan reglas de entrada a cada grupo de seguridad que permiten todo el tráfico en todos los protocolos del otro grupo de

seguridad. Esta configuración permite que los Pods de cada grupo de nodos se comuniquen entre ellos mientras migra la carga de trabajo al nuevo grupo.

```
aws ec2 authorize-security-group-ingress --group-id $oldSecGroup \
--source-group $newSecGroup --protocol -1
aws ec2 authorize-security-group-ingress --group-id $newSecGroup \
--source-group $oldSecGroup --protocol -1
```

5. Edite el mapa de configuración de `aws-auth` para asignar el rol de la instancia del nuevo nodo en RBAC.

```
kubectl edit configmap -n kube-system aws-auth
```

Agregue una nueva entrada `mapRoles` para el nuevo grupo de nodos. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
apiVersion: v1
data:
  mapRoles: |
    - rolearn: ARN of instance role (not instance profile)
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes>
    - rolearn: arn:aws:iam::111122223333:role/nodes-1-16-NodeInstanceRole-
U11V27W93CX5
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
```

Sustituya el fragmento *`ARN of instance role (not instance profile)`* con el valor de NodeInstanceRole que registró en un paso anterior. A continuación, guarde y cierre el archivo para aplicar el configmap actualizado.

6. Examine el estado de los nodos y espere a que los nuevos nodos se unan al clúster y tengan el estado `Ready` (Listo).

```
kubectl get nodes --watch
```

7. (Opcional) Si utiliza el escalador automático del clúster de Kubernetes, escale la implementación a cero (0) réplicas para evitar acciones de escalado en conflicto.

```
kubectl scale deployments/cluster-autoscaler --replicas=0 -n kube-system
```

8. Utilice el siguiente comando para agregar taints en cada uno de los nodos que desee eliminar con `NoSchedule`. Esto es para que los nuevos Pods no se programen ni se vuelvan a programar en los nodos que va a reemplazar. Para obtener más información, consulte Taints y toleraciones en la documentación de Kubernetes.

```
kubectl taint nodes node_name key=value:NoSchedule
```

Si va a actualizar los nodos a una nueva versión de Kubernetes, puede identificar y agregar taints en todos los nodos de una determinada versión de Kubernetes (en este caso `1.28`) con el siguiente fragmento de código. El número de versión no puede ser posterior a la versión de Kubernetes del plano de control. Además, no puede ser más de dos versiones secundarias anteriores a la versión de Kubernetes del plano de control. Recomendamos que utilice la misma versión que el plano de control.

```
K8S_VERSION=1.28
nodes=$(kubectl get nodes -o jsonpath="{.items[?
(@.status.nodeInfo.kubeletVersion==\"v$K8S_VERSION\")].metadata.name}")
for node in ${nodes[@]}
do
    echo "Tainting $node"
    kubectl taint nodes $node key=value:NoSchedule
done
```

9. Identifique el proveedor de DNS del clúster.

```
kubectl get deployments -l k8s-app=kube-dns -n kube-system
```

Un ejemplo de salida sería el siguiente. Este clúster utiliza CoreDNS para la resolución DNS, pero el clúster puede devolver `kube-dns` en su lugar:

```
NAME      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
coredns   1         1         1            1           31m
```

10. Si su implementación actual está ejecutando menos de dos réplicas, escale la implementación a dos réplicas. Cambie *coredns* por **kubedns** si el resultado del comando anterior ha devuelto ese valor.

```
kubectl scale deployments/coredns --replicas=2 -n kube-system
```

11. Vacíe cada uno de los nodos que desea eliminar de su clúster con el siguiente comando:

```
kubectl drain node_name --ignore-daemonsets --delete-local-data
```

Si va a actualizar los nodos a una nueva versión de Kubernetes, identifique y vacíe todos los nodos de una determinada versión de Kubernetes (en este caso *1.28*) con el siguiente fragmento de código.

```
K8S_VERSION=1.28
nodes=$(kubectl get nodes -o jsonpath="{.items[?
(@.status.nodeInfo.kubeletVersion==\"v$K8S_VERSION\")].metadata.name}")
for node in ${nodes[@]}
do
    echo "Draining $node"
    kubectl drain $node --ignore-daemonsets --delete-local-data
done
```

12. Una vez que haya terminado de vaciar los nodos antiguos, revoque las reglas de entrada del grupo de seguridad que autorizó anteriormente. A continuación, elimine la pila de AWS CloudFormation para terminar las instancias.

> **Note**
>
> Si ha adjuntado políticas de IAM adicionales al rol de IAM del grupo de nodos anterior, como agregar permisos para el escalador automático del clúster de Kubernetes, desconecte esas políticas adicionales del rol antes de eliminar la pila de AWS CloudFormation.

a. Revoque las reglas de entrada que creó anteriormente para los grupos de seguridad de nodos. En estos comandos, `oldNodes` es el nombre de pila de AWS CloudFormation de la pila de nodos antigua y `newNodes` es el nombre de la pila a la que migrará.

```
oldNodes="old_node_CFN_stack_name"
newNodes="new_node_CFN_stack_name"

oldSecGroup=$(aws cloudformation describe-stack-resources --stack-name
 $oldNodes \
--query 'StackResources[?
ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
--output text)
newSecGroup=$(aws cloudformation describe-stack-resources --stack-name
 $newNodes \
--query 'StackResources[?
ResourceType==`AWS::EC2::SecurityGroup`].PhysicalResourceId' \
--output text)
aws ec2 revoke-security-group-ingress --group-id $oldSecGroup \
--source-group $newSecGroup --protocol -1
aws ec2 revoke-security-group-ingress --group-id $newSecGroup \
--source-group $oldSecGroup --protocol -1
```

b. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.

c. Seleccione la pila de nodos antigua.

d. Elija Eliminar.

e. En el cuadro de diálogo de confirmación Eliminar pila, elija Eliminar pila.

13. Edite el mapa de configuración de `aws-auth` para eliminar el rol de la instancia del nodo anterior de RBAC.

```
kubectl edit configmap -n kube-system aws-auth
```

Elimine la entrada `mapRoles` del grupo de nodos antiguo. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
apiVersion: v1
data:
  mapRoles: |
    - rolearn: arn:aws:iam::111122223333:role/nodes-1-16-NodeInstanceRole-
W70725MZQFF8
      username: system:node:{{EC2PrivateDNSName}}
      groups:
```

```
        - system:bootstrappers
        - system:nodes
    - rolearn: arn:aws:iam::111122223333:role/nodes-1-15-NodeInstanceRole-
U11V27W93CX5
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes>
```

Guarde y cierre el archivo para aplicar el mapa de configuración actualizado.

14. (Opcional) Si utiliza el Autoescalador del clúster de Kubernetes, vuelva a escalar la implementación a una réplica.

> **Note**
>
> También debe etiquetar el nuevo grupo de Auto Scaling de forma correcta (por ejemplo, `k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/`*`my-cluster`*) y actualizar el comando de implementación del escalador automático del clúster para que señale el grupo de Auto Scaling recién etiquetado. Para obtener más información, consulte Escalador automático de clústeres en AWS.

```
kubectl scale deployments/cluster-autoscaler --replicas=1 -n kube-system
```

15. (Opcional) Verifique que utiliza la última versión del complemento CNI de Amazon VPC para Kubernetes. Es posible que tenga que actualizar la versión de CNI para utilizar los tipos de instancia admitidos más recientes. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.
16. Si el clúster utiliza `kube-dns` para la resolución DNS (consulte el paso anterior), escale la implementación de `kube-dns` a una réplica.

```
kubectl scale deployments/kube-dns --replicas=1 -n kube-system
```

Actualización de un grupo de nodos autoadministrados existente

En este tema, se describe cómo puede actualizar una pila existente de nodos autoadministrados de AWS CloudFormation con una nueva AMI. Puede utilizar este procedimiento para actualizar los nodos a una nueva versión de Kubernetes después de la actualización de un clúster. De lo contrario, puede actualizar a la última AMI optimizada de Amazon EKS para una versión existente de Kubernetes.

> **Important**
>
> En este tema, se explican las actualizaciones de los nodos autoadministrados. Para obtener más información sobre del uso de Grupos de nodos administrados, consulte Actualización de un grupo de nodos administrados.

La última plantilla de AWS CloudFormation de nodos de Amazon EKS predeterminada está configurada para lanzar una instancia con la nueva AMI en el clúster antes de eliminar una antigua, una por vez. Esta configuración garantiza que siempre tenga el número deseado de instancias activas del grupo de Auto Scaling en el clúster durante la actualización progresiva.

> **Note**
>
> Este método no es compatible con los grupos de nodos creados con `eksctl`. Si ha creado su clúster o un grupo de nodos con `eksctl`, consulte Migración a un nuevo grupo de nodos.

Para actualizar un grupo de nodos existente

1. Determine el proveedor de DNS para el clúster.

   ```
   kubectl get deployments -l k8s-app=kube-dns -n kube-system
   ```

   Un ejemplo de salida sería el siguiente. Este clúster utiliza CoreDNS para la resolución DNS, pero el clúster puede devolver `kube-dns` en su lugar. El resultado puede tener un aspecto diferente según la versión de `kubectl` que utilice.

   ```
   NAME      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
   coredns   1         1         1            1           31m
   ```

2. Si su implementación actual está ejecutando menos de dos réplicas, escale la implementación a dos réplicas. Cambie *coredns* por **kube-dns** si el resultado del comando anterior ha devuelto ese valor.

```
kubectl scale deployments/coredns --replicas=2 -n kube-system
```

3. (Opcional) Si utiliza el escalador automático del clúster de Kubernetes, escale la implementación a cero (0) réplicas para evitar acciones de escalado en conflicto.

```
kubectl scale deployments/cluster-autoscaler --replicas=0 -n kube-system
```

4. Determine el tipo de instancia y el número de instancias deseado del grupo de nodos actual. Ingrese estos valores más tarde cuando actualice la plantilla de AWS CloudFormation del grupo.
   a. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
   b. En el panel de navegación izquierdo, elija Configuraciones de lanzamiento (Configuraciones de lanzamiento) y anote el tipo de instancia de la configuración de lanzamiento de nodos existente.
   c. En el panel de navegación izquierdo, elija Auto Scaling Groups (Grupos de Auto Scaling) y anote el recuento de instancias deseado para el grupo de Auto Scaling de nodos existente.
5. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
6. Seleccione la pila del grupo de nodos y, a continuación, seleccione Update (Actualizar).
7. Seleccione Replace current template (Reemplazar plantilla actual) y seleccione Amazon S3 URL.
8. Para Amazon S3 URL (URL de Amazon S3), pegue la siguiente URL en el área de texto a fin de asegurarse de que utiliza la versión más reciente de la plantilla de AWS CloudFormation de nodos. A continuación, elija Siguiente:

```
https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2022-12-23/amazon-eks-nodegroup.yaml
```

9. En la página Especificar detalles, rellene los parámetros siguientes y seleccione Siguiente:
   - NodeAutoScalingGroupDesiredCapacity: ingrese el recuento de instancias deseado que registró en un paso anterior. O bien, ingrese el nuevo número deseado de nodos para escalar cuando se actualice la pila.

- NodeAutoScalingGroupMaxSize: ingrese el número máximo de nodos al que pueda llegar el grupo de Auto Scaling de nodos. Este valor debe ser al menos un nodo más que la capacidad deseada. Es para que pueda realizar una actualización continua de los nodos sin que se reduzca el número durante la actualización.
- NodeInstanceType: elija el tipo de instancia que registró en un paso anterior. Por otra parte, puede elegir un tipo de instancia diferente para los nodos. Antes de elegir un tipo de instancia diferente, revise Elección de un tipo de instancia de Amazon EC2. Cada tipo de instancia de Amazon EC2 admite un número máximo de interfaces de redes elásticas (interfaz de red) y cada interfaz de red admite un número máximo de direcciones IP. Dado que a cada nodo de trabajo y Pod se les asigna su propia dirección IP, es importante elegir un tipo de instancia que admita el número máximo de Pods que desea ejecutar en cada nodo de Amazon EC2. Para obtener una lista del número de interfaces de red y direcciones IP admitidas por los tipos de instancias, consulte Direcciones IP por interfaz de red por tipo de instancia. Por ejemplo, el tipo de instancia `m5.large` admite un máximo de 30 direcciones IP para el nodo de trabajo y los Pods.

  > **Note**
  >
  > Los tipos de instancias compatibles con la versión más reciente de Amazon VPC CNI plugin for Kubernetes se muestran en vpc_ip_resource_limit.go en GitHub. Es posible que tenga que actualizar la versión de Amazon VPC CNI plugin for Kubernetes para utilizar los tipos de instancia admitidos más recientes. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.

  > **Important**
  >
  > Algunos tipos de instancias podrían no estar disponibles en todas las Regiones de AWS.

- NodeImageIdSSMParam: el parámetro de Amazon EC2 Systems Manager del ID de AMI al que desee actualizar. El siguiente valor utiliza la última AMI optimizada para Amazon EKS para la versión `1.30` de Kubernetes.

  ```
  /aws/service/eks/optimized-ami/1.30/amazon-linux-2/recommended/image_id
  ```

Puede reemplazar *1.30* con una versión de Kubernetes admitida que sea la misma. O bien, debería ser hasta una versión anterior a la versión de Kubernetes que se ejecuta en el plano de control. Le recomendamos que los nodos tengan la misma versión que el plano de control. También se puede sustituir *amazon-linux-2* por un tipo de AMI diferente. Para obtener más información, consulte Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS.

> **Note**
>
> El uso del parámetro de Amazon EC2 Systems Manager lo habilita a actualizar los nodos en el futuro sin tener que buscar y especificar un ID de AMI. Si la pila de AWS CloudFormation utiliza este valor, cualquier actualización de la pila lanzará siempre la última AMI optimizada para Amazon EKS recomendada en función de la versión de Kubernetes especificada. Este es el caso incluso si no cambia ningún valor en la plantilla.

- NodeImageId: para utilizar su propia AMI personalizada, introduzca el ID de la AMI que va a utilizar.

> **Important**
>
> Este valor anula cualquier valor especificado para NodeImageIdSSMParam. Si desea utilizar el valor NodeImageIdSSMParam asegúrese de que el valor de NodeImageId esté vacío.

- DisableIMDSv1: cada nodo admite de forma predeterminada la versión 1 (IMDSv1) e IMDSv2 del servicio de metadatos de la instancia. Sin embargo, puede desactivar IMDSv1. Seleccione true (verdadero) si no desea que ningún nodo o ningún Pods programado en el grupo de nodos utilice IMDSv1. Para obtener más información, consulte Configuración del servicio de metadatos de instancia. Si ha implementado roles de IAM para cuentas de servicio, asigne los permisos necesarios de forma directa a todos los Pods que requieren acceso a servicios de AWS. De esta manera, ningún Pods en el clúster requiere el acceso a IMDS por otros motivos, como la recuperación de los datos actuales de la Región de AWS. A continuación, también puede desactivar el acceso a IMDSv2 para los Pods que no utilizan redes alojadas. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

10. (Opcional) En la página Options (Opciones), marque los recursos de la pila. Elija Siguiente.

11. En la página Review (Revisar), revise la información, confirme la advertencia de que la pila puede crear recursos de IAM y elija Update stack (Actualizar pila).

> **Note**
>
> La actualización de cada nodo del clúster tarda varios minutos. Espere a que se complete la actualización de todos los nodos antes de realizar los siguientes pasos.

12. Si su proveedor DNS del clúster es `kube-dns`, escale la implementación de `kube-dns` a una réplica.

```
kubectl scale deployments/kube-dns --replicas=1 -n kube-system
```

13. (Opcional) Si utiliza el Autoescalador de clúster de Kubernetes, vuelva a escalar la implementación a la cantidad de réplicas deseada.

```
kubectl scale deployments/cluster-autoscaler --replicas=1 -n kube-system
```

14. (Opcional) Verifique que utiliza la última versión de Amazon VPC CNI plugin for Kubernetes. Es posible que tenga que actualizar la versión de Amazon VPC CNI plugin for Kubernetes para utilizar los tipos de instancia admitidos más recientes. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.

AWS Fargate

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

En este tema se explica cómo utilizar Amazon EKS para ejecutar Pods de Kubernetes en AWS Fargate. Fargate es una tecnología que proporciona capacidad informática bajo demanda correctamente dimensionada para contenedores. Con Fargate ya no tendrá que aprovisionar, configurar ni escalar grupos de máquinas virtuales por su cuenta para ejecutar los contenedores. Tampoco tendrá que elegir tipos de servidores, decidir cuándo escalar los grupos de nodos u optimizar el empaquetado de clústeres.

Puede controlar qué Pods se comienzan en Fargate y cómo se ejecutan con perfiles de Fargate. Los perfiles de Fargate se definen como parte de su clúster de Amazon EKS. Amazon EKS integra Kubernetes con Fargate mediante la utilización de controladores creados por AWS con el modelo ascendente y extensible proporcionado por Kubernetes. Estos controladores funcionan como parte del plano de control de Kubernetes administrado por Amazon EKS y son responsables de programar los Pods Kubernetes nativos en Fargate. Los controladores de Fargate incluyen un nuevo programador que se ejecuta junto con el programador predeterminado de Kubernetes, además de varios controladores de admisión de mutación y validación. Cuando comienza un Pod que cumple los criterios para ejecutarse en Fargate, los controladores de Fargate que se ejecutan en el clúster reconocen, actualizan y programan el Pod en Fargate.

En este tema se describen los diferentes componentes de los Pods que se ejecutan en Fargate y se ofrecen consideraciones especiales sobre cómo utilizar Fargate con Amazon EKS.

Consideraciones sobre AWS Fargate

Aquí se incluyen algunos aspectos que debe tener en cuenta sobre la utilización de Fargate en Amazon EKS.

- Cada Pod que se ejecuta en Fargate tiene su propio límite de aislamiento. No comparten el kernel subyacente, los recursos de CPU, los recursos de memoria o la interfaz de red elástica con otro Pod.
- Los Network Load Balancers y los Application Load Balancers (ALB) solo se pueden utilizar con Fargate con destinos IP. Para obtener más información, consulte Crear un equilibrador de carga de red y Equilibrio de carga de aplicaciones en Amazon EKS.
- Los servicios expuestos de Fargate solo se ejecutan en modo IP de tipo de destino y no en modo IP de nodo. La forma recomendada de verificar la conectividad de un servicio que se ejecuta en un nodo administrado y un servicio que se ejecuta en Fargate es conectarse a través del nombre del servicio.
- Los pods deben coincidir con un perfil de Fargate en el momento de su programación para ejecutarse en Fargate. Los pods que no coinciden con un perfil de Fargate podrían quedarse atascados como `Pending`. Si existe un perfil de Fargate coincidente, puede eliminar los Pods pendientes que haya creado para reprogramarlos en Fargate.
- No se admiten DaemonSets en Fargate. Si su aplicación requiere un daemon, reconfigure ese daemon para que se ejecute como un contenedor asociado en sus Pods.
- No se admiten contenedores con privilegios en Fargate.

- Los pods que se ejecutan en Fargate no pueden especificar `HostPort` ni `HostNetwork` en el manifiesto del Pod.
- El límite flexible predeterminado de `nofile` y `nproc` es 1024 y el límite invariable es 65 535 para los Pods de Fargate.
- Las GPU no están disponibles actualmente en Fargate.
- Los pods que se ejecutan en Fargate solo se admiten en subredes privadas (con acceso de una puerta de enlace NAT a servicios de AWS, pero sin una ruta directa a una puerta de enlace de Internet), por lo que la VPC del clúster debe tener subredes privadas disponibles. Para los clústeres sin acceso a Internet saliente, consulte Requisitos del clúster privado.
- Puede utilizar el Escalador automático vertical de pods para medir correctamente inicialmente el tamaño de la CPU y la memoria de los Pods de Fargate y, a continuación, utilizar el Escalador automático de pods horizontales para ajustar la escala de esos Pods. Si desea que el escalador automático vertical de pods vuelva a implementar automáticamente los Pods en Fargate con combinaciones de CPU y memoria más grandes, configure el modo del escalador automático vertical de pods `Auto` o `Recreate` para garantizar la funcionalidad correcta. Para obtener más información, consulte el documento Escalador automático vertical de pods en GitHub.
- La resolución de DNS y los nombres de host DNS deben estar habilitados en la VPC. Para obtener más información, consulte Ver y actualizar el soporte de DNS para su VPC.
- Fargate de Amazon EKS agrega defensa en profundidad para las aplicaciones de Kubernetes aislando cada pod dentro de una máquina virtual (VM). Este límite de VM impide el acceso a los recursos basados en host utilizados por otros pods en caso de escape de contenedor, que es un método común para atacar aplicaciones en contenedores y obtener acceso a recursos fuera del contenedor.

  El uso de Amazon EKS no modifica sus responsabilidades en virtud del modelo de responsabilidad compartida. Debe evaluar detenidamente la configuración de los controles de seguridad y gobernanza del clúster. La forma más segura de aislar una aplicación es ejecutarla siempre en un clúster independiente.
- Los perfiles de Fargate admiten la especificación de subredes de bloques de CIDR secundarios de VPC. Puede que desee especificar un bloque de CIDR secundario. Esto es debido a que hay un número limitado de direcciones IP disponibles en una subred. Como resultado, hay también un número limitado de Pods que se pueden crear en el clúster. Si usa subredes diferentes para los Pods, puede aumentar el número de direcciones IP disponibles. Para obtener más información, consulte Adición de bloques de CIDR `IPv4` a una VPC.

- El servicio de metadatos de instancia (IMDS) de Amazon EC2 no está disponible para Pods implementados en nodos de Fargate. Si tiene Pods implementados en Fargate que necesitan credenciales de IAM, asígnelos a sus Pods con Roles de IAM para cuentas de servicio. Si los Pods necesitan acceso a otra información disponible a través de IMDS, debe realizar una codificación rígida de esta información en la especificación del Pod. Esto incluye la Región de AWS o zona de disponibilidad en la que se implementa un Pod.
- No se pueden implementar Pods de Fargate en las zonas locales AWS Outposts, AWS Wavelength o AWS.
- Amazon EKS debe aplicar parches periódicamente de Fargate Pods para mantenerlos seguros. Intentamos las actualizaciones de forma que disminuya el impacto, pero hay ocasiones en que los Pods deben eliminarse si no se expulsan correctamente. Hay algunas acciones que puede emprender para minimizar las interrupciones. Para obtener más información, consulte Parches de SO de Fargate.
- El complemento CNI de Amazon VPC para Amazon EKS se encuentra instalado en los nodos de Fargate. No puede usar Complementos CNI compatibles alternativos con los nodos de Fargate.
- Un Pod que se ejecuta en Fargate monta automáticamente un sistema de archivos de Amazon EFS. No se puede utilizar el aprovisionamiento dinámico de volúmenes persistentes con nodos de Fargate, pero se puede utilizar el aprovisionamiento estático.
- No puede montar volúmenes de Amazon EBS en Fargate Pods.
- Puede ejecutar el controlador CSI de Amazon EBS en nodos de Fargate, pero el nodo CSI de Amazon EBS DaemonSet solo se puede ejecutar en instancias de Amazon EC2.
- Después de marcar un JobKubernetes como `Completed` o`Failed`, los Pods que Job crea normalmente siguen existiendo. Este comportamiento le permite ver sus registros y resultados, pero con Fargate incurrirá en costos si no limpia después el Job.

  Para eliminar automáticamente los Pods relacionados después de que Job se complete o falle, puede especificar un período de tiempo mediante el controlador de tiempo de vida (TTL). En el siguiente ejemplo, se muestra la especificación `.spec.ttlSecondsAfterFinished` en el manifiesto de Job.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: busybox
spec:
  template:
    spec:
```

```
    containers:
    - name: busybox
      image: busybox
      command: ["/bin/sh", "-c", "sleep 10"]
    restartPolicy: Never
ttlSecondsAfterFinished: 60 # <-- TTL controller
```

Introducción a la utilización de AWS Fargate con Amazon EKS

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

En este tema, se explica cómo comenzar a ejecutar Pods en AWS Fargate con su clúster de Amazon EKS.

Si restringe el acceso al punto de conexión público del clúster mediante bloques de CIDR, recomendamos habilitar también el acceso al punto de conexión privado. De este modo, los Pods de Fargate pueden comunicarse con el clúster. Si el punto de conexión privado no está habilitado, los bloques de CIDR que especifique para el acceso público deben incluir los orígenes de salida de su VPC. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS.

Requisito previo

Un clúster existente. Si no dispone de un clúster de Amazon EKS, consulte Introducción a Amazon EKS.

Asegúrese de que los nodos existentes se puedan comunicar con los Pods de Fargate

Si está trabajando con un nuevo clúster sin nodos o con un clúster con solo grupos de nodos administrados, puede ir directamente a Crear un rol de ejecución de Pod de Fargate.

Supongamos que está trabajando con un clúster existente que ya tiene nodos asociados. Asegúrese de que los Pods de estos nodos puedan comunicarse libremente con los Pods que se ejecutan en Fargate. Los Pods que se ejecutan en Fargate se configuran automáticamente para utilizar el grupo de seguridad del clúster al que están asociados. Asegúrese de que los nodos existentes en el clúster

puedan enviar tráfico hacia el grupo de seguridad del clúster y recibirlo desde este. Los Grupos de nodos administrados se configuran de manera automática para utilizar también el grupo de seguridad del clúster, por lo que no es necesario modificarlos ni verificar si admiten esta función.

Para los grupos de nodos existentes que se crearon con `eksctl` o con las plantillas de AWS CloudFormation administrado de Amazon EKS, puede agregar manualmente el grupo de seguridad del clúster a los nodos. O bien, puede modificar la plantilla de lanzamiento del grupo de Auto Scaling para el grupo de nodos a fin de adjuntar el grupo de seguridad del clúster a las instancias. Para obtener más información, consulte Cambio de los grupos de seguridad de una instancia en la Guía del usuario de Amazon VPC.

Puede buscar un grupo de seguridad para su clúster en la AWS Management Console, en la sección Networking (Redes) del clúster. O puede hacerlo con el siguiente comando de la AWS CLI. Cuando utilice este comando, sustituya *my-cluster* por el nombre del clúster.

```
aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.clusterSecurityGroupId
```

Crear un rol de ejecución de Pod de Fargate

Cuando su clúster crea Pods en AWS Fargate, los componentes que se ejecutan en la infraestructura de Fargate deben hacer llamadas a las API de AWS en su nombre. El rol de ejecución de Pod de Amazon EKS proporciona los permisos de IAM para esta tarea. Para crear un rol de ejecución AWS Fargate Pod, consulte Rol de IAM de ejecución de Pod de Amazon EKS.

> **Note**
>
> Si creó el clúster con `eksctl` mediante la opción `--fargate`, el clúster ya tiene un rol de ejecución de Pod que puede encontrar en la consola de IAM con el patrón `eksctl-`*`my-cluster`*`-FargatePodExecutionRole-`*`ABCDEFGHIJKL`*. Del mismo modo, si utiliza `eksctl` para crear sus perfiles de Fargate, `eksctl` crea su rol de ejecución de Pod si aún no se ha creado uno.

Crear un perfil de Fargate para el clúster

Para poder programar los Pods que se ejecuten en Fargate en su clúster, debe definir un perfil de Fargate que especifique qué Pods deben utilizar Fargate cuando se lancen. Para obtener más información, consulte Perfil de AWS Fargate.

> **Note**
>
> Si creó el clúster con `eksctl` mediante la opción `--fargate`, ya se habrá creado un perfil de Fargate para el clúster con selectores para todos los Pods de los espacios de nombres `kube-system` y `default`. Utilice el siguiente procedimiento para crear perfiles de Fargate para cualquier otro espacio de nombres que desee utilizar con Fargate.

Puede crear un perfil de Fargate con `eksctl` o la AWS Management Console.

eksctl

En este procedimiento, se requiere la versión `0.183.0` o posterior de `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

Cómo crear un perfil de Fargate con **`eksctl`**

Cree el perfil de Fargate con el siguiente comando de `eksctl` y reemplace cada *`example value`* con valores propios. Debe especificar un espacio de nombres. Sin embargo, la opción `--labels` no es obligatoria.

```
eksctl create fargateprofile \
    --cluster my-cluster \
    --name my-fargate-profile \
    --namespace my-kubernetes-namespace \
    --labels key=value
```

Puede usar ciertos comodines para las etiquetas *`my-kubernetes-namespace`* y *`key=value`*. Para obtener más información, consulte Comodines de perfil de Fargate.

AWS Management Console

Cómo crear un perfil de Fargate para un clúster con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el clúster para el que desea crear un perfil de Fargate.

3. Elija la pestaña Computación.
4. En Perfiles de Fargate, elija Agregar perfil de Fargate.
5. En la página Configurar perfil de Fargate, haga lo siguiente:

   a. En Name (Nombre), ingrese un nombre único para el perfil de Fargate. El nombre debe ser único.

   b. En Rol de ejecución de pod, elija el rol de ejecución de Pod que se va a utilizar con el perfil de Fargate. Solo se muestran los roles de IAM con la entidad principal del servicio de `eks-fargate-pods.amazonaws.com`. Si no ve ningún rol, debe crear uno. Para obtener más información, consulte Rol de IAM de ejecución de Pod de Amazon EKS.

   c. Modifique las Subredes seleccionadas según sea necesario.

   > **Note**
   >
   > Solo las subredes privadas son compatibles con los Pods que se ejecutan en Fargate.

   d. En Etiquetas, puede etiquetar su perfil de Fargate si lo desea. Estas etiquetas no se propagan a otros recursos asociados con el perfil, como los Pods.

   e. Elija Siguiente.

6. En la página Configurar la selección de Pod, haga lo siguiente:

   a. En Namespace (Espacio de nombres), especifique un espacio de nombres que coincida con los Pods.

   - Puede usar espacios de nombres específicos para que coincidan, como **`kube-system`** o **`default`**.
   - Puede usar ciertos comodines (por ejemplo, **`prod-*`**) para que coincidan con varios espacios de nombres (por ejemplo, `prod-deployment` y `prod-test`). Para obtener más información, consulte Comodines de perfil de Fargate.

   b. (Opcional) Agregue etiquetas de Kubernetes al selector. Agréguelos específicamente al selector con el que deben coincidir los Pods del espacio de nombres especificado.

   - Puede agregar la etiqueta **`infrastructure: fargate`** al selector para que solo los Pods del espacio de nombres especificado que también tengan la etiqueta `infrastructure: fargate` de Kubernetes coincidan con el selector.

- Puede usar ciertos comodines (por ejemplo, **`key?: value?`**) para que coincidan con varios espacios de nombres (por ejemplo, `keya: valuea` y `keyb: valueb`). Para obtener más información, consulte Comodines de perfil de Fargate.

c. Elija Siguiente.

7. En la página Revisar y crear, revise la información de su perfil de Fargate y elija Crear.

Actualice CoreDNS

De forma predeterminada, CoreDNS está configurado para ejecutarse en la infraestructura de Amazon EC2 en clústeres de Amazon EKS. Si desea solo ejecutar los Pods de Fargate en el clúster, complete los pasos que se describen a continuación.

> **Note**
>
> Si ha creado el clúster con `eksctl` mediante la opción `--fargate`, entonces puede ir directamente a Siguientes pasos.

1. Cree un perfil de Fargate para CoreDNS con el siguiente comando. Reemplace *`my-cluster`* con su nombre de clúster, *`111122223333`* con su ID de cuenta, *`AmazonEKSFargatePodExecutionRole`* con el nombre de su rol de ejecución de Pod, y *`0000000000000001`*, *`0000000000000002`* y *`0000000000000003`* con los ID de las subredes privadas. Si no tiene un rol de ejecución de Pod, debe crear uno primero.

   > **Important**
   >
   > El ARN de rol no puede incluir una ruta de acceso que no sea `/`. Por ejemplo, si el nombre de su rol es `development/apps/my-role`, tendrá que cambiarlo a `my-role` cuando especifique el ARN del rol. El formato del ARN del rol debe ser `arn:aws:iam::111122223333:role/role-name`.

```
aws eks create-fargate-profile \
    --fargate-profile-name coredns \
    --cluster-name my-cluster \
    --pod-execution-role-arn
 arn:aws:iam::111122223333:role/AmazonEKSFargatePodExecutionRole \
```

```
    --selectors namespace=kube-system,labels={k8s-app=kube-dns} \
    --subnets subnet-0000000000000001 subnet-0000000000000002
 subnet-0000000000000003
```

2. Ejecute el siguiente comando para eliminar la anotación `eks.amazonaws.com/compute-type : ec2` de los Pods de CoreDNS.

```
kubectl patch deployment coredns \
    -n kube-system \
    --type json \
    -p='[{"op": "remove", "path": "/spec/template/metadata/annotations/
eks.amazonaws.com~1compute-type"}]'
```

Siguientes pasos

- Puede comenzar a migrar las aplicaciones existentes para que se ejecuten en Fargate con el siguiente flujo de trabajo.
  1. Cree un perfil de Fargate que coincida con el espacio de nombres de Kubernetes y las etiquetas de Kubernetes de su aplicación.
  2. Elimine y vuelva a crear los Pods existentes para que se programen en Fargate. Por ejemplo, el siguiente comando desencadena un despliegue de la implementación `coredns`. Puede modificar el espacio de nombre y el tipo de implementación para actualizar sus Pods específicos.

```
kubectl rollout restart -n kube-system deployment coredns
```

- Implemente Equilibrio de carga de aplicaciones en Amazon EKS para permitir que los objetos de entrada de los Pods se ejecuten en Fargate.
- Puede utilizar el Escalador automático vertical de pods para medir correctamente inicialmente el tamaño de la CPU y la memoria de los Pods de Fargate y, a continuación, utilizar el Escalador automático de pods horizontales para ajustar la escala de esos Pods. Si desea que el escalador automático vertical de pods vuelva a implementar automáticamente los Pods en Fargate con combinaciones de CPU y memoria mayores, configure el modo del escalador automático vertical de pods en `Auto` o `Recreate`. Esto es para garantizar una funcionalidad correcta. Para obtener más información, consulte el documento Escalador automático vertical de pods en GitHub.
- Puede configurar el recopilador AWS Distro for OpenTelemetry (ADOT) para el monitoreo de aplicaciones siguiendo estas instrucciones.

Perfil de AWS Fargate

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

Para poder programar Pods en Fargate en el clúster, debe definir al menos un perfil de Fargate que especifique qué Pods utiliza Fargate cuando se lanzan.

Como administrador, puede utilizar un perfil de Fargate para declarar qué Pods se ejecutan en Fargate. Puede hacerlo a través de los selectores del perfil. Puede agregar hasta cinco selectores a cada perfil. Cada selector debe contener un espacio de nombres. El selector también puede incluir etiquetas. El campo de etiqueta consta de varios pares de clave-valor opcionales. Los pods que coinciden con un selector se programan en Fargate. Los pods se comparan mediante un espacio de nombres y las etiquetas que se especifican en el selector. Si se define un selector de espacio de nombres sin etiquetas, Amazon EKS intenta programar todos los Pods que se ejecutan en ese espacio de nombres en Fargate mediante el perfil. Si un Pod programado coincide con alguno de los selectores del perfil de Fargate, ese Pod se programa en Fargate.

Si un Pod coincide con varios perfiles de Fargate, puede especificar qué perfil utiliza un Pod al agregar la siguiente etiqueta de Kubernetes a la especificación del Pod: `eks.amazonaws.com/fargate-profile: my-fargate-profile`. El Pod debe coincidir con un selector en ese perfil para ser programado en Fargate. Las reglas de afinidad y antiafinidad de Kubernetes no se aplican y no son necesarias con los Pods de Amazon EKS Fargate.

Cuando se crea un perfil de Fargate, se debe especificar un rol de ejecución del Pod. Este rol de ejecución es para los componentes de Amazon EKS que se ejecutan en la infraestructura de Fargate mediante el perfil. Se agrega al control de acceso basado en roles (RBAC) de Kubernetes del clúster para la autorización. De este modo, el `kubelet` que se ejecuta en la infraestructura de Fargate puede registrarse en su clúster de Amazon EKS y aparecer en su clúster como un nodo. El rol de ejecución de Pod también proporciona permisos de IAM a la infraestructura de Fargate para permitir el acceso de lectura a los repositorios de imágenes de Amazon ECR. Para obtener más información, consulte Rol de IAM de ejecución de Pod de Amazon EKS.

Los perfiles de Fargate no se pueden cambiar. Sin embargo, puede crear un nuevo perfil actualizado para reemplazar un perfil existente y, a continuación, eliminar el original.

> **Note**
>
> Los Pods que se están ejecutando con un perfil de Fargate se detienen y pasan a un estado pendiente cuando se elimina el perfil.

Si el estado de alguno de los perfiles de Fargate de un clúster es `DELETING`, hay que esperar a que se borre el perfil de Fargate antes de crear otros perfiles en ese clúster.

Amazon EKS y Fargate distribuyen los Pods en cada una de las subredes definidas en el perfil de Fargate. Sin embargo, es posible que acabe con una propagación desigual. Si debe tener una propagación uniforme, utilice dos perfiles de Fargate. La propagación uniforme es importante en los escenarios en los que se desea desplegar dos réplicas y no se desea ningún tiempo de inactividad. Se recomienda que cada perfil tenga solo una subred.

Componentes de un perfil de Fargate

Un perfil de Fargate consta de los siguientes componentes.

Rol de ejecución de pods

: Cuando el clúster crea Pods en AWS Fargate, el `kubelet` que se ejecuta en la infraestructura de Fargate debe hacer llamadas a las API de AWS en su nombre. Por ejemplo, necesita hacer llamadas para extraer imágenes del contenedor de Amazon ECR. El rol de ejecución de Pod de Amazon EKS proporciona los permisos de IAM para esta tarea.

  Al crear un perfil de Fargate, debe especificar un rol de ejecución de Pod para utilizarlo con los Pods. Este rol se agrega al control de acceso basado en roles (RBAC) de Kubernetes del clúster para su autorización. De este modo, el `kubelet` que se está ejecutando en la infraestructura de Fargate puede registrarse en el clúster de Amazon EKS y aparecer en el clúster como un nodo. Para obtener más información, consulte Rol de IAM de ejecución de Pod de Amazon EKS.

Subredes

: Los identificadores de las subredes en las que se lanzarán los Pods que utilizan este perfil. En este momento, a los Pods que se ejecutan en Fargate no se les asignan direcciones IP públicas. Por lo tanto, para este parámetro solo se aceptan subredes privadas que no tengan una ruta directa a una puerta de enlace de Internet.

Selectores

Los selectores que deben coincidir para que los Pods utilicen este perfil de Fargate. Puede especificar hasta cinco selectores en un perfil Fargate. Los selectores tienen los siguientes componentes:

- Espacio de nombres: debe especificar un espacio de nombres para un selector. El selector solo hace coincidir los Pods que se crean en este espacio de nombres. Sin embargo, puede crear varios selectores para orientar varios espacios de nombres.
- Etiquetas: puede especificar etiquetas de Kubernetes que coincidan con el selector, si lo desea. El selector solo coincide con los Pods que tienen todas las etiquetas especificadas en el selector.

Comodines de perfil de Fargate

Además de los caracteres permitidos por Kubernetes, se permite utilizar ***** y **?** en los criterios del selector para los espacios de nombres, las claves de las etiquetas y los valores de las etiquetas:

- ***** representa ninguno, uno o varios caracteres. Por ejemplo, **`prod*`** puede representar `prod` y `prod-metrics`.
- **?** representa un solo carácter (por ejemplo, **`value?`** puede representar `valuea`). Sin embargo, no puede representar `value` y `value-a`, porque ? solo puede representar exactamente un carácter.

Estos caracteres comodín se pueden usar en cualquier posición y en combinación (por ejemplo, **`prod*`**, **`*dev`** y **`frontend*?`**). No se admiten otros comodines ni formas de coincidencia de patrones, como las expresiones regulares.

Si hay varios perfiles que coinciden con el espacio de nombres y las etiquetas en la especificación del Pod, Fargate elige el perfil según la clasificación alfanumérica por nombre de perfil. Por ejemplo, si tanto el perfil A (con el nombre `beta-workload`) como el perfil B (con el nombre `prod-workload`) tienen selectores coincidentes para que los Pods se lancen, Fargate elige el perfil A (`beta-workload`) para los Pods. Los Pods tienen etiquetas con el perfil A en los Pods (por ejemplo, `eks.amazonaws.com/fargate-profile=beta-workload`).

Si desea migrar los Pods de Fargate existentes a los nuevos perfiles que utilizan comodines, hay dos maneras de hacerlo:

- Cree un nuevo perfil con los selectores correspondientes y, a continuación, elimine los perfiles antiguos. Los pods etiquetados con perfiles antiguos se reprograman con nuevos perfiles coincidentes.
- Si quiere migrar cargas de trabajo, pero no sabe con seguridad qué etiquetas de Fargate hay en cada Pod de Fargate, puede utilizar el siguiente método. Cree un nuevo perfil con un nombre que se ordene alfanuméricamente en primer lugar entre los perfiles del mismo clúster. A continuación, recicle los Pods de Fargate que se deban migrar a nuevos perfiles.

Creación de un perfil de Fargate

En este tema se explica cómo crear un perfil de Fargate. También debe haber creado un rol de ejecución de Pod para utilizarlo en su perfil de Fargate. Para obtener más información, consulte Rol de IAM de ejecución de Pod de Amazon EKS. Los Pods que se ejecutan en Fargate solo se admiten en subredes privadas con acceso de la puerta de enlace de NAT a Servicios de AWS, pero no una ruta directa a una puerta de enlace de Internet. Esto es para que la VPC de su clúster tenga subredes privadas disponibles. Puede crear un perfil con `eksctl` o la AWS Management Console.

En este procedimiento, se requiere la versión `0.183.0` o posterior de `eksctl`. Puede verificar la versión con el siguiente comando:

```
eksctl version
```

Para obtener instrucciones sobre cómo instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

eksctl

Cómo crear un perfil de Fargate con **`eksctl`**

Cree el perfil de Fargate con el siguiente comando de `eksctl` y reemplace cada *`example value`* con valores propios. Debe especificar un espacio de nombres. Sin embargo, la opción `--labels` no es obligatoria.

```
eksctl create fargateprofile \
    --cluster my-cluster \
    --name my-fargate-profile \
    --namespace my-kubernetes-namespace \
    --labels key=value
```

Puede usar ciertos comodines para las etiquetas *my-kubernetes-namespace* y *key=value*. Para obtener más información, consulte Comodines de perfil de Fargate.

AWS Management Console

Cómo crear un perfil de Fargate para un clúster con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el clúster para el que desea crear un perfil de Fargate.
3. Elija la pestaña Computación.
4. En Perfiles de Fargate, elija Agregar perfil de Fargate.
5. En la página Configurar perfil de Fargate, haga lo siguiente:
   a. En Name (Nombre), ingrese un nombre único para su perfil de Fargate; por ejemplo, ***my-profile***.
   b. En Pod execution role (Rol de ejecución de pods), elija el rol de ejecución de Pod que se va a utilizar con el perfil de Fargate. Solo se muestran los roles de IAM con la entidad principal del servicio de `eks-fargate-pods.amazonaws.com`. Si no ve ningún rol, debe crear uno. Para obtener más información, consulte Rol de IAM de ejecución de Pod de Amazon EKS.
   c. Modifique las Subredes seleccionadas según sea necesario.

      > **Note**
      >
      > Solo las subredes privadas son compatibles con los Pods que se ejecutan en Fargate.

   d. En Etiquetas, puede etiquetar su perfil de Fargate si lo desea. Estas etiquetas no se propagan a otros recursos asociados con el perfil, como los Pods.
   e. Elija Siguiente.
6. En la página Configurar la selección de Pod, haga lo siguiente:
   a. En Namespace (Espacio de nombres), especifique un espacio de nombres que coincida con los Pods.
      - Puede usar espacios de nombres específicos para que coincidan, como **`kube-system`** o **`default`**.

- Puede usar ciertos comodines (por ejemplo, **`prod-*`**) para que coincidan con varios espacios de nombres (por ejemplo, `prod-deployment` y `prod-test`). Para obtener más información, consulte Comodines de perfil de Fargate.

b. (Opcional) Agregue etiquetas de Kubernetes al selector. Agréguelos específicamente al selector con el que deben coincidir los Pods del espacio de nombres especificado.

- Puede agregar la etiqueta **`infrastructure: fargate`** al selector para que solo los Pods del espacio de nombres especificado que también tengan la etiqueta `infrastructure: fargate` de Kubernetes coincidan con el selector.
- Puede usar ciertos comodines (por ejemplo, **`key?: value?`**) para que coincidan con varios espacios de nombres (por ejemplo, `keya: valuea` y `keyb: valueb`). Para obtener más información, consulte Comodines de perfil de Fargate.

c. Elija Siguiente.

7. En la página Revisar y crear, revise la información de su perfil de Fargate y elija Crear.

Eliminación de un perfil de Fargate

En este tema se explica cómo eliminar un perfil de Fargate.

Al eliminar un perfil de Fargate, se eliminan todos los Pods que se programaron en Fargate con el perfil. Si esos Pods coinciden con otro perfil de Fargate, se programan en Fargate con ese perfil. Si ya no coinciden con ningún perfil de Fargate, significa que no están programados en Fargate y pueden permanecer como pendientes.

Solo un perfil de Fargate de un clúster puede tener el estado `DELETING` a la vez. Espere a que un perfil de Fargate termine de eliminarse para poder eliminar cualquier otro perfil de ese clúster.

Puede eliminar un perfil con `eksctl`, la AWS Management Console o la AWS CLI. Seleccione la pestaña con el nombre de la herramienta con la que desea eliminar el perfil.

eksctl

Para eliminar un perfil de Fargate con **`eksctl`**

Utilice el siguiente comando para eliminar un perfil de un clúster. Reemplace cada *`example value`* con valores propios.

```
eksctl delete fargateprofile  --name my-profile --cluster my-cluster
```

AWS Management Console

Para eliminar un perfil de Fargate de un clúster con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres). En la lista de clústeres, elija el clúster del que desea eliminar el perfil de Fargate.
3. Elija la pestaña Computación.
4. Elija el perfil de Fargate que desea eliminar y, a continuación, elija Delete (Eliminar).
5. En la página Delete Fargate Profile (Eliminar perfil de Fargate), ingrese el nombre del perfil y, a continuación, elija Delete (Eliminar).

AWS CLI

Para eliminar un perfil de Fargate con la AWS CLI

Utilice el siguiente comando para eliminar un perfil de un clúster. Reemplace cada *example value* con valores propios.

```
aws eks delete-fargate-profile --fargate-profile-name my-profile --cluster-name my-cluster
```

Configuración de Pod de Fargate

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

En esta sección se describen algunos de los detalles de configuración únicos de los Pod para ejecutar Kubernetes Pods en AWS Fargate.

CPU y memoria de Pod

Con Kubernetes, puede definir las solicitudes, la cantidad mínima de vCPU y los recursos de memoria que se asignan a cada contenedor de un Pod. Los Pods están programados por Kubernetes para garantizar que al menos los recursos solicitados para cada Pod estén

disponibles en el recurso de cómputo. Para obtener más información, consulte la Administración de recursos de computación para contenedores en la documentación de Kubernetes.

> **Note**
>
> Dado que Amazon EKS Fargate ejecuta solo un Pod por nodo, no se produce el escenario de expulsión de Pods en caso de menos recursos. Todos los Pods de Amazon EKS Fargate se ejecutan con prioridad garantizada, por lo que la CPU y la memoria solicitadas deben ser iguales al límite de todos los contenedores. A fin de obtener más información, consulte Configurar la calidad del servicio para los Pods en la documentación de Kubernetes.

Cuando los Pods están programados en Fargate, las reservas de vCPU y memoria dentro de la especificación del Pod determinan cuánta CPU y memoria se debe aprovisionar para el Pod.

- La solicitud máxima de cualquier contenedor Init se utiliza para determinar los requisitos de vCPU y de memoria de la solicitud Init.
- Las solicitudes para todos los contenedores de larga duración se suman para determinar los requisitos de memoria y vCPU de las solicitudes de larga duración.
- El mayor de los dos valores anteriores se elige para la solicitud de vCPU y de memoria que se utilizará para el Pod.
- Fargate agrega 256 MB a la reserva de memoria de cada Pod para los componentes requeridos de Kubernetes (`kubelet`, `kube-proxy` y `containerd`).

Fargate redondea hasta la siguiente configuración informática que más se acerque a la suma de las solicitudes de vCPU y memoria para garantizar que los Pods siempre tengan los recursos que necesitan para ejecutarse.

Si no especifica una combinación de vCPU y memoria, se utilizará la menor combinación disponible (0,25 vCPU y 0,5 GB de memoria).

La siguiente tabla muestra las combinaciones de vCPU y memoria disponibles para los Pods que se ejecutan en Fargate.

Valor de vCPU	Valor de memoria
0,25 vCPU	0,5 GB, 1 GB, 2 GB

Valor de vCPU	Valor de memoria
0,5 vCPU	1 GB, 2 GB, 3 GB, 4 GB
1 vCPU	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB
2 vCPU	Entre 4 GB y 16 GB en incrementos de 1 GB
4 vCPU	Entre 8 GB y 30 GB en incrementos de 1 GB
8 vCPU	Entre 16 GB y 60 GB en incrementos de 4 GB
16 vCPU	Entre 32 GB y 120 GB en incrementos de 8 GB

La memoria adicional reservada para los componentes de Kubernetes puede generar una tarea de Fargate con más vCPU de las que se solicitaron aprovisionar. Por ejemplo, una solicitud de 1 vCPU y 8 GB de memoria tendrá 256 MB agregados a su solicitud de memoria y aprovisionará una tarea de Fargate con 2 vCPU y 9 GB de memoria, ya que no hay ninguna tarea con 1 vCPU y 9 GB de memoria disponible.

No hay correlación entre el tamaño del Pod que se ejecuta en Fargate y el tamaño del nodo informado por Kubernetes con `kubectl get nodes`. El tamaño del nodo informado suele ser mayor que la capacidad del Pod. Puede verificar la capacidad de los Pod con el siguiente comando. Reemplace *`default`* con espacio de nombres de Pod y *`pod-name`* con el nombre de Pod.

```
kubectl describe pod --namespace default pod-name
```

Un ejemplo de salida sería el siguiente.

```
[...]
annotations:
    CapacityProvisioned: 0.25vCPU 0.5GB
[...]
```

El comentario `CapacityProvisioned` representa la capacidad del Pod forzada y determina el costo del Pod que se ejecuta en Fargate. Para obtener información sobre los precios de las configuraciones informáticas, consulte Precios de AWS Fargate.

Almacenamiento de Fargate

Un Pod que se ejecuta en Fargate monta automáticamente un sistema de archivos de Amazon EFS. No se puede utilizar el aprovisionamiento dinámico de volúmenes persistentes con nodos de Fargate, pero se puede utilizar el aprovisionamiento estático. Para obtener más información, consulte Controlador de CSI de Amazon EFS en GitHub.

Cuando se aprovisiona, cada Pod que se ejecuta en Fargate recibe un almacenamiento efímero predeterminado de 20 GiB. Este tipo de almacenamiento se elimina después de un Pod se detiene. Los nuevos Pods lanzados en Fargate tienen el cifrado del volumen de almacenamiento efímero habilitado de forma predeterminada. El almacenamiento de Pod efímero se cifra con un algoritmo de cifrado AES-256 mediante claves administradas por AWS Fargate.

> **Note**
>
> El almacenamiento utilizable predeterminado para Pods de Amazon EKS que se ejecuta en Fargate tiene menos de 20 GiB. Esto se debe a que parte del espacio es utilizado por `kubelet` y otros módulos de Kubernetes que se cargan dentro del Pod.

La cantidad total de almacenamiento efímero se puede aumentar hasta un máximo de 175 GiB. Para configurar el tamaño con Kubernetes, especifique las solicitudes del recurso de `ephemeral-storage` para cada contenedor en un Pod. Cuando Kubernetes programa Pods, asegura que la suma de las solicitudes de recursos para cada Pod es inferior a la capacidad de la tarea de Fargate. Para obtener más información, consulte Administración de recursos para Pods y contenedores en la documentación de Kubernetes.

Amazon EKS Fargate proporciona más almacenamiento efímero del solicitado para el uso del sistema. Por ejemplo, una solicitud de 100 GiB aprovisionará una tarea de Fargate con 115 GiB de almacenamiento efímero.

Parches de SO de Fargate

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

Amazon EKS debe aplicar parches periódicamente de SO para nodos de AWS Fargate a fin de mantenerlos seguros. Como parte del proceso de aplicación de parches, reciclamos los nodos para instalar los parches del sistema operativo. Las actualizaciones se intentan de tal manera que se produzca el menor impacto en sus servicios. Sin embargo, si los Pods no se expulsan correctamente, hay ocasiones en que deben eliminarse. A continuación, se muestran las acciones que puede realizar para minimizar las posibles interrupciones:

- Establezca los presupuestos de interrupción del Pod (PDB) adecuados para controlar el número de Pods que están inactivos simultáneamente.
- Cree reglas de Amazon EventBridge para controlar las expulsiones fallidas antes de que se eliminen los Pods.
- Cree una configuración de notificaciones en Notificaciones de usuario de AWS.

Amazon EKS trabaja en estrecha colaboración con la comunidad de Kubernetes para que las correcciones de errores y los parches de seguridad estén disponibles lo antes posible. Todos los Pods de Fargate empiezan en la versión de parche de Kubernetes más recientes, que está disponible en Amazon EKS para la versión Kubernetes de su clúster. Si tiene un Pod con una versión de parche anterior, Amazon EKS podría reciclarlo para actualizarlo a la última versión. Esto garantiza que los Pods estén equipados con las últimas actualizaciones de seguridad. De esa forma, si hay un problema de Vulnerabilidades y exposiciones comunes (CVE) crítico, se mantiene actualizado para reducir los riesgos de seguridad.

Para limitar el número de Pods que están inactivos al mismo tiempo cuando se reciclan Pods, puede establecer presupuestos de interrupción de Pod (PDB). Puede utilizar PDB para definir la disponibilidad mínima en función de los requisitos de cada una de las aplicaciones y, al mismo tiempo, permitir que se produzcan actualizaciones. Para obtener más información, consulte Especificación de un presupuesto de interrupción para su aplicación en la Documentación de Kubernetes.

Amazon EKS utiliza la API de expulsión para drenar el Pod de forma segura respetando los PDB que configuró para la aplicación. La zona de disponibilidad expulsa los pods para minimizar el impacto. Si la expulsión tiene éxito, el nuevo Pod obtiene el parche más reciente y no será necesario realizar más acciones.

Cuando falla la expulsión de un Pod, Amazon EKS envía un evento a su cuenta con detalles sobre los Pods que han fallado la expulsión. Puede actuar en función del mensaje antes de la hora de finalización programada. El tiempo específico varía según la urgencia del parche. Cuando llega el

momento, Amazon EKS intenta expulsar de nuevo los Pods. Sin embargo, esta vez no se envía un nuevo evento si la expulsión falla. Si la expulsión vuelve a fallar, los Pods existentes se eliminan periódicamente para que los nuevos Pods puedan tener el último parche.

A continuación, se muestra un evento de ejemplo recibido cuando falla la expulsión de Pod. Contiene detalles sobre el clúster, el nombre del Pod, el espacio de nombres del Pod, el perfil de Fargate y la hora de finalización programada.

```
{
    "version": "0",
    "id": "12345678-90ab-cdef-0123-4567890abcde",
    "detail-type": "EKS Fargate Pod Scheduled Termination",
    "source": "aws.eks",
    "account": "111122223333",
    "time": "2021-06-27T12:52:44Z",
    "region": "region-code",
    "resources": [
        "default/my-database-deployment"
    ],
    "detail": {
        "clusterName": "my-cluster",
        "fargateProfileName": "my-fargate-profile",
        "podName": "my-pod-name",
        "podNamespace": "default",
        "evictErrorMessage": "Cannot evict pod as it would violate the pod's disruption
 budget",
        "scheduledTerminationTime": "2021-06-30T12:52:44.832Z[UTC]"
    }
}
```

Además, tener varios PDB asociados a un Pod puede provocar un error de expulsión. Este evento devuelve el siguiente mensaje de error.

```
"evictErrorMessage": "This pod has multiple PodDisruptionBudget, which the eviction
 subresource does not support",
```

Puede crear una acción deseada basada en este evento. Por ejemplo, puede ajustar el presupuesto de interrupción del Pod (PDB) para controlar cómo se expulsan los Pods. Más concretamente, supongamos que empieza con un PDB que especifica el porcentaje objetivo de Pods que están disponibles. Antes de que los Pods finalicen forzosamente durante una actualización, puedes ajustar el PDB a un porcentaje diferente de Pods. Para recibir este evento, debe crear una regla de Amazon

EventBridge en la Cuenta de AWS y en Región de AWS a la que pertenece el clúster. La regla debe utilizar el siguiente patrón personalizado. Para obtener más información, consulte Creación de reglas de EventBridge que reaccionan a eventos en la Guía del usuario de Amazon EventBridge.

```
{
  "source": ["aws.eks"],
  "detail-type": ["EKS Fargate Pod Scheduled Termination"]
}
```

Se puede establecer un objetivo adecuado para que el evento lo capture. Para obtener una lista completa de los destinos admitidos, consulte Destinos de Amazon EventBridge en la Guía del usuario de Amazon EventBridge. También puede crear una configuración de notificaciones en Notificaciones de usuario de AWS. Al utilizar la AWS Management Console para crear la notificación, en Reglas de eventos, seleccione Elastic Kubernetes Service (EKS) como nombre del Servicio de AWS y Terminación programada de EKS Fargate Pod para el tipo de evento. Para obtener más información, consulte Introducción a las notificaciones de usuario de AWS en la Guía de usuario de notificaciones de usuario de AWS.

Métricas de Fargate

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (EE. UU. Este) y AWS GovCloud (EE. UU. Oeste).

Puede recopilar métricas del sistema y métricas de uso de CloudWatch para AWS Fargate.

Métricas de aplicación

Para aplicaciones que se ejecutan en Amazon EKS y AWS Fargate, puede utilizar AWS Distro for OpenTelemetry (ADOT). ADOT le permite recopilar métricas del sistema y enviarlas a los paneles de CloudWatch Container Insights. Para empezar a utilizar ADOT para aplicaciones que se ejecutan en Fargate, consulte Uso de Información de contenedores de CloudWatch con AWS Distro para OpenTelemetry en la documentación de ADOT.

Métricas de uso

Puede utilizar las métricas de uso de CloudWatch para proporcionar visibilidad sobre el uso de los recursos de su cuenta. Utilice estas métricas para visualizar el uso actual del servicio en paneles y gráficos de CloudWatch.

Las métricas de uso de AWS Fargate se corresponden con las cuotas de servicio de AWS. Puede configurar alarmas que le avisen cuando su uso se acerque a una cuota de servicio. Para obtener más información acerca de las cuotas de servicio de Fargate, consulte Cuotas de servicio de Amazon EKS.

AWS Fargate publica las siguientes métricas en el espacio de nombres `AWS/Usage`.

Métrica	Descripción
`ResourceCount`	El número total de los recursos especificados que se ejecutan en su cuenta. Los recursos se definen por las dimensiones asociadas a la métrica.

Las siguientes dimensiones se utilizan para ajustar las métricas de uso publicadas por AWS Fargate.

Dimensión	Descripción
`Service`	El nombre del servicio de AWS que contiene el recurso. Para las métricas de uso de AWS Fargate, el valor de esta dimensión es `Fargate`.
`Type`	El tipo de entidad que se notifica. Actualmente, el único valor válido para las métricas de uso de AWS Fargate es `Resource`.
`Resource`	El tipo de recurso que se ejecuta. En la actualidad, AWS Fargate devuelve información sobre la utilización bajo demanda de Fargate. El valor de recurso para la utilización bajo demanda de Fargate es `OnDemand`.

Dimensión	Descripción
	Note La utilización bajo demanda de Fargate combina los Pods de Amazon EKS que utilizan Fargate, las tareas de Amazon ECS que utilizan el tipo de lanzamiento de Fargate y las tareas de Amazon ECS que utilizan el proveedor de capacidad de `FARGATE`.
`Class`	La clase de recurso a la que se realiza el seguimiento. En la actualidad, AWS Fargate no utiliza la dimensión de clase.

Creación de una alarma de CloudWatch para monitorear las métricas de uso de recursos de Fargate

AWS Fargate proporciona métricas de uso de CloudWatch que corresponden a las cuotas de servicio de AWS para el uso de recursos bajo demanda de Fargate En la consola de Service Quotas, puede ver el uso en un gráfico. También puede configurar alarmas que le avisen cuando su uso se acerque a una cuota de servicio. Para obtener más información, consulte Métricas de Fargate.

Siga estos pasos para crear una alarma de CloudWatch basada en las métricas de uso de recursos de Fargate.

Para crear una alarma basada en las cuotas de uso de Fargate (AWS Management Console)

1. Abra la consola de Service Quotas en https://console.aws.amazon.com/servicequotas/.
2. En el panel de navegación de la izquierda, elija Servicios de AWS.
3. En la lista AWS services, busque y seleccione AWS Fargate.
4. En la lista Service quotas (Cuotas de servicio), elija la cuota de utilización de Fargate para la que desee crear una alarma.
5. En la sección Amazon CloudWatch alarms (Alarmas de Amazon CloudWatch), elija Create (Crear).
6. En Alarm threshold (Umbral de alarma), elija el porcentaje del valor de la cuota aplicada que desee establecer como valor de la alarma.
7. En Alarm name (Nombre de la alarma), escriba el nombre de la alarma y elija Create (Crear).

Registros de Fargate

> **Important**
>
> AWS Fargate con Amazon EKS no está disponible en AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.).

Amazon EKS en Fargate ofrece un enrutador de registros integrado basado en Fluent Bit. Esto significa que no ejecuta explícitamente un contenedor de Fluent Bit como archivo sidecar, sino que Amazon lo ejecuta por usted. Todo lo que tiene que hacer es configurar el enrutador de registros. La configuración se realiza a través de un `ConfigMap` dedicado que debe cumplir los siguientes criterios:

- Tener un `aws-logging` con nombre.
- Haber sido creado en un espacio de nombres dedicado llamado `aws-observability`.
- No puede superar los 5300 caracteres.

Una vez que haya creado `ConfigMap`, Amazon EKS en Fargate lo detecta automáticamente y configura el enrutador de registros con él. Fargate utiliza una versión de AWS para Fluent Bit, una distribución conforme del cliente al servidor de Fluent Bit administrada por AWS. Para obtener más información, consulte AWS para Fluent Bit en GitHub.

El enrutador de registros le permite utilizar la amplia gama de servicios de AWS para el análisis y el almacenamiento de registros. Puede transmitir registros desde Fargate directamente a Amazon CloudWatch, Amazon OpenSearch Service. También puede transmitir registros a destinos como Amazon S3, Amazon Kinesis Data Streams y herramientas de socios a través de Amazon Data Firehose.

Requisitos previos

- Un perfil de Fargate existente que especifica un espacio de nombres de Kubernetes existente en el que se implementan Pods de Fargate. Para obtener más información, consulte Crear un perfil de Fargate para el clúster.
- Un rol de ejecución de Pod de Fargate existente. Para obtener más información, consulte Crear un rol de ejecución de Pod de Fargate.

Configuración del enrutador de registros

Configurar el enrutador de registros

En los pasos siguientes, reemplace cada *example value* por valores propios.

1. Cree un espacio de nombres Kubernetes dedicado denominado `aws-observability`.

   a. Guarde el siguiente contenido en un archivo llamado *aws-observability-namespace*`.yaml` en el equipo. El valor de `name` debe ser `aws-observability` y la etiqueta `aws-observability: enabled` es obligatoria.

   ```
   kind: Namespace
   apiVersion: v1
   metadata:
     name: aws-observability
     labels:
       aws-observability: enabled
   ```

   b. Cree el espacio de nombres.

   ```
   kubectl apply -f aws-observability-namespace.yaml
   ```

2. Cree un `ConfigMap` con un valor de datos de `Fluent Conf` para enviar los registros de contenedores a un destino.. Fluent Conf es Fluent Bit, que es un lenguaje de configuración del procesador de registros rápido y ligero que se utiliza para dirigir los registros del contenedor a un destino de registro de su elección. Para obtener más información, consulte Archivo de configuración en la documentación de Fluent Bit.

   > **Important**
   >
   > En un `Fluent Conf` típico, las secciones principales incluidas son `Service`, `Input`, `Filter` y `Output`. Sin embargo, el enrutador de registros de Fargate solo acepta:
   >
   > - Las secciones `Filter` y `Output`.
   > - Una sección `Parser`.
   >
   > Si proporciona otras secciones, se rechazarán.

El enrutador de registros de Fargate administra las secciones `Service` e `Input`. Tiene la siguiente sección `Input`, la cual no se puede modificar y no es necesaria en su `ConfigMap`. Sin embargo, puede obtener información a partir de ella, como el límite del búfer de memoria y la etiqueta aplicada a los registros.

```
[INPUT]
    Name tail
    Buffer_Max_Size 66KB
    DB /var/log/flb_kube.db
    Mem_Buf_Limit 45MB
    Path /var/log/containers/*.log
    Read_From_Head On
    Refresh_Interval 10
    Rotate_Wait 30
    Skip_Long_Lines On
    Tag kube.*
```

Al crear el `ConfigMap`, debe tener en cuenta las siguientes reglas que Fargate utiliza para validar campos:

- Se supone que `[FILTER]`, `[OUTPUT]` y `[PARSER]` deben especificarse en cada clave correspondiente. Por ejemplo, `[FILTER]` debe estar en `filters.conf`. Puede tener uno o más `[FILTER]` en `filters.conf`. Las secciones `[OUTPUT]` y `[PARSER]` también deben estar en sus claves correspondientes. Mediante la especificación de varias secciones `[OUTPUT]`, puede dirigir sus registros a diferentes destinos al mismo tiempo.
- Fargate valida las claves requeridas para cada sección. `Name` y `match` son necesarias para cada `[FILTER]` y `[OUTPUT]`. `Name` y `format` son necesarias para cada `[PARSER]`. Las claves distinguen entre mayúsculas y minúsculas.
- Las variables de entorno como `${ENV_VAR}` no se permiten en `ConfigMap`.
- La sangría tiene que ser la misma para la política o el par clave-valor dentro de cada `filters.conf`, `output.conf` y `parsers.conf`. Los pares clave-valor deben tener más sangría que las políticas.
- Fargate valida con los siguientes filtros compatibles: `grep`, `parser`, `record_modifier`, `rewrite_tag`, `throttle`, `nest`, `modify` y `kubernetes`.
- Fargate verifica las siguientes salidas compatibles: `es`, `firehose`, `kinesis_firehose`, `cloudwatch`, `cloudwatch_logs` y `kinesis`.

- Tiene que proporcionarse al menos un complemento de `Output` compatible en el `ConfigMap` a fin de habilitar el registro. `Filter` y `Parser` no son necesarios para habilitar el registro.

También puede ejecutar Fluent Bit en Amazon EC2 con la configuración deseada para solucionar cualquier problema que surja de la validación. Cree su `ConfigMap` con uno de los siguientes ejemplos.

> **Important**
>
> El registro de Fargate de Amazon EKS no admite la configuración dinámica del `ConfigMaps`. Cualquier cambio en `ConfigMaps` solo se aplica a los Pods nuevos. Los cambios no se aplican a los Pods existentes.

Cree un `ConfigMap` con el ejemplo para el destino de registro deseado.

> **Note**
>
> También puede utilizar Amazon Kinesis Data Streams como destino para su registro. Si usa Kinesis Data Streams, asegúrese de que la función de ejecución del pod tenga otorgado el permiso `kinesis:PutRecords`. Para obtener más información, consulte Permisos de Amazon Kinesis Data Streams en Fluent Bit: Manual oficial.

CloudWatch

Para crear un **`ConfigMap`** para CloudWatch

Tiene dos opciones de salida al utilizar CloudWatch:

- Un complemento de salida escrito en C
- Un complemento de salida escrito en Golang

En el siguiente ejemplo se muestra cómo utilizar el complemento de `cloudwatch_logs` para enviar registros a CloudWatch.

1. Guarde los siguientes contenidos en un archivo llamado *aws-logging-cloudwatch-configmap*.yaml. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Se requieren los parámetros en [OUTPUT].

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
data:
  flb_log_cw: "false"  # Set to true to ship Fluent Bit process logs to
 CloudWatch.
  filters.conf: |
    [FILTER]
        Name parser
        Match *
        Key_name log
        Parser crio
    [FILTER]
        Name kubernetes
        Match kube.*
        Merge_Log On
        Keep_Log Off
        Buffer_Size 0
        Kube_Meta_Cache_TTL 300s
  output.conf: |
    [OUTPUT]
        Name cloudwatch_logs
        Match   kube.*
        region region-code
        log_group_name my-logs
        log_stream_prefix from-fluent-bit-
        log_retention_days 60
        auto_create_group true
  parsers.conf: |
    [PARSER]
        Name crio
        Format Regex
        Regex ^(?<time>[^ ]+) (?<stream>stdout|stderr) (?<logtag>P|F) (?
<log>.*)$
        Time_Key    time
        Time_Format %Y-%m-%dT%H:%M:%S.%L%z
```

2. Aplique el manifiesto al clúster.

```
kubectl apply -f aws-logging-cloudwatch-configmap.yaml
```

3. Descargue la política de IAM de CloudWatch en su equipo. También puede ver la política en GitHub.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-eks-fluent-logging-examples/mainline/examples/fargate/cloudwatchlogs/permissions.json
```

Amazon OpenSearch Service

Para crear un **ConfigMap** de Amazon OpenSearch Service

Si desea enviar registros a Amazon OpenSearch Service, puede utilizar salida es, que es un complemento escrito en C. En el siguiente ejemplo se muestra cómo utilizar el complemento para enviar registros a OpenSearch.

1. Guarde los siguientes contenidos en un archivo llamado *aws-logging-opensearch-configmap*.yaml. Sustituya cada *example value* con valores propios.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
data:
  output.conf: |
    [OUTPUT]
      Name  es
      Match *
      Host  search-example-gjxdcilagiprbglqn42jsty66y.region-code.es.amazonaws.com
      Port  443
      Index example
      Type  example_type
      AWS_Auth On
      AWS_Region region-code
      tls   On
```

2. Aplique el manifiesto al clúster.

```
kubectl apply -f aws-logging-opensearch-configmap.yaml
```

3. Descargue la política de IAM de OpenSearch en su ordenador. También puede ver la política en GitHub.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-eks-
fluent-logging-examples/mainline/examples/fargate/amazon-elasticsearch/
permissions.json
```

   Asegúrese de que el control de acceso de OpenSearch Dashboards esté configurado correctamente. El `all_access role` en OpenSearch Dashboards debe tener el rol de ejecución de Pod de Fargate y el rol de IAM asignados. Se debe hacer el mismo mapeo para el rol de `security_manager`. Puede agregar los mapeos anteriores al seleccionar `Menu`, `Security` y `Roles` y, a continuación, seleccionar los roles respectivos. Para obtener más información, consulte ¿Cómo puedo solucionar los problemas de CloudWatch Logs para que transmita a mi dominio de Amazon ES?.

Firehose

Para crear una **`ConfigMap`** para Firehose

Tiene dos opciones de salida al enviar registros a Firehose:

- `kinesis_firehose`: un complemento de salida escrito en C.
- `firehose`: un complemento de salida escrito en Golang.

En el siguiente ejemplo se muestra cómo utilizar el complemento de `kinesis_firehose` para enviar registros a Firehose.

1. Guarde los siguientes contenidos en un archivo llamado *`aws-logging-firehose-configmap`*`.yaml`. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
```

```
data:
  output.conf: |
    [OUTPUT]
     Name  kinesis_firehose
     Match *
     region region-code
     delivery_stream my-stream-firehose
```

2. Aplique el manifiesto al clúster.

```
kubectl apply -f aws-logging-firehose-configmap.yaml
```

3. Descargue la política de IAM de Firehose en su ordenador. También puede ver la política en GitHub.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-eks-fluent-logging-examples/mainline/examples/fargate/kinesis-firehose/permissions.json
```

3. Cree una política de IAM a partir del archivo de política que descargó en el paso anterior.

```
aws iam create-policy --policy-name eks-fargate-logging-policy --policy-document
 file://permissions.json
```

4. Adjunte la política de IAM al rol de ejecución de pods especificado para el perfil de Fargate con el siguiente comando. Reemplace *111122223333* por su ID de cuenta. Reemplace *AmazonEKSFargatePodExecutionRole* por el rol de ejecución de Pod (para obtener más información, consulte Crear un rol de ejecución de Pod de Fargate).

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::111122223333:policy/eks-fargate-logging-policy \
  --role-name AmazonEKSFargatePodExecutionRole
```

Compatibilidad de filtro de Kubernetes

Esta característica requiere la siguiente versión mínima de Kubernetes y nivel de plataforma, o posterior.

Versión de Kubernetes	Nivel de plataforma
1.23 y posterior	eks.1

El filtro Fluent Bit Kubernetes le permite agregar metadatos de Kubernetes a los archivos de registro. Para obtener más información acerca del filtro, consulte Kubernetes en la documentación de Fluent Bit. Puede aplicar un filtro mediante el punto de conexión del servidor de la API.

```
filters.conf: |
    [FILTER]
        Name                kubernetes
        Match               kube.*
        Merge_Log           On
        Buffer_Size         0
        Kube_Meta_Cache_TTL 300s
```

Important

- `Kube_URL`, `Kube_CA_File`, `Kube_Token_Command` y `Kube_Token_File` son parámetros de configuración propiedad del servicio y no deben especificarse. Amazon EKS Fargate completa estos valores.
- `Kube_Meta_Cache_TTL` es el tiempo que Fluent Bit espera hasta que se comunica con el servidor de la API para obtener los metadatos más recientes. Si `Kube_Meta_Cache_TTL` no se especifica, Amazon EKS Fargate agrega un valor predeterminado de 30 minutos para reducir la carga en el servidor de la API.

Para enviar registros de procesos de Fluent Bit a su cuenta

Puede enviar opcionalmente registros de procesos de Fluent Bit a Amazon CloudWatch mediante el siguiente `ConfigMap`. El envío de registros de procesos de Fluent Bit a CloudWatch requiere costos adicionales de ingesta y almacenamiento de registros. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: aws-logging
  namespace: aws-observability
  labels:
data:
  # Configuration files: server, input, filters and output
  # ======================================================
```

```
  flb_log_cw: "true"  # Ships Fluent Bit process logs to CloudWatch.

  output.conf: |
    [OUTPUT]
        Name cloudwatch
        Match kube.*
        region region-code
        log_group_name fluent-bit-cloudwatch
        log_stream_prefix from-fluent-bit-
        auto_create_group true
```

Los registros se encuentran en la Región de AWS en la que reside el clúster en CloudWatch. El nombre del grupo de registros es *my-cluster*`-fluent-bit-logs` y el nombre del flujo de registros de Fluent Bit es `fluent-bit-`*podname*`-`*pod-namespace*.

> **Note**
>
> - Los registros de proceso se envían solo cuando el proceso de Fluent Bit se inicia de forma correcta. Si se produce un error al iniciar Fluent Bit, se pierden los registros del proceso. Solo puede enviar los registros del proceso a CloudWatch.
> - Para depurar los registros del proceso de envío en la cuenta, puede aplicar el `ConfigMap` anterior para obtener los registros del proceso. El hecho de que Fluent Bit no pueda iniciar suele deberse a que su `ConfigMap` no ha sido analizado ni aceptado por Fluent Bit durante el inicio.

Para detener el envío de registros de procesos de Fluent Bit

El envío de registros de procesos de Fluent Bit a CloudWatch requiere costos adicionales de ingesta y almacenamiento de registros. Para excluir los registros de procesos de una configuración de `ConfigMap` existente, siga estos pasos.

1. Busque el grupo de registros de CloudWatch creado automáticamente para los registros de procesos de Fluent Bit del clúster de Amazon EKS después de habilitar el registro de Fargate. Sigue el formato `{cluster_name}-fluent-bit-logs`.
2. Elimine los flujos de registro de CloudWatch existentes creados para los registros de procesos de cada Pod's en el grupo de registros de CloudWatch.
3. Edite el `ConfigMap` y configure `flb_log_cw: "false"`.

4. Reinicie todos los Pods existentes en el clúster.

Probar la aplicación

1. Implemente un Pod de ejemplo.

   a. Guarde el siguiente contenido en un archivo llamado *sample-app*.yaml en el equipo.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sample-app
  namespace: same-namespace-as-your-fargate-profile
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - name: http
              containerPort: 80
```

   b. Aplique el manifiesto al clúster.

```
kubectl apply -f sample-app.yaml
```

2. Vea los registros NGINX con los destinos que configuró en el ConfigMap.

Consideraciones sobre el tamaño

Le sugerimos que planee utilizar hasta 50 MB de memoria para el enrutador de registros. Si anticipa que su aplicación generará registros con un rendimiento muy alto, entonces debe planificar utilizar hasta 100 MB.

Solución de problemas

Para confirmar si la característica de registro está habilitada o deshabilitada por algún motivo, como un `ConfigMap` que no es válido y desea saber por qué no es válido, verifique los eventos de Pod con **`kubectl describe pod`** ***`pod_name`***. La salida puede incluir eventos del Pod que aclaran si el registro está habilitado o no, como la siguiente salida de ejemplo.

```
[...]
Annotations:          CapacityProvisioned: 0.25vCPU 0.5GB
                      Logging: LoggingDisabled: LOGGING_CONFIGMAP_NOT_FOUND
                      kubernetes.io/psp: eks.privileged
[...]
Events:
  Type     Reason           Age        From
               Message
  ----     ------           ----       ----
               -------
  Warning  LoggingDisabled  <unknown>  fargate-scheduler
                Disabled logging because aws-logging configmap was not found. configmap
 "aws-logging" not found
```

Los eventos de Pod son efímeros con un periodo de tiempo en función de la configuración. También puede ver los comentarios de un Pod's con **`kubectl describe pod`** ***`pod-name`***. En el comentario del Pod, hay información sobre si la característica de registro está habilitada o deshabilitada y el motivo.

Elección de un tipo de instancia de Amazon EC2

Amazon EC2 proporciona una amplia selección de tipos de instancias para nodos de trabajo. Cada tipo de instancia ofrece diferentes capacidades de computación, memoria y almacenamiento. Cada instancia se agrupa también en una familia de instancias en función de dichas características. Para obtener una lista, consulte Tipos de instancias disponibles en la Guía del usuario de Amazon EC2 y Tipos de instancias disponibles en la Guía del usuario de Amazon EC2. Amazon EKS publica diferentes variaciones de las AMI de Amazon EC2 para habilitar el soporte. Para asegurarse de que el tipo de instancia que seleccione es compatible con Amazon EKS, tenga en cuenta los siguientes criterios.

- En la actualidad, las AMI de Amazon EKS no admiten las familias `g5g` y `mac`.
- Las AMI de Arm y no aceleradas de Amazon EKS no admiten las familias `g3`, `g4`, `inf` y `p`.

- Las AMI aceleradas de Amazon EKS no admiten las familias `a`, `c`, `hpc`, `m` y `t`.
- Para las instancias basadas en ARM, Amazon Linux 2023 (AL2023) solo admite tipos de instancias que utilizan procesadores Graviton2 o posteriores. AL2023 no admite instancias `A1`.

Al elegir entre los tipos de instancias admitidos por Amazon EKS, tenga en cuenta las siguientes capacidades de cada tipo.

Número de instancias de un grupo de nodos

En general, que haya menos instancias y que sean más grandes es mejor, especialmente si tiene muchos Daemonsets. Cada instancia requiere llamadas a la API para el servidor de API, por lo que cuantas más instancias tenga, más carga tendrá el servidor de API.

Sistema operativo

Revise los tipos de instancias admitidos para Linux, Windows y Bottlerocket. Antes de crear instancias de Windows, revise Activación de la compatibilidad con Windows para su clúster de Amazon EKS.

Arquitectura de hardware

¿Necesita x86 o Arm? Solo puede implementar Linux en Arm. Antes de implementar instancias de Arm, revise AMI de Amazon Linux optimizada para Amazon EKS Arm. ¿Necesita instancias integradas en Nitro System (Linux o Windows) o que tengan capacidades aceleradas? Si necesita capacidades aceleradas, solo puede utilizar Linux con Amazon EKS.

Número máximo de Pods

Dado que a cada Pod se le asigna su propia dirección IP, la cantidad de direcciones IP admitidas por un tipo de instancia es un factor que se considera a la hora de determinar el número de Pods que se pueden ejecutar en la instancia. Para determinar en forma manual cuántos Pods admite un tipo de instancia, consulte Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2.

> **Note**
>
> Si utiliza una AMI de Amazon Linux 2 optimizada para Amazon EKS, `v20220406` o posterior, puede utilizar un nuevo tipo de instancia sin actualizar a la última AMI. Para estas AMI, la AMI calcula automáticamente el valor `max-pods` necesario si no se incluye en el archivo `eni-max-pods.txt`. Es posible que Amazon EKS no admita los tipos de

instancias que se encuentran en vista previa de forma predeterminada. Aún se deben agregar valores para `max-pods` para estos tipos a `eni-max-pods.txt` en nuestra AMI.

Los tipos de instancia AWS Nitro System admiten opcionalmente más direcciones IP que los tipos de instancias que no son Nitro System. Sin embargo, no todas las direcciones IP asignadas a una instancia están disponibles para los Pods. Para asignar un número significativamente mayor de direcciones IP a sus instancias, debe tener la versión `1.9.0` o posterior del complemento Amazon VPC CNI instalada en el clúster y configurada de forma adecuada. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2. Para asignar el mayor número de direcciones IP a sus instancias, debe tener la versión `1.10.1` o posterior del complemento Amazon VPC CNI instalada en su clúster, e implementar este con la familia `IPv6`.

Familia de IP

Puede usar cualquier tipo de instancia compatible cuando utilice la familia `IPv4` para un clúster, que permite que su clúster asigne direcciones privadas `IPv4` a sus Pods y servicios. Pero si desea usar la familia `IPv6` para su clúster, entonces debe usar tipos de instancias AWS Nitro System o tipos de ejemplares bare metal. Solo se admite `IPv4` en las instancias de Windows. Su clúster debe ejecutar la versión `1.10.1` o posterior del complemento Amazon VPC CNI. Para obtener más información acerca del uso de `IPv6`, consulte Direcciones `IPv6` de clústeres, Pods y services.

Versión del complemento CNI de Amazon VPC que ejecuta

La versión más reciente del complemento CNI de Amazon VPC para Kubernetes es compatible con estos tipos de instancias. Es posible que tenga que actualizar la versión del complemento CNI de Amazon VPC para aprovechar los últimos tipos de instancia admitidos. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS. La última versión admite las características más recientes para el uso con Amazon EKS. Las versiones anteriores no admiten todas las características. Puede ver las características compatibles con las distintas versiones en Changelog en GitHub.

Región de AWS en la que va a crear los nodos

No todos los tipos de instancias están disponibles en todas las Regiones de AWS.

Si utiliza grupos de seguridad para Pods

Si utiliza grupos de seguridad para Pods, solo se admiten tipos de instancia específicos. Para obtener más información, consulte Grupos de seguridad de Pods.

Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2

Dado que a cada Pod se le asigna su propia dirección IP, la cantidad de direcciones IP admitidas por un tipo de instancia es un factor que se considera a la hora de determinar el número de Pods que se pueden ejecutar en la instancia. Amazon EKS proporciona un script que puede descargar y ejecutar para determinar el número máximo de Pods recomendado por Amazon EKS para ejecutar en cada tipo de instancia. El script utiliza los atributos de hardware de cada instancia y las opciones de configuración para determinar el número máximo de Pods. Puede utilizar el número devuelto en estos pasos para habilitar capacidades como la asignación de direcciones IP a Pods desde una subred diferente a la de la instancia y el aumento significativo del número de direcciones IP de la instancia. Si utiliza un grupo de nodos administrado con varios tipos de instancias, utilice un valor que funcione para todos los tipos de instancias.

1. Descargue un script que pueda utilizar para calcular el número máximo de Pods para cada tipo de instancia.

   ```
   curl -O https://raw.githubusercontent.com/awslabs/amazon-eks-ami/master/templates/al2/runtime/max-pods-calculator.sh
   ```

2. Marque el script como ejecutable en el equipo.

   ```
   chmod +x max-pods-calculator.sh
   ```

3. Ejecute el script, mediante el reemplazo de *`m5.large`* por el tipo de instancia que planea implementar y *`1.9.0-eksbuild.1`* por su versión del complemento CNI de Amazon VPC. Para determinar la versión del complemento, consulte los procedimientos de actualización en Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.

   ```
   ./max-pods-calculator.sh --instance-type m5.large --cni-version 1.9.0-eksbuild.1
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   29
   ```

   Puede agregar las siguientes opciones al script para ver el número máximo de Pods admitido cuando se utilizan capacidades opcionales.

- `--cni-custom-networking-enabled`: utilice esta opción si desea asignar direcciones IP desde una subred distinta a la de su instancia. Para obtener más información, consulte Redes personalizadas para los pods. La adición de esta opción al script anterior con los mismos valores de ejemplo produce `20`.
- `--cni-prefix-delegation-enabled`: utilice esta opción cuando desee asignar un número significativamente mayor de direcciones IP a cada interfaz de red elástica. Esta capacidad requiere una instancia de Amazon Linux que se ejecute en Nitro System y en la versión `1.9.0` o posterior del complemento CNI de Amazon VPC. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2. La adición de esta opción al script anterior con los mismos valores de ejemplo produce `110`.

También puede ejecutar el script con la opción `--help` para ver todas las opciones disponibles.

> **Note**
>
> El script para el cálculo del máximo de Pods limita el valor devuelto a `110` en función de los umbrales de escalabilidad de Kubernetes y la configuración recomendada. Si su tipo de instancia tiene más de 30 vCPU, este límite aumenta a `250`, un número basado en las pruebas internas del equipo de escalabilidad de Amazon EKS. Para obtener más información, consulte la entrada del blog Complemento CNI de Amazon VPC que aumenta los límites de pods por nodo.

AMI optimizadas para Amazon EKS

Puede implementar nodos con Imágenes de máquina de Amazon (AMI) optimizadas para Amazon EKS y creadas previamente o sus propias AMI personalizadas. A fin de obtener información acerca de cada tipo de AMI optimizada para Amazon EKS, consulte uno de los siguientes temas. Para obtener instrucciones sobre cómo crear su propia AMI personalizada, consulte Script de compilación de la AMI de Amazon Linux optimizada para Amazon EKS.

Temas

- Amazon EKS dejó de ser compatible con Dockershim
- AMI de Amazon Linux optimizada para Amazon EKS
- AMI de Bottlerocket optimizadas para Amazon EKS
- AMI de Ubuntu Linux optimizada para Amazon EKS

- AMI de Windows optimizadas para Amazon EKS

Amazon EKS dejó de ser compatible con **`Dockershim`**

Kubernetes ya no es compatible con `Dockershim`. El equipo de Kubernetes eliminó el tiempo de ejecución en la versión `1.24` de Kubernetes. Para obtener más información, consulte Kubernetes is Moving on From Dockershim: Commitments and Next Steps (sigue adelante desde Dockershim: compromisos y siguientes pasos) en el Blog de Kubernetes.

Amazon EKS también dejará de ser compatible con `Dockershim` a partir del lanzamiento de la versión `1.24` de Kubernetes. Las AMI de Amazon EKS publicadas oficialmente incluyen `containerd` como único tiempo de ejecución a partir de la versión `1.24`. En este tema se describen algunos detalles, pero hay más información disponible en Todo lo que necesita saber sobre la migración a containerd en Amazon EKS.

Hay un complemento `kubectl` que puede usar para ver cuál de sus cargas de trabajo de Kubernetes están montando el volumen del socket de Docker. Para obtener más información, consulte Detector para socket Docker (DDS) en GitHub. Las AMI de Amazon EKS que ejecutan versiones de Kubernetes que son anteriores a la `1.24` utilizadas por Docker como tiempo de ejecución predeterminado. Sin embargo, estas AMI de Amazon EKS tienen una opción de marca de arranque que puede utilizar para probar sus cargas de trabajo en cualquier clúster compatible con `containerd`. Para obtener más información, consulte Prueba de la migración de Docker a `containerd`.

Seguiremos publicando AMI para las versiones de Kubernetes existentes hasta su fecha de cese de la compatibilidad. Para obtener más información, consulte Calendario de lanzamientos de Amazon EKS de Kubernetes. Si necesita más tiempo para probar sus cargas de trabajo en `containerd`, use una versión compatible anterior a la `1.24`. Pero, cuando desee actualizar las AMI oficiales de Amazon EKS a la versión `1.24` o superior, asegúrese de validar que sus cargas de trabajo se ejecutan en `containerd`.

El tiempo de ejecución de `containerd` proporciona un rendimiento y una seguridad más fiables. `containerd` es el tiempo de ejecución que se está estandarizando en Amazon EKS. Fargate y Bottlerocket ya usan solo `containerd`. `containerd` ayuda a minimizar el número de versiones de AMI de Amazon EKS necesarias para abordar las vulnerabilidades y exposiciones comunes (CVE) de `Dockershim`. Como `Dockershim` ya usa `containerd` internamente, es posible que no tenga que hacer ningún cambio. Sin embargo, hay algunas situaciones en las que es posible que se requieran cambios:

- Deberá realizar cambios en cualquier aplicación que monte el socket de Docker. Por ejemplo, se verá afectada la creación de imágenes de contenedor creadas a partir de un contenedor. Muchas herramientas de supervisión también montan el socket de Docker. Es posible que tenga que esperar las actualizaciones o volver a implementar las cargas de trabajo para la supervisión del tiempo de ejecución.
- Es posible que tenga que realizar cambios en las aplicaciones que dependan de una configuración específica de Docker. Por ejemplo, el protocolo `HTTPS_PROXY` ya no es compatible. Debe actualizar las aplicaciones que utilicen este protocolo. Para obtener más información, consulte `dockerd` en la documentación de Docker.
- Si utiliza el asistente de credenciales de Amazon ECR para extraer imágenes, tendrá que cambiar al proveedor de credenciales de imágenes de `kubelet`. Para obtener más información, consulte Configurar un proveedor de credenciales de imagen de `kubelet` en la documentación de Kubernetes.
- Dado que Amazon EKS `1.24` ya no es compatible con Docker, ya no se admiten algunos indicadores que el script de arranque de Amazon EKS admitía anteriormente. Antes de pasar a Amazon EKS `1.24` o a una versión posterior, debe eliminar cualquier referencia a los indicadores que ahora no son compatibles:
  - `--container-runtime dockerd` (`containerd` es el único valor admitido)
  - `--enable-docker-bridge`
  - `--docker-config-json`
- Si ya ha configurado Fluentd para Container Insights, debe migrar Fluentd a Fluent Bit antes de cambiar a `containerd`. Los analizadores de Fluentd están configurados para analizar únicamente los mensajes de registro en formato JSON. A diferencia de `dockerd`, el tiempo de ejecución del contenedor `containerd`contiene mensajes de registro que no están en formato JSON. Si no migra a Fluent Bit, algunos de los analizadores de Fluentd's configurados generarán una enorme cantidad de errores dentro del contenedor de Fluentd. por el número de versión compatible con Amazon EKS al que desea actualizar su clústerPara enviar registros a CloudWatch Logs, consulte Configurar Fluent Bit como DaemonSet para enviar registros a CloudWatch Logs.
- Si utiliza una AMI personalizada y va a actualizar a Amazon EKS `1.24`, debe asegurarse de que el reenvío de IP esté habilitado para sus nodos de trabajo. Esta configuración no era necesaria con Docker, pero es obligatoria para `containerd`. Es necesario para solucionar problemas de conectividad de red de Pod a Pod, Pod a externa o Pod a apiserver.

  Para comprobar esta configuración en un nodo de trabajo, ejecute uno de los siguientes comandos:

- `sysctl net.ipv4.ip_forward`
- `cat /proc/sys/net/ipv4/ip_forward`

Si el resultado es `0`, ejecute uno de los siguientes comandos para activar la variable `net.ipv4.ip_forward` del kernel:

- `sysctl -w net.ipv4.ip_forward=1`
- `echo 1 > /proc/sys/net/ipv4/ip_forward`

Para obtener información sobre la activación de la configuración en las AMI de Amazon EKS en tiempo de ejecución `containerd`, consulte `install-worker.sh` en GitHub.

AMI de Amazon Linux optimizada para Amazon EKS

La AMI de Amazon Linux optimizada para Amazon EKS, está construida sobre Amazon Linux 2 (AL2) y Amazon Linux 2023 (AL2023). Está configurada de modo que sirva de imagen base para los nodos de Amazon EKS. La AMI está configurada para funcionar con Amazon EKS e incluye los siguientes componentes:

- `kubelet`
- Autenticador de AWS IAM
- Docker (versión de Amazon EKS `1.23` y anterior)
- `containerd`

> **Note**
>
> - Puede realizar un seguimiento de los eventos de seguridad o privacidad de AL2 en el Centro de seguridad de Amazon Linux o suscribirse a la fuente RSS asociada. Los eventos de seguridad y privacidad incluyen información general del problema, qué paquetes están afectados y cómo actualizar las instancias para corregir el problema.
> - Antes de implementar una AMI acelerada o de Arm, revise la información en AMI de Amazon Linux acelerada optimizada para Amazon EKS y AMI de Amazon Linux optimizada para Amazon EKS Arm.

- Para la versión1.23 de Kubernetes, puede utilizar un indicador de arranque opcional para probar la migración de Docker a containerd. Para obtener más información, consulte Prueba de la migración de Docker a containerd.
- A partir de la versión 1.25 de Kubernetes, ya no podrá utilizar instancias P2 de Amazon EC2 con las AMI de Amazon Linux aceleradas y optimizadas de Amazon EKS listas para usar. Estas AMI para las versiones 1.25 o posteriores de Kubernetes serán compatibles con controladores de la serie NVIDIA 525 o posteriores, que son incompatibles con las instancias P2. Sin embargo, los controladores de la serie NVIDIA 525 o posteriores son compatibles con las instancias P3, P4 y P5, de modo que puede utilizarlas con las AMI para la versión 1.25 o posterior de Kubernetes. Antes de que sus clústeres de Amazon EKS se actualicen a la versión 1.25, migre todas las instancias P2 a instancias P3, P4 y P5. También debe actualizar sus aplicaciones de manera proactiva para que funcionen con la serie NVIDIA 525 o posterior. Tenemos previsto volver a portar los controladores más recientes de la serie NVIDIA 525 o posteriores a las versiones 1.23 y 1.24 de Kubernetes a finales de enero de 2024.
- A partir de la versión 1.30 o posterior, todos los grupos de nodos administrados recién creados utilizarán automáticamente AL2023 como sistema operativo de nodos de forma predeterminada. Anteriormente, los nuevos grupos de nodos utilizaban AL2 de forma predeterminada. Puede seguir utilizando AL2 si lo elige como tipo de AMI cuando crea un nuevo grupo de nodos.
- AL2 dejará de ser compatible el 30 de junio de 2025. Para obtener más información, consulte las Preguntas frecuentes de Amazon Linux 2.

Actualización de AL2 a AL2023

La AMI optimizada de Amazon EKS está disponible en dos familias con base en AL2 y AL2023. AL2023 es un nuevo sistema operativo basado en Linux diseñado para proporcionar un entorno seguro, estable y de alto rendimiento para las aplicaciones en la nube. Es la próxima generación de Amazon Linux de Amazon Web Services y está disponible en todas las versiones compatibles de Amazon EKS, incluidas las versiones 1.23 y 1.24 con soporte ampliado. Las AMI aceleradas de Amazon EKS basadas en AL2023 estarán disponibles en una fecha posterior. Si tiene cargas de trabajo aceleradas, debe seguir utilizando la AMI acelerada de AL2 o Bottlerocket.

AL2023 ofrece varias mejoras con respecto al AL2. Para obtener una comparación completa, consulte Comparación de AL2 y Amazon Linux 2023 en la Guía del usuario de Amazon Linux 2023.

Se han añadido, actualizado y eliminado varios paquetes de AL2. Se recomienda encarecidamente probar las aplicaciones con AL2023 antes de realizar la actualización. Para ver una lista de todos los cambios de paquetes en AL2023, consulte Cambios de paquetes en Amazon Linux 2023 en las Notas de la versión de Amazon Linux 2023.

Además de estos cambios, debe tener en cuenta lo siguiente:

- AL2023 presenta un nuevo proceso de inicialización de nodos `nodeadm` que utiliza un esquema de configuración YAML. Si utiliza grupos de nodos autoadministrados o una AMI con una plantilla de lanzamiento, ahora tendrá que proporcionar metadatos del clúster adicionales de forma explícita cuando cree un nuevo grupo de nodos. A continuación, se muestra un ejemplo de los parámetros mínimos necesarios, en los que ahora se necesitan `apiServerEndpoint`, `certificateAuthority` y el servicio de `cidr`:

  ```
  ---
  apiVersion: node.eks.aws/v1alpha1
  kind: NodeConfig
  spec:
    cluster:
      name: my-cluster
      apiServerEndpoint: https://example.com
      certificateAuthority: Y2VydGlmaWNhdGVBdXRob3JpdHk=
      cidr: 10.100.0.0/16
  ```

  En AL2, los metadatos de estos parámetros se descubrieron a partir de la llamada a la API `DescribeCluster` de Amazon EKS. Con AL2023, este comportamiento ha cambiado, ya que la llamada a la API adicional corre el riesgo de limitarse durante los escalados verticales de nodos a gran escala. Este cambio no le afecta si utiliza grupos de nodos administrados sin una plantilla de lanzamiento o si utiliza Karpenter. Para obtener más información sobre `certificateAuthority` y el servicio de `cidr`, consulte `DescribeCluster` en la Referencia de la API de Amazon EKS.
- Docker no es compatible con AL2023 para todas las versiones compatibles de Amazon EKS. La compatibilidad con Docker finalizó y se eliminó con la versión `1.24` o posterior de Amazon EKS en AL2. Para obtener más información sobre la obsolescencia, consulte Amazon EKS ya no es compatible con `Dockershim`.
- Se requiere la versión `1.16.2` o posterior de CNI de Amazon VPC para AL2023.
- De forma predeterminada, AL2023 requiere `IMDSv2`. `IMDSv2` tiene varios beneficios que ayudan a mejorar la postura de seguridad. Utiliza un método de autenticación orientado a la sesión que requiere la creación de un token secreto en una solicitud sencilla de HTTP PUT para iniciar la

sesión. El tiempo de validez de un token de sesión puede oscilar entre 1 segundo y 6 horas. Para obtener más información sobre cómo realizar la transición de `IMDSv1` a `IMDSv2`, consulte Transición a la versión 2 del servicio de metadatos de instancias y Cómo aprovechar todos los beneficios de IMDSv2 e inhabilitar IMDSv1 en toda la infraestructura de AWS. Si desea utilizar `IMDSv1`, puede hacerlo si anula de manera manual la configuración mediante las propiedades de inicio de la opción de metadatos de la instancia.

> **Note**
>
> Para `IMDSv2`, el recuento de saltos predeterminado para los grupos de nodos administrados se establece en 1. Esto significa que los contenedores no tendrán acceso a las credenciales del nodo mediante IMDS. Si necesita acceso del contenedor a las credenciales del nodo, debe anular `HttpPutResponseHopLimit` de manera manual en una plantilla de lanzamiento personalizada de Amazon EC2 y aumentarlo a 2. Como alternativa, puede utilizar Pod Identity de Amazon EKS para proporcionar credenciales en lugar de `IMDSv2`.

- AL2023 presenta la siguiente generación de jerarquías de grupos de control unificados (`cgroupv2`). `cgroupv2` se utiliza para implementar un tiempo de ejecución de contenedores y por `systemd`. Si bien AL2023 sigue incluyendo un código que puede hacer que el sistema funcione con `cgroupv1`, esta configuración no se recomienda ni se admite. Esta configuración se eliminará por completo en una futura versión importante de Amazon Linux.
- Se requiere una versión de `eksctl` `0.176.0` o superior para que `eksctl` sea compatible con AL2023.

En el caso de los grupos de nodos administrados que existían con anterioridad, puede realizar una actualización local o una actualización azul/verde, según cómo utilice la plantilla de lanzamiento:

- Si utiliza una AMI personalizada con un grupo de nodos administrado, puede realizar una actualización local si intercambia el ID de la AMI en la plantilla de lanzamiento. Debe asegurarse de que las aplicaciones y cualquier dato de usuario se transfieran primero a AL2023 antes de llevar a cabo esta estrategia de actualización.
- Si utiliza grupos de nodos administrados con la plantilla de lanzamiento estándar o con una plantilla de lanzamiento personalizada que no especifica el ID de la AMI, deberá actualizar mediante una estrategia azul/verde. Una actualización azul/verde suele ser más compleja e implica la creación de un grupo de nodos completamente nuevo en el que se especificará AL2023

como tipo de AMI. Luego, será necesario configurar con cuidado el nuevo grupo de nodos para garantizar que todos los datos personalizados del grupo de nodos de AL2 sean compatibles con el nuevo sistema operativo. Una vez que el nuevo grupo de nodos se haya probado y validado con sus aplicaciones, podrá migrar Pods del grupo de nodos anterior al nuevo grupo de nodos. Una vez completada la migración, puede eliminar el grupo de nodos anterior.

Si utiliza Karpenter y quiere utilizar AL2023, deberá modificar el campo `EC2NodeClass` `amiFamily` con AL2023. De forma predeterminada, la desviación está habilitada en Karpenter. Esto significa que, una vez que se haya cambiado el campo `amiFamily`, Karpenter actualizará automáticamente los nodos de trabajo a la AMI más reciente cuando esté disponible.

AMI de Amazon Linux acelerada optimizada para Amazon EKS

> **Note**
>
> Las AMI aceleradas de Amazon EKS basadas en AL2023 estarán disponibles en una fecha posterior. Si tiene cargas de trabajo aceleradas, debe seguir utilizando la AMI acelerada de AL2 o Bottlerocket.

La AMI de Amazon Linux acelerada optimizada para Amazon EKS se basa en la AMI de Amazon Linux optimizada estándar de Amazon EKS. Está configurada de modo que sirva como imagen opcional para los nodos de Amazon EKS a fin de admitir GPU y cargas de trabajo basadas en Inferentia y Trainium.

Además de la configuración de la AMI optimizada para Amazon EKS estándar, la AMI acelerada incluye lo siguiente:

- Controladores NVIDIA
- `nvidia-container-runtime`
- Controlador AWS Neuron

Para obtener una lista de los componentes más recientes incluidos en la AMI acelerada, consulte las versiones `amazon-eks-ami` en GitHub.

> **Note**
>
> - La AMI acelerada optimizada para Amazon EKS solo admite tipos de instancias basadas en GPU e Inferentia. Asegúrese de especificar estos tipos de instancia en la plantilla de nodos de AWS CloudFormation. Al utilizar la AMI acelerada optimizada para Amazon EKS, acepta el acuerdo de licencia de usuario (EULA) de NVIDIA.
> - La AMI acelerada optimizada para Amazon EKS se denominaba anteriormente AMI optimizada para Amazon EKS compatible con GPU.
> - Las versiones anteriores de la AMI acelerada optimizada para Amazon EKS instalaron el repositorio `nvidia-docker`. El repositorio ya no se incluye en la versión `v20200529` de la AMI de Amazon EKS y versiones posteriores.

Para habilitar AWS cargas de trabajo basadas en Neuron (acelerador de aprendizaje automático)

Para obtener información detallada sobre las cargas de trabajo de formación e inferencia que utilizan Neuron en Amazon EKS, consulte las siguientes referencias:

- Contenedores - Kubernetes - Introducción a la documentación de AWS Neuron
- Formación en muestras de AWS Neuron EKS en GitHub
- Inferencia de machine learning mediante el uso de AWS Inferentia

Habilitar cargas de trabajo basadas en GPU

El siguiente procedimiento describe cómo ejecutar una carga de trabajo en una instancia basada en GPU con la AMI acelerada optimizada para Amazon EKS.

1. Una vez que los nodos de GPU estén unidos al clúster, debe aplicar el complemento de dispositivos NVIDIA para Kubernetes como un DaemonSet en su clúster. Reemplace *vX.X.X* con la versión Plugin de dispositivo NVidia/K8Sdeseada antes de ejecutar el siguiente comando.

   ```
   kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/
   nvidia-device-plugin.yml
   ```

2. Puede verificar que los nodos tienen GPU asignables con el siguiente comando.

```
kubectl get nodes "-o=custom-
columns=NAME:.metadata.name,GPU:.status.allocatable.nvidia\.com/gpu"
```

Implementar un Pod con el fin de probar que los nodos de la GPU están configurados correctamente

1. Cree un archivo denominado `nvidia-smi.yaml` con el siguiente contenido. Reemplace *tag* con la etiqueta deseada para `nvidia/cuda`. Este manifiesto lanza un contenedor de NVIDIA CUDA que ejecuta `nvidia-smi` en un nodo de trabajo.

```
apiVersion: v1
kind: Pod
metadata:
  name: nvidia-smi
spec:
  restartPolicy: OnFailure
  containers:
  - name: nvidia-smi
    image: nvidia/cuda:tag
    args:
    - "nvidia-smi"
    resources:
      limits:
        nvidia.com/gpu: 1
```

2. Aplique el manifiesto con el siguiente comando.

```
kubectl apply -f nvidia-smi.yaml
```

3. Una vez que el Pod termine de ejecutarse, consulte sus registros con el siguiente comando.

```
kubectl logs nvidia-smi
```

Un ejemplo de salida sería el siguiente.

```
Mon Aug  6 20:23:31 20XX
+-----------------------------------------------------------------------------+
| NVIDIA-SMI XXX.XX                 Driver Version: XXX.XX                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
```

```
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:00:1C.0 Off |                    0 |
| N/A   46C    P0    47W / 300W |      0MiB / 16160MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

AMI de Amazon Linux optimizada para Amazon EKS Arm

Las instancias Arm ofrecen un importante ahorro de costos para aplicaciones de escalado horizontal y aplicaciones basadas en Arm, como servidores web, microservicios en contenedores, flotas de almacenamiento en caché y almacenes de datos distribuidos. Al agregar nodos de Arm al clúster, tenga en cuenta las siguientes consideraciones.

Consideraciones

- Si el clúster se implementó antes del 17 de agosto de 2020, debe realizar una actualización única de los manifiestos complementarios de clúster críticos. Esto es para que Kubernetes pueda extraer la imagen correcta para cada arquitectura de hardware que se utilice en el clúster. Para obtener más información acerca de la actualización de complementos de clúster, consulte Actualice la versión de Kubernetes de un clúster de Amazon EKS. Si implementó el clúster a partir del 17 de agosto de 2020, entonces sus complementos CoreDNS, `kube-proxy` y Amazon VPC CNI plugin for Kubernetes ya son aptos en múltiples arquitecturas.
- Las aplicaciones implementadas en los nodos de Arm deben compilarse para Arm.
- Si tiene algún DaemonSets implementado en un clúster existente o desea implementarlos en un clúster nuevo en el que también quiera implementar nodos de Arm, compruebe que el DaemonSet se pueda ejecutar en todas las arquitecturas de hardware del clúster.
- Puede ejecutar grupos de nodos de Arm y grupos de nodos x86 en el mismo clúster. Si lo hace, considere la posibilidad de implementar imágenes de contenedor de varias arquitecturas en un repositorio de contenedores como Amazon Elastic Container Registry y, a continuación, agregar selectores de nodo a los manifiestos para que Kubernetes sepa en qué arquitectura de hardware se puede implementar un Pod. Para obtener más información, consulte Introducir una imagen de

varias arquitecturas en la Guía del usuario de Amazon ECR y en la publicación del blog Introducing multi-architecture container images for Amazon ECR.

Prueba de la migración de Docker a `containerd`

Amazon EKS dejó de estar disponible para Docker a partir del lanzamiento de la versión `1.24` de Kubernetes. Para obtener más información, consulte Amazon EKS dejó de ser compatible con `Dockershim`.

Para la versión `1.23` de Kubernetes, puede utilizar una marca de arranque opcional para habilitar el tiempo de ejecución `containerd` de las AMI de AL2 optimizadas para Amazon EKS. Esta característica le otorga una ruta clara para migrar a `containerd` al actualizar a una versión `1.24` o posterior. Amazon EKS dejó de estar disponible para Docker a partir del lanzamiento de la versión `1.24` de Kubernetes. El tiempo de ejecución `containerd` ha sido ampliamente adoptado en la comunidad de Kubernetes y es un proyecto graduado con la CNCF. Puede probarlo al agregarle un grupo de nodos a un clúster nuevo o existente.

Puede habilitar el indicador del arranque al crear uno de los siguientes tipos de grupos de nodos.

Autoadministrado

Cree el grupo de nodos con las instrucciones en Lanzar nodos autoadministrados de Amazon Linux. Especifique una AMI optimizada para Amazon EKS y el siguiente texto para el parámetro `BootstrapArguments`.

```
--container-runtime containerd
```

Administrado

Si utiliza `eksctl`, cree un archivo llamado *`my-nodegroup`*`.yaml` con el siguiente contenido. Reemplace cada *`example value`* con valores propios. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Para recuperar un ID de AMI optimizada para `ami-`*`1234567890abcdef0`*, consulte Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: my-cluster
```

```
  region: region-code
  version: 1.23
managedNodeGroups:
  - name: my-nodegroup
    ami: ami-1234567890abcdef0
    overrideBootstrapCommand: |
      #!/bin/bash
      /etc/eks/bootstrap.sh my-cluster --container-runtime containerd
```

> **Note**
>
> Si lanza muchos nodos al mismo tiempo, es posible que desee especificar también valores para los argumentos `--apiserver-endpoint`, `--b64-cluster-ca` y `--dns-cluster-ip` de arranque para evitar errores. Para obtener más información, consulte Especificación de una AMI.

Ejecute el siguiente comando para crear el grupo de nodos.

```
eksctl create nodegroup -f my-nodegroup.yaml
```

Si prefiere utilizar una herramienta diferente para crear el grupo de nodos administrado, debe implementar el grupo de nodos mediante una plantilla de lanzamiento. En la plantilla de lanzamiento, especifique un ID de AMI optimizada para Amazon EKS y, a continuación, implemente el grupo de nodos mediante una plantilla de lanzamiento y proporcione los siguientes datos de usuario. Estos datos de usuario pasan los argumentos en el archivo `bootstrap.sh`. Para obtener más información acerca del archivo del proceso de arranque, consulte bootstrap.sh en GitHub.

```
/etc/eks/bootstrap.sh my-cluster --container-runtime containerd
```

Más información

Para obtener más información sobre el uso de las AMI de Amazon Linux optimizadas para Amazon EKS, consulte las siguientes secciones:

- Para usar Amazon Linux con grupos de nodos administrados, consulte Grupos de nodos administrados.

- Para lanzar nodos autoadministrados de Amazon Linux, consulte Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS.
- Para obtener información sobre la versión, consulte Versiones de la AMI de Amazon Linux optimizada para Amazon EKS.
- Para recuperar los identificadores más recientes de las AMI de Amazon Linux optimizadas para Amazon EKS, consulte Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS.
- Para los scripts de código abierto que se utilizan para crear la AMI optimizada para Amazon EKS, consulte Script de compilación de la AMI de Amazon Linux optimizada para Amazon EKS.

Versiones de la AMI de Amazon Linux optimizada para Amazon EKS

Las AMI de Amazon Linux optimizados para Amazon EKS están versionadas por la versión de Kubernetes y la fecha de lanzamiento de la AMI en el siguiente formato:

```
k8s_major_version.k8s_minor_version.k8s_patch_version-release_date
```

Cada versión de AMI incluye varias versiones de `kubelet`, Docker, el kernel de Linux y `containerd`. La AMI acelerada también incluye varias versiones del controlador de NVIDIA. Puede encontrar la información de esta versión en Registro de cambios en GitHub.

Recuperación de los ID de la AMI de Amazon Linux optimizada para Amazon EKS

Puede recuperar el ID de Amazon Machine Image (AMI) de las AMI optimizadas para Amazon EKS mediante programación al consultar la API de Parameter Store de AWS Systems Manager. Este parámetro elimina la necesidad de buscar de manera manual los ID de la AMI optimizada para Amazon EKS. Para obtener más información acerca de la API de Systems Manager Parameter Store, consulte GetParameter.

Para recuperar un ID de AMI para las AMI optimizadas para Amazon EKS con el AWS CLI

1. Determine la región en la que se implementará la instancia de nodo, como por ejemplo, `us-west-2`.
2. Determine el tipo de AMI que necesita. Para más información sobre los tipos de instancias de Amazon EC2, consulte Tipos de instancias.

   - `amazon-linux-2` es para instancias basadas en Amazon Linux 2 (AL2) `x86`.

- `amazon-linux-2-arm64` es para las instancias ARM AL2, como las instancias basadas en AWSGraviton.
- `amazon-linux-2-gpu` es para las instancias aceleradas por GPU de AL2.
- `amazon-linux-2023/x86_64/standard` es para instancias basadas en Amazon Linux 2023 (AL2023) `x86`.
- `amazon-linux-2023/arm64/standard` es para instancias ARM AL2023.

3. Determine la versión de Kubernetes del clúster a la que se conectará su nodo, por ejemplo, la 1.30.
4. Ejecute el comando AWS CLI para recuperar el ID de AMI correspondiente. Sustituya la Región de AWS, la versión de Kubernetes y la plataforma según corresponda. Debe estar registrado en la AWS CLI con una entidad principal de IAM que utiliza el permiso de IAM `ssm:GetParameter` para recuperar los metadatos AMI optimizados de Amazon EKS.

```
aws ssm get-parameter --name /aws/service/eks/optimized-ami/1.30/amazon-linux-2/
recommended/image_id \
                --region region-code --query "Parameter.Value" --output text
```

Un ejemplo de salida sería el siguiente.

```
ami-1234567890abcdef0
```

Script de compilación de la AMI de Amazon Linux optimizada para Amazon EKS

Amazon Elastic Kubernetes Service (Amazon EKS) tiene scripts de código abierto que se utilizan para crear la AMI optimizada para Amazon EKS. Estos scripts de compilación están disponibles en GitHub.

La AMI de Amazon Linux optimizada para Amazon EKS se crea sobre Amazon Linux 2 (AL2) y Amazon Linux 2023 (AL2023), específicamente para su uso como un nodo en clústeres de Amazon EKS. Puede utilizar este repositorio a fin de ver los detalles de cómo el equipo de Amazon EKS configura `kubelet`, Docker, el autenticador de IAM de AWS para Kubernetes, y construir su propia AMI basada en Amazon Linux desde cero.

El repositorio de scripts de compilación incluye una plantilla HashiCorp packer y crea scripts para generar una AMI. Estos scripts son el origen de confianza para las compilaciones de la AMI optimizada para Amazon EKS, de modo que pueda seguir el repositorio de GitHub para monitorear

los cambios en nuestras AMI. Por ejemplo, quizás desee su propia AMI para utilizar la misma versión de Docker que el equipo de Amazon EKS utiliza para la AMI oficial.

El repositorio de GitHub también contiene el script de arranque especializado y el script de nodeadm, que se ejecuta en el momento del arranque para configurar los datos de certificado de la instancia, el punto de conexión del plano de control, el nombre del clúster, etcétera.

Además, el repositorio de GitHub contiene nuestras plantillas de nodo de Amazon EKS de AWS CloudFormation. Estas plantillas facilitan la puesta en marcha de una instancia que ejecuta la AMI optimizada para Amazon EKS y la registra con un clúster.

Para obtener más información, consulte los repositorios en GitHub en https://github.com/awslabs/amazon-eks-ami.

La AMI de AL2 optimizada para Amazon EKS contiene un indicador de proceso de arranque opcional con el fin de habilitar el tiempo de ejecución de `containerd`.

Configuración de VT1 para la AMI de Amazon Linux personalizada

Las AMI personalizadas de Amazon Linux en Amazon EKS pueden admitir la familia de instancias de transcodificación de vídeo VT1 para Amazon Linux 2 (AL2), Ubuntu 18 y Ubuntu 20. VT1 admite las tarjetas de transcodificación de medios Xilinx U30 con códecs H.264/AVC y H.265/HEVC acelerados. Para obtener el beneficio de estas instancias aceleradas, debe seguir estos pasos:

1. Cree y lance una AMI básica desde AL2, Ubuntu 18 o Ubuntu 20.
2. Después de lanzar la AMI base, instale el controlador XRT y el tiempo de ejecución en el nodo.
3. Creación de un clúster de Amazon EKS.
4. Instale el complemento FPGA de Kubernetes en el clúster.

```
kubectl apply -f fpga-device-plugin.yml
```

El complemento anunciará los dispositivos Xilinx U30 por nodo en el clúster de Amazon EKS. Puede utilizar la imagen de Docker FFMPEG para ejecutar cargas de trabajo de transcodificación de vídeo de ejemplo en su clúster de Amazon EKS.

Configuración de DL1 para la AMI personalizada de Amazon Linux 2

Las AMI personalizadas de Amazon Linux 2 (AL2) en Amazon EKS pueden admitir cargas de trabajo de aprendizaje profundo a escala mediante configuración adicional y complementos de Kubernetes.

En este documento se describen los componentes necesarios para configurar una solución de Kubernetes genérica para una configuración en las instalaciones o como base de una configuración de nube más grande. Para admitir esta función, deberá realizar los siguientes pasos en su entorno personalizado:

- Controladores de SynapaseAI® Software cargados en el sistema: se incluyen en las AMI disponibles en GitHub.
- El complemento para dispositivos Habana: un DaemonSet que le permite habilitar automáticamente el registro de dispositivos Habana en el clúster de Kubernetes y realizar un seguimiento del estado del dispositivo.
- Helm 3.x
- Gráfico de Helm para instalar el operador de MPI.
- Operador de MPI

1. Cree y lance una AMI básica desde AL2, Ubuntu 18 o Ubuntu 20.
2. Siga estas instrucciones para configurar el entorno para DL1.

AMI de Bottlerocket optimizadas para Amazon EKS

Bottlerocket es una distribución de Linux de código abierto patrocinada y respaldada por AWS. Bottlerocket está diseñado específicamente para alojar cargas de trabajo de contenedores. Con Bottlerocket, puede mejorar la disponibilidad de las implementaciones en contenedores y reducir los costos operativos mediante la automatización de las actualizaciones de la infraestructura de contenedores. Bottlerocket incluye solo el software esencial para ejecutar los contenedores, lo que mejora el uso de los recursos, reduce las amenazas a la seguridad y reduce los gastos de administración. La AMI de Bottlerocket incluye `containerd`, `kubelet` y el Autenticador de AWS IAM. Además de los grupos de nodos administrados y los nodos autoadministrados, Bottlerocket también es compatible con Karpenter.

Ventajas

El uso de Bottlerocket con el clúster de Amazon EKS tiene las siguientes ventajas:

- Mayor tiempo de actividad con un menor costo operativo y una menor complejidad de administración: Bottlerocket ocupa menos recursos, tiempos de arranque más cortos y es menos vulnerable a las amenazas de seguridad que otras distribuciones de Linux. Una huella

más pequeña de Bottlerocket's ayuda a reducir los costos al utilizar menos recursos de almacenamiento, cómputos y redes.

- Seguridad mejorada gracias a las actualizaciones automáticas del sistema operativo: las actualizaciones de Bottlerocket se aplican como una sola unidad y se pueden anular si es necesario. Esto elimina el riesgo de actualizaciones dañadas o fallidas que pueden dejar el sistema inutilizable. Con Bottlerocket, las actualizaciones de seguridad se pueden aplicar automáticamente tan pronto como estén disponibles de manera mínimamente disruptiva, y revertirse si se producen errores.
- Soporte premium: las versiones proporcionadas por AWS de Bottlerocket en Amazon EC2 están cubiertas por los mismos planes de AWS Support, que también cubren servicios de AWS como Amazon EC2, Amazon EKS y Amazon ECR.

Consideraciones

Tenga en cuenta lo siguiente cuando utilice Bottlerocket para su tipo de AMI:

- Bottlerocket admite instancias de Amazon EC2 con procesadores `x86_64` y `arm64`. No se recomienda utilizar la AMI Bottlerocket con instancias de Amazon EC2 con un chip Inferentia.
- Actualmente, no hay ninguna plantilla de CloudFormation de AWS con la que se puedan implementar nodos de Bottlerocket.
- Las imágenes de Bottlerocket no incluyen un servidor SSH ni un shell. Puede utilizar métodos de acceso fuera de banda para permitir SSH. Estos métodos permiten el contenedor de administrador y superar algunos pasos de configuración de arranque con datos de usuario. Para obtener más información, consulte las siguientes secciones en OS de Bottlerocket en GitHub:
  - Exploration (Exploración)
  - Contenedor de administrador
  - Configuración de Kubernetes
- Bottlerocket utiliza diferentes tipos de contenedores:
  - De forma predeterminada, se habilita un contenedor de control. Este contenedor ejecuta el agente de AWS Systems Manager que puede utilizar para ejecutar comandos o iniciar sesiones de shell en instancias Bottlerocket de Amazon EC2. Para obtener más información, consulte Configuración del administrador de sesiones en la Guía del usuario de AWS Systems Manager.
  - Si se proporciona una clave SSH al crear el grupo de nodos, se habilita un contenedor de administración. Recomendamos utilizar el contenedor de administración solo para escenarios de

desarrollo y pruebas. No recomendamos utilizarlo en entornos de producción. Para obtener más información, consulte Contenedor de administración en GitHub.

Más información

Para obtener más información sobre el uso de las AMI de Bottlerocket optimizadas para Amazon EKS, consulte las siguientes secciones:

- Para obtener más información acerca de Bottlerocket, consulte la documentación y las versiones en GitHub.
- Para usar Bottlerocket con grupos de nodos administrados, consulte Grupos de nodos administrados.
- Para lanzar nodos de Bottlerocket autoadministrados, consulte Lanzamiento de nodos de Bottlerocket autoadministrados.
- Para recuperar los identificadores más recientes de las AMI de Bottlerocket optimizadas para Amazon EKS, consulte Recuperación de los ID de la AMI de Bottlerocket optimizada para Amazon EKS.
- Para obtener más información sobre el soporte de cumplimiento, consulte soporte de cumplimiento de Bottlerocket.

Recuperación de los ID de la AMI de Bottlerocket optimizada para Amazon EKS

Puede recuperar el ID de Amazon Machine Image (AMI) de las AMI optimizadas para Amazon EKS al consultar la API de Parameter Store de AWS Systems Manager. Al utilizar este parámetro, no necesita buscar de manera manual los ID de la AMI optimizada para Amazon EKS. Para obtener más información acerca de la API de Systems Manager Parameter Store, consulte GetParameter. La entidad principal de IAM que utiliza debe tener el permiso `ssm:GetParameter` de IAM para recuperar los metadatos de la AMI optimizada para Amazon EKS.

Puede recuperar el ID de imagen de la última AMI de Bottlerocket optimizada para Amazon EKS recomendada con el siguiente comando AWS CLI mediante el parámetro secundario `image_id`. Reemplace *`1.30`* con una versión compatible y *`region-code`* con una región compatible con Amazon EKS para la que desea el ID de la AMI.

```
aws ssm get-parameter --name /aws/service/bottlerocket/aws-k8s-1.30/x86_64/latest/
image_id --region region-code --query "Parameter.Value" --output text
```

Un ejemplo de salida sería el siguiente.

```
ami-1234567890abcdef0
```

soporte de cumplimiento de Bottlerocket

Bottlerocket cumple con las recomendaciones definidas por varias organizaciones:

- Hay un punto de referencia del CIS definido por Bottlerocket. En una configuración predeterminada, la imagen de Bottlerocket tiene la mayoría de los controles requeridos por el perfil de configuración de nivel 1 del CIS. Puede implementar los controles requeridos para un perfil de configuración de nivel 2 de CIS. Para obtener más información, consulte Validar la AMI de Bottlerocketoptimizada de Amazon EKS con el punto de referencia del CIS en el blog de AWS.
- El conjunto de funciones optimizado y la reducción de la superficie de ataque significan que las instancias de Bottlerocket requieren menos configuración para cumplir con los requisitos de PCI DSS. El Punto de referencia de CIS para Bottlerocket es un recurso excelente para reforzar las directrices y cumple con los requisitos de estándares de configuración segura según el requisito 2.2 de PCI DSS. También puede aprovechar Fluent Bit para cumplir con sus requisitos de registro de auditorías a nivel de sistema operativo según el requisito 10.2 de PCI DSS. AWS publica instancias nuevas (con parches) de Bottlerocket periódicamente para ayudarlo a cumplir con los requisitos 6.2 de PCI DSS (para la versión 3.2.1) y el requisito 6.3.3 (para la versión 4.0).
- Bottlerocket es una función que cumple los requisitos de la HIPAA y está autorizada para su uso con cargas de trabajo reguladas tanto para Amazon EC2 como para Amazon EKS. Para obtener más información, consulte el documento Diseño de la seguridad y el cumplimiento de HIPAA en Amazon EKS.

AMI de Ubuntu Linux optimizada para Amazon EKS

Canonical se ha asociado con Amazon EKS para crear AMI de nodo que puede utilizar en sus clústeres.

Canonical distribuye una imagen de sistema operativo de nodo de Kubernetes personalizada. Esta imagen de Ubuntu minimizada está optimizada para Amazon EKS e incluye el kernel de AWS personalizado desarrollado conjuntamente con AWS. Para obtener más información, consulte Ubuntu en Amazon Elastic Kubernetes Service (EKS) y Lanzamiento de nodos de Ubuntu autoadministrados. Para obtener información sobre el Support, consulte la sección de software de terceros de las Preguntas frecuentes sobre el soporte premium de AWS.

AMI de Windows optimizadas para Amazon EKS

Las AMI de Windows optimizadas para Amazon EKS se basan en Windows Server 2019 y Windows Server 2022. Están configuradas de modo que sirvan de imagen base para los nodos de Amazon EKS. De forma predeterminada, las AMI incluyen los siguientes componentes:

- `kubelet`
- `kube-proxy`
- Autenticador de AWS IAM para Kubernetes
- `csi-proxy`
- `containerd`

> **Note**
>
> Puede realizar un seguimiento de eventos de seguridad o privacidad para Windows Server con la Guía de actualización de seguridad de Microsoft.

Amazon EKS ofrece las siguientes variantes de AMI optimizadas para contenedores de Windows:

- AMI de Windows Server 2019 Core optimizada para Amazon EKS
- AMI de Windows Server 2019 Full optimizada para Amazon EKS
- AMI de Windows Server 2022 Core optimizada para Amazon EKS
- AMI de Windows Server 2022 Full optimizada para Amazon EKS

> **Important**
>
> - La AMI de Windows Server 20H2 Core optimizada para Amazon EKS ha quedado obsoleta. No se lanzarán nuevas versiones de esta AMI.
> - Para asegurarse de que dispone de las actualizaciones de seguridad más recientes de forma predeterminada, Amazon EKS mantiene optimizada las AMI de Windows durante los últimos cuatro meses. Cada nueva AMI estará disponible durante cuatro meses a partir del momento de su lanzamiento inicial. Después de este período, las AMI más antiguas pasan a ser privadas y ya no se podrá acceder a ellas. Recomendamos utilizar las AMI

más recientes para evitar vulnerabilidades de seguridad y perder el acceso a las AMI más antiguas que hayan llegado al final de su vida útil. Si bien no podemos garantizar el acceso a las AMI que se han convertido en privadas, puede solicitar el acceso presentando un ticket a AWS Support.

Calendario de versiones

En la siguiente tabla se enumeran las fechas de lanzamiento y finalización de la compatibilidad para las versiones de Windows de Amazon EKS. Si una fecha de finalización está en blanco, significa que la versión aún es compatible.

Versión de Windows	Versión de Amazon EKS	Fin de la compatibilidad de Amazon EKS
Windows Server 2022 Core	10/17/2022	
Windows Server 2022 Full	10/17/2022	
Windows Server 20H2 Core	8/12/2021	8/9/2022
Windows Server 2004 Core	8/19/2020	12/14/2021
Windows Server 2019 Core	10/7/2019	
Windows Server 2019 Full	10/7/2019	
Windows Server 1909 Core	10/7/2019	12/8/2020

Parámetros de configuración del script de arranque

Al crear un nodo de Windows, hay un script en el nodo que permite la configuración de diferentes parámetros. Según la configuración, este script se puede encontrar en el nodo en una ubicación similar a: `C:\Program Files\Amazon\EKS\Start-EKSBootstrap.ps1`. Para especificar valores de parámetros personalizados, puede especificarlos como argumentos del script de arranque. Por ejemplo, puede actualizar los datos de usuario en la plantilla de lanzamiento. Para obtener más información, consulte Datos de usuario de Amazon EC2.

El script incluye los siguientes parámetros de la línea de comandos:

- `-EKSClusterName`: especifica el nombre del clúster de Amazon EKS para que este nodo de trabajo se va a unir.
- `-KubeletExtraArgs`: especifica argumentos adicionales para `kubelet` (opcional).
- `-KubeProxyExtraArgs`: especifica argumentos adicionales para `kube-proxy` (opcional).
- `-APIServerEndpoint`: especifica el punto de conexión del servidor de API del clúster de Amazon EKS (opcional). Solo válido cuando se utiliza con `-Base64ClusterCA`. Se omite llamando a `Get-EKSCluster`.
- `-Base64ClusterCA`: especifica el contenido de CA de clúster codificado en base64 (opcional). Solo válido cuando se utiliza con `-APIServerEndpoint`. Se omite llamando a `Get-EKSCluster`.
- `-DNSClusterIP`: sustituye la dirección IP que se va a utilizar para las consultas DNS dentro del clúster (opcional). El valor predeterminado es `10.100.0.10` o `172.20.0.10` en función de la dirección IP de la interfaz principal.
- `-ServiceCIDR`: anula el rango de direcciones IP del servicio de Kubernetes desde el que se direccionan los servicios del clúster. El valor predeterminado es `172.20.0.0/16` o `10.100.0.0/16` en función de la dirección IP de la interfaz principal.
- `-ExcludedSnatCIDRs`: lista de los CIDR de `IPv4` que se deben excluir de la traducción de direcciones de red de origen (SNAT). Esto significa que la IP privada del pod, que es direccionable por VPC, no se traduciría a la dirección IP de la dirección `IPv4` principal de la ENI de la instancia para el tráfico saliente. De forma predeterminada, se agrega el CIDR de `IPv4` de la VPC para el nodo de Windows de Amazon EKS. Al especificar los CIDR en este parámetro también se excluyen los CIDR especificados. Para obtener más información, consulte SNAT para Pods.

Además de los parámetros de la línea de comandos, también puede especificar algunos parámetros de variables de entorno. Cuando se especifica un parámetro de la línea de comandos, tiene prioridad sobre la variable de entorno correspondiente. Las variables de entorno deben definirse con la máquina (o sistema) como ámbito, ya que el script de arranque solo leerá variables cuyo ámbito sea la máquina.

El script tiene en cuenta las siguientes variables de entorno:

- `SERVICE_IPV4_CIDR`: consulte el parámetro de la línea de comandos `ServiceCIDR` para ver la definición.
- `EXCLUDED_SNAT_CIDRS`: debe ser una cadena separada por comas. Consulte el parámetro de la línea de comandos `ExcludedSnatCIDRs` para ver la definición.

Lance nodos de Windows Server 2022 autoadministrados con **eksctl**

Puede usar el siguiente **test-windows-2022.**yaml como referencia para ejecutar Windows Server 2022 como nodos autoadministrados. Reemplace cada *example value* con valores propios.

> **Note**
>
> Debe usar eksctl versión 0.116.0 o posterior para ejecutar nodos de Windows Server 2022 autoadministrados.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: windows-2022-cluster
  region: region-code
  version: '1.30'

nodeGroups:
  - name: windows-ng
    instanceType: m5.2xlarge
    amiFamily: WindowsServer2022FullContainer
    volumeSize: 100
    minSize: 2
    maxSize: 3
  - name: linux-ng
    amiFamily: AmazonLinux2
    minSize: 2
    maxSize: 3
```

Los grupos de nodos se pueden crear con el siguiente comando.

```
eksctl create cluster -f test-windows-2022.yaml
```

Soporte de autenticación de gMSA

Los Pods de Windows de Amazon EKS permiten diferentes tipos de autenticación de cuenta de servicio administrada por grupo (gMSA).

- Amazon EKS admite identidades de dominio Active Directory para la autenticación. Para obtener más información sobre gMSA unido a un dominio, consulte Autenticación de Windows en los pods de Windows de Amazon EKS en el blog de AWS.
- Amazon EKS ofrece un complemento que permite que los nodos de Windows que no están unidos a un dominio recuperen credenciales de gMSA con una identidad de usuario portátil. Para obtener más información sobre gMSA sin dominio, consulte Autenticación de Windows sin dominio para los pods de Windows para Amazon EKS en el blog de AWS.

Imágenes de contenedor en caché

Las AMI de Windows optimizadas para Amazon EKS poseen algunas imágenes de contenedor en caché para los tiempos de ejecución de `containerd`. Las imágenes de los contenedores se almacenan en caché al crear AMI personalizadas mediante componentes de compilación administrados por Amazon. Para obtener más información, consulte Uso del componente de compilación administrado por Amazon.

Las siguientes imágenes de contenedores en caché son para el tiempo de ejecución de `containerd`:

- `amazonaws.com/eks/pause-windows`
- `mcr.microsoft.com/windows/nanoserver`
- `mcr.microsoft.com/windows/servercore`

Más información

Para obtener más información sobre el uso de las AMI de Windows optimizadas para Amazon EKS, consulte las siguientes secciones:

- Para usar Windows con grupos de nodos administrados, consulte Grupos de nodos administrados.
- Para lanzar nodos de Windows autoadministrados, consulte Lanzamiento de nodos de Windows autoadministrados.
- Para obtener información sobre la versión, consulte Versiones de AMI optimizadas para Amazon EKS de Windows.
- Para recuperar los identificadores más recientes de las AMI de Windows optimizadas para Amazon EKS, consulte Recuperación de los ID de la AMI de Windows optimizada para Amazon EKS.

- Para utilizar Amazon EC2 Image Builder para crear las AMI de Windows personalizadas optimizadas para Amazon EKS, consulte Creación de AMI de Windows personalizadas optimizadas para Amazon EKS.
- Para conocer las prácticas recomendadas, consulte la Administración de AMI de Windows optimizadas para Amazon EKS en la Guía de prácticas recomendadas de EKS.

Versiones de AMI optimizadas para Amazon EKS de Windows

> **Important**
>
> El soporte extendido para las AMI de Windows optimizadas para Amazon EKS que sean publicadas por AWS no está disponible para la versión `1.23` de Kubernetes; sin embargo, no está disponible para la versión `1.24` o una posterior de Kubernetes.

En este tema, se enumeran las AMI de Windows optimizadas para Amazon EKS y sus versiones correspondientes de `kubelet`, `containerd` y `csi-proxy`.

Los metadatos de la AMI optimizada para Amazon EKS, incluido el ID de la AMI, de cada variante se pueden recuperar mediante programación. Para obtener más información, consulte Recuperación de los ID de la AMI de Windows optimizada para Amazon EKS.

Las AMI están versionadas por la versión de Kubernetes y la fecha de lanzamiento de la AMI en el siguiente formato:

```
k8s_major_version.k8s_minor_version-release_date
```

> **Note**
>
> Los grupos de nodos administrados de Amazon EKS con compatibles con las versiones de las AMI de Windows lanzadas en noviembre de 2022 y posteriores.

AMI de Windows Server 2022 Core optimizada para Amazon EKS

En las siguientes tablas, se enumeran las versiones actuales y anteriores de la AMI de Windows Server 2022 Core optimizada para Amazon EKS.

Kubernetes version 1.30

Versión Kubernetes**1.30**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1.29

Versión Kubernetes**1.29**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.29-2024.05.15	1.29.3	1.7.11	1.1.2	Se actualizó `containerd` a 1.7.11. Se actualizó `kubelet` a 1.29.3.
1.29-2024.04.09	1.29.0	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28. Se volvieron a crear CNI y `csi-proxy` mediante `golang` 1.22.1.
1.29-2024.03.12	1.29.0	1.6.25	1.1.2	
1.29-2024.02.13	1.29.0	1.6.25	1.1.2	
1.29-2024.02.06	1.29.0	1.6.25	1.1.2	Se corrigió el error de que la imagen de pausa se borraba incorrectamente durante el

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
				proceso de recogida de basura de `kubelet`.
`1.29-2024.01.11`	`1.29.0`	`1.6.18`	`1.1.2`	Se excluyó la actualización independiente de Windows KB5034439 en las AMI principales de Windows Server 2022. La base de conocimientos se aplica solo a las instalaciones de Windows con una partición de WinRE independiente, las cuales no están incluidas en ninguna de las AMI de Windows optimizadas para Amazon EKS.

Kubernetes version 1.28

Versión Kubernetes**1.28**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
`1.28-2024.05.14`	`1.28.8`	`1.6.28`	`1.1.2`	Se actualizó `containerd` a `1.6.28`. Se actualizó `kubelet` a `1.28.8`.
`1.28-2024.04.09`	`1.28.5`	`1.6.25`	`1.1.2`	Se actualizó `containerd` a `1.6.25`. Se volvieron a crear CNI y `csi-proxy` mediante `golang 1.22.1`.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2024.03.12	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	
1.28-2024.01.11	1.28.5	1.6.18	1.1.2	Se excluyó la actualización independiente de Windows KB5034439 en las AMI principales de Windows Server 2022. La base de conocimientos se aplica solo a las instalaciones de Windows con una partición de WinRE independiente, las cuales no están incluidas en ninguna de las AMI de Windows optimizadas para Amazon EKS.
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Versión Kubernetes**1.27**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28. Se actualizó `kubelet` a 1.27.12.
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	Se actualizó `containerd` a 1.6.25. Se volvieron a crear CNI y `csi-proxy` mediante golang 1.22.1.
1.27-2024.03.12	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.11	1.27.9	1.6.18	1.1.2	Se excluyó la actualización independiente de Windows KB5034439 en las AMI principales de Windows Server 2022. La base de conocimientos se aplica solo a las instalaciones de

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
				Windows con una partición de WinRE independiente, las cuales no están incluidas en ninguna de las AMI de Windows optimizadas para Amazon EKS.
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Se solucionó containers-roadmap el problema #2042, que provocaba que los nodos no pudieran extraer imágenes privadas de Amazon ECR.
1.27-2023.05.17	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Versión Kubernetes**1.26**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28. Se actualizó `kubelet` a 1.26.15.
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	Se actualizó `containerd` a 1.6.25. Se volvieron a crear CNI y `csi-proxy` mediante golang 1.22.1.
1.26-2024.03.12	1.26.12	1.6.18	1.1.2	
1.26-2024.02.13	1.26.12	1.6.18	1.1.2	
1.26-2024.01.11	1.26.12	1.6.18	1.1.2	Se excluyó la actualización independiente de Windows KB5034439 en las AMI principales de Windows Server 2022. La base de conocimientos se aplica solo a las instalaciones de Windows con una partición de WinRE independiente, las cuales no están incluidas en ninguna de las AMI de Windows optimizadas para Amazon EKS.
1.26-2023.12.12	1.26.10	1.6.18	1.1.2	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.11.14	1.26.10	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.26-2023.10.19	1.26.9	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.26.9. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.26-2023.09.12	1.26.7	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Se actualizó Kubernetes a 1.26.4. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Versión Kubernetes**1.25**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	Se actualizó `containerd` a 1.6.25. Se volvieron a crear CNI y `csi-proxy` mediante `golang` 1.22.1.
1.25-2024.03.12	1.25.16	1.6.18	1.1.2	
1.25-2024.02.13	1.25.16	1.6.18	1.1.2	
1.25-2024.01.11	1.25.16	1.6.18	1.1.2	Se excluyó la actualización independiente de Windows KB5034439 en las AMI principales de Windows Server 2022. La base de conocimientos se aplica solo a las instalaciones de Windows con una partición de WinRE independiente, las cuales no están incluidas en ninguna de las AMI de Windows optimizadas para Amazon EKS.
1.25-2023.12.12	1.25.15	1.6.18	1.1.2	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.11.14	1.25.15	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.25-2023.10.19	1.25.14	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.25.14. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Se actualizó Kubernetes a 1.25.9. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Versión Kubernetes**1.24**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	Se actualizó `containerd` a 1.6.25. Se volvieron a crear CNI y `csi-proxy` mediante `golang` 1.22.1.
1.24-2024.03.12	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.11	1.24.17	1.6.18	1.1.2	Se excluyó la actualización independiente de Windows KB5034439 en las AMI principales de Windows Server 2022. La base de conocimientos se aplica solo a las instalaciones de Windows con una partición de

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
				WinRE independiente, las cuales no están incluidas en ninguna de las AMI de Windows optimizadas para Amazon EKS.
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.24.17. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.20	1.24.13	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Se actualizó Kubernetes a 1.24.13. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación gMSA de los contenedores de Windows en Amazon EKS.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	La versión de Kubernetes se disminuyó a 1.24.7 porque 1.24.10 tiene un problema notificado en kube-proxy .
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	
1.24-2022.12.13	1.24.7	1.6.6	1.1.1	
1.24-2022.10.11	1.24.7	1.6.6	1.1.1	

AMI de Windows Server 2022 Full optimizada para Amazon EKS

En las siguientes tablas, se enumeran las versiones actuales y anteriores de la AMI de Windows Server 2022 Full optimizada para Amazon EKS.

Kubernetes version 1.30

Versión Kubernetes**1.30**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1.29

Versión Kubernetes**1.29**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.29-2024.05.15	1.29.3	1.7.11	1.1.2	Se actualizó `containerd` a 1.7.11. Se actualizó `kubelet` a 1.29.3.
1.29-2024.04.09	1.29.0	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28. Se volvieron a crear CNI y `csi-proxy` mediante `golang` 1.22.1.
1.29-2024.03.12	1.29.0	1.6.25	1.1.2	
1.29-2024.02.13	1.29.0	1.6.25	1.1.2	
1.29-2024.02.06	1.29.0	1.6.25	1.1.2	Se corrigió el error de que la imagen de pausa se borraba incorrectamente durante el

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
				proceso de recogida de basura de `kubelet`.
`1.29-2024.01.09`	`1.29.0`	`1.6.18`	`1.1.2`	

Kubernetes version 1.28

Versión Kubernetes**1.28**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
`1.28-2024.05.14`	`1.28.8`	`1.6.28`	`1.1.2`	Se actualizó `containerd` a `1.6.28`. Se actualizó `kubelet` a `1.28.8`.
`1.28-2024.04.09`	`1.28.5`	`1.6.25`	`1.1.2`	Se actualizó `containerd` a `1.6.25`. Se volvieron a crear CNI y `csi-proxy` mediante `golang 1.22.1`.
`1.28-2024.03.12`	`1.28.5`	`1.6.18`	`1.1.2`	
`1.28-2024.02.13`	`1.28.5`	`1.6.18`	`1.1.2`	
`1.28-2024.01.09`	`1.28.5`	`1.6.18`	`1.1.2`	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Versión Kubernetes**1.27**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.27.12.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.27-2024.03.12	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.09	1.27.9	1.6.18	1.1.2	
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Se solucionó containers-roadmap el problema #2042, que provocaba que los nodos no pudieran extraer imágenes privadas de Amazon ECR.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.05.18	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Versión Kubernetes**1.26**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	Se actualizó `containerd` a 1.6.28. Se actualizó `kubelet` a 1.26.15.
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	Se actualizó `containerd` a 1.6.25. Se volvieron a crear CNI y `csi-proxy` mediante golang 1.22.1.
1.26-2024.03.12	1.26.12	1.6.18	1.1.2	
1.26-2024.02.13	1.26.12	1.6.18	1.1.2	
1.26-2024.01.09	1.26.12	1.6.18	1.1.2	
1.26-2023.12.12	1.26.10	1.6.18	1.1.2	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.11.14	1.26.10	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.26-2023.10.19	1.26.9	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.26.9. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.26-2023.09.12	1.26.7	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Se actualizó Kubernetes a 1.26.4. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Versión Kubernetes**1.25**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	Se actualizó containerd a 1.6.28.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.25-2024.03.12	1.25.16	1.6.18	1.1.2	
1.25-2024.02.13	1.25.16	1.6.18	1.1.2	
1.25-2024.01.09	1.25.16	1.6.18	1.1.2	
1.25-2023.12.12	1.25.15	1.6.18	1.1.2	
1.25-2023.11.14	1.25.15	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.25-2023.10.19	1.25.14	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.25.14. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Se actualizó Kubernetes a 1.25.9. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Versión Kubernetes**1.24**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	Se actualizó containerd a 1.6.28.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.24-2024.03.12	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.09	1.24.17	1.6.18	1.1.2	
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.24.17. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.20	1.24.13	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Se actualizó Kubernetes a 1.24.13. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	La versión de Kubernetes se disminuyó a 1.24.7 porque 1.24.10 tiene un problema notificado en kube-proxy .
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2022.12.14	1.24.7	1.6.6	1.1.1	
1.24-2022.10.11	1.24.7	1.6.6	1.1.1	

La AMI de Windows Server 2019 Core optimizada para Amazon EKS

En las siguientes tablas, se enumeran las versiones actuales y anteriores de la AMI de Windows Server 2019 Core optimizada para Amazon EKS.

Kubernetes version 1.30

Versión Kubernetes**1.30**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1.29

Versión Kubernetes**1.29**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.29-2024.05.15	1.29.3	1.7.11	1.1.2	Se actualizó containerd a 1.7.11. Se actualizó kubelet a 1.29.3.
1.29-2024.04.09	1.29.0	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.29-2024.03.13	1.29.0	1.6.25	1.1.2	
1.29-2024.02.13	1.29.0	1.6.25	1.1.2	
1.29-2024.02.06	1.29.0	1.6.25	1.1.2	Se corrigió el error de que la imagen de pausa se borraba incorrectamente durante el proceso de recogida de basura de kubelet.
1.29-2024.01.09	1.29.0	1.6.18	1.1.2	

Kubernetes version 1.28

Versión Kubernetes**1.28**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2024.05.14	1.28.8	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.28.8.
1.28-2024.04.09	1.28.5	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.28-2024.03.13	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	
1.28-2024.01.09	1.28.5	1.6.18	1.1.2	
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Versión Kubernetes**1.27**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.27.12.
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.27-2024.03.13	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.09	1.27.9	1.6.18	1.1.2	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Se solucionó containers-roadmap el problema #2042, que provocaba que los nodos no pudieran extraer imágenes privadas de Amazon ECR.
11.27-2023.05.18	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Versión Kubernetes**1.26**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.26.15.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.26-2024.03.13	1.26.12	1.6.18	1.1.2	
1.26-2024.02.13	1.26.12	1.6.18	1.1.2	
1.26-2024.01.09	1.26.12	1.6.18	1.1.2	
1.26-2023.12.12	1.26.10	1.6.18	1.1.2	
1.26-2023.11.14	1.26.10	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.26-2023.10.19	1.26.9	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.26.9. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.09.12	1.26.7	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Se actualizó Kubernetes a 1.26.4. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Versión Kubernetes**1.25**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	Se actualizó containerd a 1.6.28.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
				CNI y `csi-proxy` mediante `golang 1.22.1`.
`1.25-2024.03.13`	`1.25.16`	`1.6.18`	`1.1.2`	
`1.25-2024.02.13`	`1.25.16`	`1.6.18`	`1.1.2`	
`1.25-2024.01.09`	`1.25.16`	`1.6.18`	`1.1.2`	
`1.25-2023.12.12`	`1.25.15`	`1.6.18`	`1.1.2`	
`1.25-2023.11.14`	`1.25.15`	`1.6.18`	`1.1.2`	Incluye parches para `CVE-2023-5528` .
`1.25-2023.10.19`	`1.25.14`	`1.6.18`	`1.1.2`	Se actualizó `containerd` a `1.6.18`. Se actualizó `kubelet` a `1.25.14`. Se agregaron nuevas variables de entorno de script de arranque (`SERVICE_IPV4_CIDR` y `EXCLUDED_SNAT_CIDRS`).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Se actualizó Kubernetes a 1.25.9. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Versión Kubernetes**1.24**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	Se actualizó containerd a 1.6.28.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.24-2024.03.13	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.09	1.24.17	1.6.18	1.1.2	
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.24.17. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.20	1.24.13	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Se actualizó Kubernetes a 1.24.13. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	La versión de Kubernetes se disminuyó a 1.24.7 porque 1.24.10 tiene un problema notificado en kube-proxy .
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2022.12.13	1.24.7	1.6.6	1.1.1	
1.24-2022.11.08	1.24.7	1.6.6	1.1.1	

AMI de Windows Server 2019 Full optimizada para Amazon EKS

En las siguientes tablas, se enumeran las versiones actuales y anteriores de la AMI de Windows Server 2019 Full optimizada para Amazon EKS.

Kubernetes version 1.30

Versión Kubernetes**1.30**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.30-2024.05.15	1.30.0	1.6.28	1.1.2	

Kubernetes version 1.29

Versión Kubernetes**1.29**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.29-2024.05.15	1.29.3	1.7.11	1.1.2	Se actualizó containerd a 1.7.11. Se actualizó kubelet a 1.29.3.
1.29-2024.04.09	1.29.0	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.29-2024.03.13	1.29.0	1.6.25	1.1.2	
1.29-2024.02.13	1.29.0	1.6.25	1.1.2	
1.29-2024.02.06	1.29.0	1.6.25	1.1.2	Se corrigió el error de que la imagen de pausa se borraba incorrectamente durante el proceso de recogida de basura de kubelet.
1.29-2024.01.09	1.29.0	1.6.18	1.1.2	

Kubernetes version 1.28

Versión Kubernetes**1.28**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2024.05.14	1.28.8	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.28.8.
1.28-2024.04.09	1.28.5	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.28-2024.03.13	1.28.5	1.6.18	1.1.2	
1.28-2024.02.13	1.28.5	1.6.18	1.1.2	
1.28-2024.01.09	1.28.5	1.6.18	1.1.2	
1.28-2023.12.12	1.28.3	1.6.18	1.1.2	
1.28-2023.11.14	1.28.3	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.28-2023.10.19	1.28.2	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.28-2023-09.27	1.28.2	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.
1.28-2023.09.12	1.28.1	1.6.6	1.1.2	

Kubernetes version 1.27

Versión Kubernetes**1.27**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2024.05.14	1.27.12	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.27.12.
1.27-2024.04.09	1.27.9	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.27-2024.03.13	1.27.9	1.6.18	1.1.2	
1.27-2024.02.13	1.27.9	1.6.18	1.1.2	
1.27-2024.01.09	1.27.9	1.6.18	1.1.2	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.12.12	1.27.7	1.6.18	1.1.2	
1.27-2023.11.14	1.27.7	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.27-2023.10.19	1.27.6	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).
1.27-2023-09.27	1.27.6	1.6.6	1.1.2	Se ha corregido un aviso de seguridad en kubelet.
1.27-2023.09.12	1.27.4	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.27-2023.08.17	1.27.4	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.27-2023.08.08	1.27.3	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.27-2023.07.11	1.27.3	1.6.6	1.1.1	
1.27-2023.06.20	1.27.1	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.27-2023.06.14	1.27.1	1.6.6	1.1.1	Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.
1.27-2023.06.06	1.27.1	1.6.6	1.1.1	Se solucionó containers-roadmap el problema #2042, que provocaba que los nodos no pudieran extraer imágenes privadas de Amazon ECR.
1.27-2023.05.17	1.27.1	1.6.6	1.1.1	

Kubernetes version 1.26

Versión Kubernetes**1.26**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2024.05.14	1.26.15	1.6.28	1.1.2	Se actualizó containerd a 1.6.28. Se actualizó kubelet a 1.26.15.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2024.04.09	1.26.12	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.26-2024.03.13	1.26.12	1.6.18	1.1.2	
1.26-2024.02.13	1.26.12	1.6.18	1.1.2	
1.26-2024.01.09	1.26.12	1.6.18	1.1.2	
1.26-2023.12.12	1.26.10	1.6.18	1.1.2	
1.26-2023.11.14	1.26.10	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.26-2023.10.19	1.26.9	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.26.9. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.09.12	1.26.7	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.26-2023.08.17	1.26.7	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.26-2023.08.08	1.26.6	1.6.6	1.1.1	
1.26-2023.07.11	1.26.6	1.6.6	1.1.1	
1.26-2023.06.20	1.26.4	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.26-2023.06.14	1.26.4	1.6.6	1.1.1	Se actualizó Kubernetes a 1.26.4. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.26-2023.05.09	1.26.2	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.26-2023.04.26	1.26.2	1.6.6	1.1.1	
1.26-2023.04.11	1.26.2	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.26-2023.03.24	1.26.2	1.6.6	1.1.1	

Kubernetes version 1.25

Versión Kubernetes**1.25**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2024.05.14	1.25.16	1.6.28	1.1.2	Se actualizó containerd a 1.6.28.
1.25-2024.04.09	1.25.16	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
				CNI y `csi-proxy` mediante `golang 1.22.1`.
`1.25-2024.03.13`	`1.25.16`	`1.6.18`	`1.1.2`	
`1.25-2024.02.13`	`1.25.16`	`1.6.18`	`1.1.2`	
`1.25-2024.01.09`	`1.25.16`	`1.6.18`	`1.1.2`	
`1.25-2023.12.12`	`1.25.15`	`1.6.18`	`1.1.2`	
`1.25-2023.11.14`	`1.25.15`	`1.6.18`	`1.1.2`	Incluye parches para `CVE-2023-5528` .
`1.25-2023.10.19`	`1.25.14`	`1.6.18`	`1.1.2`	Se actualizó `containerd` a `1.6.18`. Se actualizó `kubelet` a `1.25.14`. Se agregaron nuevas variables de entorno de script de arranque (`SERVICE_IPV4_CIDR` y `EXCLUDED_SNAT_CIDRS`).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.09.12	1.25.12	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.25-2023.08.17	1.25.12	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.25-2023.08.08	1.25.9	1.6.6	1.1.1	
1.25-2023.07.11	1.25.9	1.6.6	1.1.1	
1.25-2023.06.20	1.25.9	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.25-2023.06.14	1.25.9	1.6.6	1.1.1	Se actualizó Kubernetes a 1.25.9. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.25-2023.05.09	1.25.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.25-2023.04.11	1.25.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.25-2023.03.27	1.25.6	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.25-2023.03.20	1.25.6	1.6.6	1.1.1	
1.25-2023.02.14	1.25.6	1.6.6	1.1.1	

Kubernetes version 1.24

Versión Kubernetes**1.24**

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2024.05.14	1.24.17	1.6.28	1.1.2	Se actualizó containerd a 1.6.28.
1.24-2024.04.09	1.24.17	1.6.25	1.1.2	Se actualizó containerd a 1.6.25. Se volvieron a crear CNI y csi-proxy mediante golang 1.22.1.
1.24-2024.03.13	1.24.17	1.6.18	1.1.2	
1.24-2024.02.13	1.24.17	1.6.18	1.1.2	
1.24-2024.01.09	1.24.17	1.6.18	1.1.2	
1.24-2023.12.12	1.24.17	1.6.18	1.1.2	
1.24-2023.11.14	1.24.17	1.6.18	1.1.2	Incluye parches para CVE-2023-5528 .
1.24-2023.10.19	1.24.17	1.6.18	1.1.2	Se actualizó containerd a 1.6.18. Se actualizó kubelet a 1.24.17. Se agregaron nuevas variables de entorno de script de arranque (SERVICE_IPV4_CIDR y EXCLUDED_SNAT_CIDRS).

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.09.12	1.24.16	1.6.6	1.1.2	Se actualizó el complemento CNI de Amazon VPC para que utilice el binario del conector de Kubernetes, que obtiene la dirección IP del Pod del servidor de API de Kubernetes. Se combinó la solicitud de extracción n.º 100.
1.24-2023.08.17	1.24.16	1.6.6	1.1.2	Incluye parches para CVE-2023-3676 , CVE-2023-3893 y CVE-2023-3955 .
1.24-2023.08.08	1.24.13	1.6.6	1.1.1	
1.24-2023.07.11	1.24.13	1.6.6	1.1.1	
1.24-2023.06.21	1.24.13	1.6.6	1.1.1	Se ha resuelto el problema que provocaba que la lista de búsqueda de sufijos DNS se rellenara incorrectamente.
1.24-2023.06.14	1.24.13	1.6.6	1.1.1	Se actualizó Kubernetes a 1.24.13. Se agregó soporte para el mapeo de host en CNI. Solicitud de extracción combinada #93.

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2023.05.09	1.24.7	1.6.6	1.1.1	Se corrigió un error que causaba el problema de conectividad de red #1126 en los pods tras el reinicio del nodo. Se introdujo un nuevo parámetro de configuración del script de arranque (ExcludedSnatCIDRs).
1.24-2023.04.11	1.24.7	1.6.6	1.1.1	Se agregó un mecanismo de recuperación para kubelet y kube-proxy durante la caída del servicio.
1.24-2023.03.27	1.24.7	1.6.6	1.1.1	Se instaló un complemento de gMSA sin dominio para facilitar la autenticación de gMSA de los contenedores de Windows en Amazon EKS.
1.24-2023.03.20	1.24.7	1.6.6	1.1.1	La versión de Kubernetes se disminuyó a 1.24.7 porque 1.24.10 tiene un problema notificado en kube-proxy .
1.24-2023.02.14	1.24.10	1.6.6	1.1.1	
1.24-2023.01.23	1.24.7	1.6.6	1.1.1	
1.24-2023.01.11	1.24.7	1.6.6	1.1.1	

Versión de AMI	Versión de **kubelet**	Versión de **containerd**	Versión de **csi-proxy**	Notas de la versión
1.24-2022.12.14	1.24.7	1.6.6	1.1.1	
1.24-2022.10.12	1.24.7	1.6.6	1.1.1	

Recuperación de los ID de la AMI de Windows optimizada para Amazon EKS

Puede recuperar el ID de Amazon Machine Image (AMI) de las AMI optimizadas para Amazon EKS mediante programación al consultar la API de Parameter Store de AWS Systems Manager. Este parámetro elimina la necesidad de buscar de manera manual los ID de la AMI optimizada para Amazon EKS. Para obtener más información acerca de la API de Systems Manager Parameter Store, consulte GetParameter. La entidad principal de IAM que utiliza debe tener el permiso `ssm:GetParameter` de IAM para recuperar los metadatos de la AMI optimizada para Amazon EKS.

Puede recuperar el ID de imagen de la última AMI de Windows optimizada para Amazon EKS recomendada con el siguiente comando mediante el parámetro secundario `image_id`. Puede reemplazar *1.30* por cualquier versión compatible de Amazon EKS y puede reemplazar *region-code* por una región compatible con Amazon EKS para la que desea el ID de la AMI. Sustituya *Core* con `Full` para ver el ID de la AMI completo de Windows Server. Para Kubernetes versión `1.24` o posterior, puede reemplazar *2019* por `2022` para ver el ID de la AMI de Windows Server 2022.

```
aws ssm get-parameter --name /aws/service/ami-windows-latest/Windows_Server-2019-English-Core-EKS_Optimized-1.30/image_id --region region-code --query "Parameter.Value" --output text
```

Un ejemplo de salida sería el siguiente.

```
ami-1234567890abcdef0
```

Creación de AMI de Windows personalizadas optimizadas para Amazon EKS

Puede utilizar Image Builder de EC2 para crear las AMI de Windows personalizadas optimizadas para Amazon EKS con una de las siguientes opciones:

- Uso de una AMI de Windows optimizada para Amazon EKS como base
- Uso del componente de compilación administrado por Amazon

Con ambos métodos, debe crear su propia receta de Image Builder. Para obtener más información, consulte Crear una nueva versión de una receta de imagen en la Guía del usuario de Image Builder.

> **Important**
>
> Los siguientes componentes administrados por Amazon para `eks` incluyen parches para `CVE-2023-5528`.
>
> - `1.24.3` y posteriores
> - `1.25.2` y posteriores
> - `1.26.2` y posteriores
> - `1.27.0` y posteriores
> - `1.28.0` y posteriores

Uso de una AMI de Windows optimizada para Amazon EKS como base

Esta opción es la forma recomendada de crear las AMI de Windows personalizadas. Las AMI de Windows optimizadas para Amazon EKS que ofrecemos se actualizan con más frecuencia que el componente de compilación administrado por Amazon.

1. Inicie una receta nueva de Image Builder.
   a. Abra la consola de EC2 Image Builder enhttps://console.aws.amazon.com/imagebuilder.
   b. En el panel de navegación izquierdo, elija Recetas de imágenes.
   c. Seleccione Crear receta de imagen.
2. En la sección Detalles de la receta, ingrese un nombre y una versión.
3. Especifique el ID de la AMI de Windows optimizada para Amazon EKS en la sección Imagen base.

a. Elija Escribir el ID personalizado de la AMI.

b. Recupere el ID de la AMI de la versión del sistema operativo de Windows que necesita. Para obtener más información, consulte Recuperación de los ID de la AMI de Windows optimizada para Amazon EKS.

c. Ingrese el ID de la AMI personalizada. Si no encuentra el ID de la AMI, asegúrese de que la Región de AWS del ID de la AMI coincida con la Región de AWS que se muestra en la esquina superior derecha de la consola.

4. (Opcional) Para obtener las actualizaciones de seguridad más recientes, agrega el componente `update-windows` en la sección Componentes de compilación: .

a. En la lista desplegable situada a la derecha del cuadro de búsqueda Buscar componentes por nombre, seleccione Administrado por Amazon.

b. En el cuadro de búsqueda Buscar componentes por nombre, ingrese **`update-windows`**.

c. Seleccione la casilla de verificación del resultado de búsqueda **`update-windows`**. Este componente incluye las revisiones de Windows más recientes para el sistema operativo.

5. Complete las entradas restantes de la receta de imágenes con las configuraciones necesarias. Para obtener más información, consulte Crear una nueva versión de una receta de imagen (consola) en la Guía del usuario de Image Builder.

6. Elija Crear receta.

7. Utilice la nueva receta de imagen en una canalización de imágenes nueva o existente. Una vez que la canalización de imágenes se ejecute correctamente, la AMI personalizada aparecerá como imagen de salida y estará lista para su uso. Para obtener más información, consulte Crear una canalización de imágenes mediante el asistente de la consola de Image Builder de EC2.

Uso del componente de compilación administrado por Amazon

Si no es viable utilizar una AMI de Windows optimizada para Amazon EKS como base, puede utilizar en su lugar el componente de compilación administrado por Amazon. Esta opción puede estar retrasada con respecto a las versiones de Kubernetes compatibles más recientes.

1. Inicie una receta nueva de Image Builder.

a. Abra la consola de EC2 Image Builder enhttps://console.aws.amazon.com/imagebuilder.

b. En el panel de navegación izquierdo, elija Recetas de imágenes.

c. Seleccione Crear receta de imagen.

2. En la sección Detalles de la receta, ingrese un nombre y una versión.
3. Determine qué opción utilizará para crear su AMI personalizada en la sección Imagen base:
   - Seleccione imágenes administradas: elija Windows para su sistema operativo (SO) de imágenes. A continuación, elija una de las siguientes opciones para Origen de la imagen.
     - Inicio rápido (administrado por Amazon): en el menú desplegable Nombre de la imagen, seleccione una versión de Windows Server compatible con Amazon EKS. Para obtener más información, consulte AMI de Windows optimizadas para Amazon EKS.
     - Imágenes de mi propiedad: en Nombre de la imagen, seleccione el ARN de su propia imagen con su propia licencia. La imagen que proporcione no puede tener ya instalados los componentes de Amazon EKS.
   - Escribir el ID personalizado de la AMI: en el ID de la AMI, ingrese el ID de su AMI con su propia licencia. La imagen que proporcione no puede tener ya instalados los componentes de Amazon EKS.
4. En la sección Componentes de compilación: Windows, haga lo siguiente:
   a. En la lista desplegable situada a la derecha del cuadro de búsqueda Buscar componentes por nombre, seleccione Administrado por Amazon.
   b. En el cuadro de búsqueda Buscar componentes por nombre, ingrese **`eks`**.
   c. Seleccione el resultado de búsqueda **`eks-optimized-ami-windows`**, aunque el resultado devuelto puede no ser la versión que desea.
   d. En el cuadro de búsqueda Buscar componentes por nombre, ingrese **`update-windows`**.
   e. Seleccione la casilla de verificación del resultado de búsqueda update-windows. Este componente incluye las revisiones de Windows más recientes para el sistema operativo.
5. En la sección Componentes seleccionados, haga lo siguiente:
   a. Elija Opciones de control de versiones para **`eks-optimized-ami-windows`**.
   b. Elija Especificar la versión del componente.
   c. En el campo Versión del componente, ingrese ***`version.x`*** y reemplace *`version`* por una versión de Kubernetes compatible. Al ingresar una *`x`* para parte del número de versión, se indica que se debe utilizar la versión más reciente del componente, que también se alinea con la parte de la versión que defina explícitamente. Preste atención a la salida de la consola, ya que le indicará si la versión que desea está disponible como componente administrado. Tenga en cuenta que es posible que las versiones de Kubernetes más recientes no estén disponibles para el componente de compilación. Para obtener más

información sobre versiones disponibles, consulte Recuperación de información sobre las versiones de los componentes de `eks-optimized-ami-windows`.

> **Note**
>
> Las siguientes versiones de componentes de compilación de `eks-optimized-ami-windows` requieren la versión de `eksctl` `0.129` o una inferior:
>
> - `1.24.0`

6. Complete las entradas restantes de la receta de imágenes con las configuraciones necesarias. Para obtener más información, consulte Crear una nueva versión de una receta de imagen (consola) en la Guía del usuario de Image Builder.
7. Elija Crear receta.
8. Utilice la nueva receta de imagen en una canalización de imágenes nueva o existente. Una vez que la canalización de imágenes se ejecute correctamente, la AMI personalizada aparecerá como imagen de salida y estará lista para su uso. Para obtener más información, consulte Crear una canalización de imágenes mediante el asistente de la consola de Image Builder de EC2.

Recuperación de información sobre las versiones de los componentes de **`eks-optimized-ami-windows`**

Puede recuperar información específica sobre lo que se instala con cada componente. Por ejemplo, puede comprobar qué versión de `kubelet` está instalada. Los componentes pasan por pruebas funcionales en las versiones de sistemas operativos Windows compatibles con Amazon EKS. Para obtener más información, consulte Calendario de versiones. Es posible que otras versiones de sistemas operativos Windows que se indican como compatibles o que han llegado al fin del soporte técnico no sean compatibles con el componente.

1. Abra la consola de EC2 Image Builder enhttps://console.aws.amazon.com/imagebuilder.
2. En el panel de navegación izquierdo, elija Componentes.
3. En la lista desplegable situada a la derecha del cuadro de búsqueda Buscar componentes por nombre, cambie De mi propiedad a Inicio rápido (administrado por Amazon).
4. En el recuadro Find components by name (Buscar componentes por nombre), ingrese **`eks`**.
5. (Opcional) Si utiliza una versión reciente, seleccione dos veces la columna Versión para ordenarla en orden descendente.

6. Elija el enlace **`eks-optimized-ami-windows`** con la versión que desee.

La descripción de la página resultante muestra la información específica.

Almacenamiento

En este capítulo se tratan las opciones de almacenamiento para clústeres de Amazon EKS.

Temas

Controlador CSI de Amazon EBS

El controlador de la interfaz de almacenamiento de contenedores (CSI) de Amazon Elastic Block Store (Amazon EBS) administra el ciclo de vida de los volúmenes de Amazon EBS como almacenamiento para los volúmenes de Kubernetes que se creen. El controlador de CSI de Amazon EBS crea volúmenes de Amazon EBS para estos tipos de volúmenes de Kubernetes: volúmenes efímeros genéricos y volúmenes persistentes.

Aquí se incluyen algunos aspectos que debe tener en cuenta sobre la utilización del controlador de CSI de Amazon EBS.

- El complemento CSI de Amazon EBS requiere permisos de IAM para realizar llamadas a las API de AWS en su nombre. Para obtener más información, consulte Creación del rol de IAM del controlador de CSI de Amazon EBS.
- No puede montar volúmenes de Amazon EBS en Fargate Pods.
- Puede ejecutar el controlador CSI de Amazon EBS en nodos de Fargate, pero el nodo CSI de Amazon EBS DaemonSet solo se puede ejecutar en instancias de Amazon EC2.

El controlador de CSI de Amazon EBS no se instala la primera vez que crea un clúster. Para utilizar el controlador, debe agregarlo como complemento de Amazon EKS o como complemento autoadministrado.

- Para obtener instrucciones sobre cómo agregarlo como complemento de Amazon EKS, consulteAdministración del controlador de CSI de Amazon EBS como complemento de Amazon EKS.
- Para obtener instrucciones sobre cómo agregarlo como instalación autoadministrada, consulte el proyecto Controlador de Amazon EBS Container Storage Interface (CSI) en GitHub.

Después de instalar el controlador CSI con cualquiera de los dos métodos, puede probar la funcionalidad con una aplicación de ejemplo. Para obtener más información, consulte Implemente una aplicación de muestra y verifique que el controlador CSI funciona.

Creación del rol de IAM del controlador de CSI de Amazon EBS

El complemento CSI de Amazon EBS requiere permisos de IAM para realizar llamadas a las API de AWS en su nombre. Para obtener más información, consulte Configurar el permiso del controlador en GitHub.

> **Note**
>
> Los Pods tendrán acceso a los permisos asignados al rol de IAM, a menos que bloquee el acceso al IMDS. Para obtener más información, consulte Prácticas recomendadas de seguridad para Amazon EKS.

Requisitos previos

- Un clúster existente.
- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.

El siguiente procedimiento muestra cómo crear un rol de IAM y asociarle la política administrada de AWS. Puede utilizar la `eksctl`, la AWS Management Console o la AWS CLI.

> **Note**
>
> Los pasos específicos de este procedimiento están diseñados para usar el controlador como complemento de Amazon EKS. Se necesitan diferentes pasos para usar el controlador como complemento autoadministrado. Para obtener más información, consulte Configurar los permisos de los controladores en GitHub.

eksctl

Para crear el rol de IAM del complemento CSI de Amazon EBS con la **`eksctl`**

1. Cree un rol de IAM y asóciele una política. AWS mantiene una política administrada de AWS, pero también puede crear su propia política personalizada. Puede crear un rol de IAM y asociar la política administrada de AWS con el siguiente comando. Reemplace *`my-cluster`* por el nombre del clúster. El comando implementa una pila de AWS CloudFormation que crea un rol de IAM y le adjunta la política de IAM. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

   ```
   eksctl create iamserviceaccount \
       --name ebs-csi-controller-sa \
       --namespace kube-system \
       --cluster my-cluster \
       --role-name AmazonEKS_EBS_CSI_DriverRole \
       --role-only \
       --attach-policy-arn arn:aws:iam::aws:policy/service-role/
   AmazonEBSCSIDriverPolicy \
       --approve
   ```

2. Si utiliza una clave de KMS personalizada para el cifrado en los volúmenes de Amazon EBS, personalice el rol de IAM según sea necesario. Por ejemplo, haga lo siguiente:

   a. Copie y pegue el siguiente código en un nuevo archivo *`kms-key-for-encryption-on-ebs`*`.json`. Reemplace *`custom-key-arn`* con el ARN de la clave de KMS personalizado.

      ```
      {
        "Version": "2012-10-17",
        "Statement": [
      ```

```
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["custom-key-arn"],
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": ["custom-key-arn"]
    }
  ]
}
```

b. Cree la política. Puede cambiar *KMS_Key_For_Encryption_On_EBS_Policy* a un nombre diferente. Sin embargo, si lo hace, asegúrese de hacerlo también en pasos posteriores.

```
aws iam create-policy \
  --policy-name KMS_Key_For_Encryption_On_EBS_Policy \
  --policy-document file://kms-key-for-encryption-on-ebs.json
```

c. Adjunte una política de IAM y adjunte la política de IAM al rol con el siguiente comando. Reemplace *111122223333* por su ID de cuenta. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
```

```
  --policy-arn
arn:aws:iam::111122223333:policy/KMS_Key_For_Encryption_On_EBS_Policy \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

AWS Management Console

Para crear el rol de IAM del complemento CSI de Amazon EBS con la AWS Management Console

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la página Roles, elija Crear rol.
4. En la página Seleccionar entidad de confianza, haga lo siguiente:
   a. En la sección Tipo de entidad de confianza, elija Identidad web.
   b. En Identity provider (Proveedor de identidades), elija la URL del proveedor de OpenID Connect para su clúster (como se muestra en Overview [Información general] en Amazon EKS).
   c. En Audiencia, elija `sts.amazonaws.com`.
   d. Elija Siguiente.
5. En la página Agregar permisos, haga lo siguiente:
   a. En el cuadro Filtrar políticas, escriba `AmazonEBSCSIDriverPolicy`.
   b. Marque la casilla situada a la izquierda del nombre de la `AmazonEBSCSIDriverPolicy` que obtuvo en la búsqueda.
   c. Elija Siguiente.
6. En la página Nombrar, revisar y crear, haga lo siguiente:
   a. En Nombre del rol, ingrese un nombre único para su rol, por ejemplo, ***AmazonEKS_EBS_CSI_DriverRole***.
   b. En Agregar etiquetas (Opcional), de manera opcional, agregue metadatos al rol asociando etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetado de recursos de IAM en la Guía de usuario de IAM.
   c. Seleccione Crear rol.

7. Una vez creado el rol, seleccione el rol en la consola para abrirlo y editarlo.
8. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.
9. Busque la línea que se parezca a la siguiente:

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud":
 "sts.amazonaws.com"
```

Agregue una coma al final de la línea anterior y, luego, agregue la siguiente línea después de la anterior. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Reemplace *EXAMPLED539D4633E53DE1B71EXAMPLE* con el ID del proveedor OIDC del clúster.

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub":
 "system:serviceaccount:kube-system:ebs-csi-controller-sa"
```

10. Elija Actualizar política para terminar.
11. Si utiliza una clave de KMS personalizada para el cifrado en los volúmenes de Amazon EBS, personalice el rol de IAM según sea necesario. Por ejemplo, haga lo siguiente:
    a. En el panel de navegación izquierdo, elija Policies (Políticas).
    b. En la página Políticas, seleccione Crear una política.
    c. En la página Crear política, elija la pestaña JSON.
    d. Copie y pegue el siguiente código en el editor y reemplace *custom-key-arn* con el ARN de la clave KMS personalizado.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["custom-key-arn"],
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
```

```
          }
        }
      },
      {
        "Effect": "Allow",
        "Action": [
          "kms:Encrypt",
          "kms:Decrypt",
          "kms:ReEncrypt*",
          "kms:GenerateDataKey*",
          "kms:DescribeKey"
        ],
        "Resource": ["custom-key-arn"]
      }
    ]
}
```

e. Elija Siguiente: etiquetas.

f. En la página Agregar etiquetas (opcional), elija Siguiente: revisar.

g. Para Nombre, escriba un nombre único para la política (por ejemplo, ***KMS_Key_For_Encryption_On_EBS_Policy***).

h. Elija Crear política.

i. En el panel de navegación izquierdo, seleccione Roles.

j. Elija ***AmazonEKS_EBS_CSI_DriverRole*** en la consola para abrirlo y editarlo.

k. En la lista desplegable Agregar permisos, seleccione Asociar políticas.

l. En el cuadro Filtrar políticas, escriba *KMS_Key_For_Encryption_On_EBS_Policy*.

m. Marque la casilla situada a la izquierda del nombre de la *KMS_Key_For_Encryption_On_EBS_Policy* que obtuvo en la búsqueda.

n. Seleccione Asociar políticas.

AWS CLI

Para crear el rol de IAM del complemento CSI de Amazon EBS con la AWS CLI

1. Vea la URL del proveedor de OIDC de su clúster. Reemplace *my-cluster* por el nombre de su clúster. Si la salida del comando es `None`, revise los Requisitos previos.

```
aws eks describe-cluster --name my-cluster --query
 "cluster.identity.oidc.issuer" --output text
```

Un ejemplo de salida sería el siguiente.

```
https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
```

2. Cree el rol de IAM y otórguele la acción `AssumeRoleWithWebIdentity`.

   a. Copie el siguiente contenido en un archivo denominado *aws-ebs-csi-driver-trust-policy*`.json`. Reemplace *111122223333* por su ID de cuenta. Reemplace *EXAMPLED539D4633E53DE1B71EXAMPLE* y *region-code* por los valores que se devolvieron en el paso anterior. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:ebs-csi-controller-sa"
        }
      }
    }
  ]
}
```

b. Cree el rol. Puede cambiar *AmazonEKS_EBS_CSI_DriverRole* a un nombre diferente. Si lo cambia, asegúrese de hacerlo también en pasos posteriores.

```
aws iam create-role \
  --role-name AmazonEKS_EBS_CSI_DriverRole \
  --assume-role-policy-document file://"aws-ebs-csi-driver-trust-policy.json"
```

3. Asocie una política. AWS mantiene una política administrada de AWS, pero también puede crear su propia política personalizada. Asocie la política administrada de AWS al rol con el siguiente comando. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

4. Si utiliza una clave de KMS personalizada para el cifrado en los volúmenes de Amazon EBS, personalice el rol de IAM según sea necesario. Por ejemplo, haga lo siguiente:

a. Copie y pegue el siguiente código en un nuevo archivo *kms-key-for-encryption-on-ebs*`.json`. Reemplace *custom-key-arn* con el ARN de la clave de KMS personalizado.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": ["custom-key-arn"],
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": ["custom-key-arn"]
    }
  ]
}
```

b. Cree la política. Puede cambiar *KMS_Key_For_Encryption_On_EBS_Policy* a un nombre diferente. Sin embargo, si lo hace, asegúrese de hacerlo también en pasos posteriores.

```
aws iam create-policy \
  --policy-name KMS_Key_For_Encryption_On_EBS_Policy \
  --policy-document file://kms-key-for-encryption-on-ebs.json
```

c. Adjunte una política de IAM y adjunte la política de IAM al rol con el siguiente comando. Reemplace *111122223333* por su ID de cuenta. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
  --policy-arn
 arn:aws:iam::111122223333:policy/KMS_Key_For_Encryption_On_EBS_Policy \
  --role-name AmazonEKS_EBS_CSI_DriverRole
```

Ahora que ha creado el rol de IAM del controlador de CSI de Amazon EBS, puede continuar con Agregar el controlador CSI de Amazon EBS. Cuando implementa el complemento en ese procedimiento, crea una cuenta de servicio que se llama `ebs-csi-controller-sa` y se configura para utilizarla. La cuenta de servicio está vinculada a un `clusterrole` de Kubernetes denominado, al que se le asignan los permisos de Kubernetes necesarios.

Administración del controlador de CSI de Amazon EBS como complemento de Amazon EKS

Para mejorar la seguridad y reducir la cantidad de trabajo, puede administrar el controlador de CSI de Amazon EBS como un complemento de Amazon EKS. Para obtener más información sobre los complementos de Amazon EKS, consulte Complementos de Amazon EKS. Puede agregar el complemento CSI de Amazon EBS si sigue los pasos descritos en Agregar el controlador CSI de Amazon EBS.

Si ha agregado el complemento CSI de Amazon EBS, puede administrarlo si sigue los pasos de las secciones Actualización del controlador de CSI de Amazon EBS como complemento de Amazon EKS y Eliminar el complemento CSI de Amazon EBS.

Requisitos previos

- Un clúster existente. Para ver la versión de plataforma requerida, ejecute el siguiente comando.

```
aws eks describe-addon-versions --addon-name aws-ebs-csi-driver
```

- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- Un rol de IAM del controlador de CSI de Amazon EBS. Si no cumple este requisito previo, aparecerá `failed to provision volume with StorageClass` con un error `could not create volume in EC2: UnauthorizedOperation` al intentar instalar el complemento y ejecutar `kubectl describe pvc`. Para obtener más información, consulte Creación del rol de IAM del controlador de CSI de Amazon EBS.
- Si usa una `PodSecurityPolicy` restringida a nivel de clúster, asegúrese de que el complemento tenga los permisos suficientes para implementarla. Para ver los permisos que necesita cada Pod del complemento, consulte la definición de manifiesto del complemento pertinente en GitHub.

> **Important**
>
> Para utilizar la funcionalidad de instantáneas del controlador CSI de Amazon EBS, debe instalar el capturador de instantáneas externo antes de instalar el complemento. Los componentes del capturador de instantáneas externo deben instalarse en el siguiente orden:

- `CustomResourceDefinition` (CRD) para `volumesnapshotclasses`, `volumesnapshots` y `volumesnapshotcontents`
- RBAC (`ClusterRole`, `ClusterRoleBinding` y así sucesivamente)
- controlador de implementación

Para obtener más información, consulte Capturador de instantáneas de CSI en GitHub.

Agregar el controlador CSI de Amazon EBS

> **Important**
>
> Antes de añadir el controlador de Amazon EBS como complemento de Amazon EKS, confirme que no tiene una versión autogestionada del controlador instalada en su clúster. Si es así, consulte Desinstalar un controlador CSI autogestionado de Amazon EBS en GitHub.

Puede usar `eksctl`, el AWS Management Console o el AWS CLI para agregar el complemento CSI de Amazon EBS a su clúster.

eksctl

Para agregar el complemento CSI de Amazon EBS con **`eksctl`**

Ejecute el siguiente comando de la . Reemplace *`my-cluster`* por el nombre del clúster, *`111122223333`* por el ID de cuenta y *`AmazonEKS_EBS_CSI_DriverRole`* por el nombre del rol de IAM creado anteriormente. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
eksctl create addon --name aws-ebs-csi-driver --cluster my-cluster --service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole --force
```

Si quita la opción **`--force`** y cualquiera de las configuraciones del complemento de Amazon EKS entran en conflicto con la configuración existente, se produce un error al actualizar el complemento de Amazon EKS y recibe un mensaje de error para ayudarlo a resolver el conflicto. Antes de especificar esta opción, asegúrese de que el complemento de Amazon

EKS no administra la configuración que necesita administrar, ya que dicha configuración se sobrescribe con esta opción. Para obtener más información acerca de otras opciones para este ajuste, consulte Addons (Complementos) en la documentación de `eksctl`. Para obtener más información sobre la administración de campos de Amazon EKS deKubernetes, consulte Administración de campos de Kubernetes.

AWS Management Console

Para agregar el complemento CSI de Amazon EBS con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Seleccione el nombre del clúster para el que desea configurar el complemento CSI de Amazon EBS.
4. Elija la pestaña Complementos.
5. Escoja Obtener más complementos.
6. En la página Seleccionar complementos, haga lo siguiente:
   a. En la sección Complementos de Amazon EKS, seleccione la casilla de verificación Controlador CSI de Amazon EBS.
   b. Elija Siguiente.
7. En la página Configurar las opciones de complementos seleccionados, haga lo siguiente:
   a. Seleccione la Version (Versión) que desea utilizar.
   b. En Seleccionar rol de IAM, seleccione el nombre de un rol de IAM al que ha adjuntado la política de IAM del controlador CSI de Amazon EBS.
   c. (Opcional) Puede ampliar las Valores de configuración opcionales. Si selecciona Anular en Método de resolución de conflictos, una o varias de las configuraciones del complemento existente pueden sobrescribirse con la configuración del complemento de Amazon EKS. Si no habilita esta opción y hay un conflicto con la configuración existente, la operación falla. Puede utilizar el mensaje de error resultante para solucionar el conflicto. Antes de seleccionar esta opción, asegúrese de que el complemento de Amazon EKS no administra las configuraciones que se necesitan autoadministrar.
   d. Elija Siguiente.
8. En la página Revisar y añadir, elija Crear. Una vez finalizada la instalación del complemento, verá el complemento instalado.

AWS CLI

Para agregar el complemento CSI de Amazon EBS con la AWS CLI

Ejecute el siguiente comando de la . Reemplace *my-cluster* por el nombre del clúster, *111122223333* por el ID de cuenta y *AmazonEKS_EBS_CSI_DriverRole* por el nombre del rol creado anteriormente. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

```
aws eks create-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole
```

Ahora que ha agregado el controlador de CSI de Amazon EBS como complemento de Amazon EKS, puede continuar con Implemente una aplicación de muestra y verifique que el controlador CSI funciona. Este procedimiento incluye la configuración de la clase de almacenamiento.

Actualización del controlador de CSI de Amazon EBS como complemento de Amazon EKS

Amazon EKS no actualiza de forma automática CSI de Amazon EBS para el clúster cuando se lanzan versiones nuevas o después de actualizar el clúster a una versión nueva secundaria de Kubernetes. Para actualizar CSI de Amazon EBS en un clúster existente, debe iniciar la actualización y, a continuación, Amazon EKS actualiza el complemento en su nombre.

eksctl

Para actualizar el complemento CSI de Amazon EBS con **eksctl**

1. Verifique la versión actual del complemento CSI de Amazon EBS. Reemplace *my-cluster* por el nombre del clúster.

   ```
   eksctl get addon --name aws-ebs-csi-driver --cluster my-cluster
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME                    VERSION                         STATUS  ISSUES  IAMROLE
    UPDATE AVAILABLE
   ```

```
aws-ebs-csi-driver      v1.11.2-eksbuild.1          ACTIVE  0
  v1.11.4-eksbuild.1
```

2. Actualice el complemento a la versión devuelta en `UPDATE AVAILABLE` en el resultado del paso anterior.

```
eksctl update addon --name aws-ebs-csi-driver --version v1.11.4-eksbuild.1 --
cluster my-cluster \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole --force
```

Si quita la opción **`--force`** y cualquiera de las configuraciones del complemento de Amazon EKS entran en conflicto con la configuración existente, se produce un error al actualizar el complemento de Amazon EKS y recibe un mensaje de error para ayudarlo a resolver el conflicto. Antes de especificar esta opción, asegúrese de que el complemento de Amazon EKS no administra la configuración que necesita administrar, ya que dicha configuración se sobrescribe con esta opción. Para obtener más información acerca de otras opciones para este ajuste, consulte Addons (Complementos) en la documentación de `eksctl`. Para obtener más información sobre la administración de campos de Amazon EKS deKubernetes, consulte Administración de campos de Kubernetes.

AWS Management Console

Para actualizar el complemento CSI de Amazon EBS con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Elija el nombre del clúster para el que desea actualizar el complemento CSI de Amazon EBS.
4. Elija la pestaña Complementos.
5. Elija Controlador CSI de Amazon EBS.
6. Elija Editar.
7. En la página Configurar el controlador CSI de Amazon EBS, haga lo siguiente:
   a. Seleccione la Version (Versión) que desea utilizar.
   b. En Seleccionar rol de IAM, seleccione el nombre de un rol de IAM al que ha adjuntado la política de IAM del controlador CSI de Amazon EBS.

c. (Opcional) Puede ampliar las Valores de configuración opcionales y modificarlas según sea necesario.

d. Elija Guardar cambios.

AWS CLI

Para actualizar el complemento CSI de Amazon EBS con la AWS CLI

1. Verifique la versión actual del complemento CSI de Amazon EBS. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver
 --query "addon.addonVersion" --output text
```

Un ejemplo de salida sería el siguiente.

```
v1.11.2-eksbuild.1
```

2. Determine qué versiones del complemento CSI de Amazon EBS se encuentran disponibles para la versión de su clúster.

```
aws eks describe-addon-versions --addon-name aws-ebs-csi-driver --kubernetes-
version 1.23 \
  --query "addons[].addonVersions[].[addonVersion,
 compatibilities[].defaultVersion]" --output text
```

Un ejemplo de salida sería el siguiente.

```
v1.11.4-eksbuild.1
True
v1.11.2-eksbuild.1
False
```

La versión con `True` debajo es la versión predeterminada que se implementa cuando se crea el complemento. Es posible que la versión que se implementa al crear el complemento no sea la versión más reciente disponible. En el resultado anterior, la versión más reciente se implementa cuando se crea el complemento.

3. Actualice el complemento a la versión que haya devuelto `True` en la salida del paso anterior. También se puede actualizar a una versión posterior si se devuelve en la salida.

```
aws eks update-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver
 --addon-version v1.11.4-eksbuild.1 \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_EBS_CSI_DriverRole --resolve-
conflicts PRESERVE
```

La opción *PRESERVE* conserva cualquier configuración personalizada que haya establecido para el complemento. Para obtener más información acerca de otras opciones para este ajuste, consulte update-addon en la Referencia de la línea de comandos de Amazon EKS. Para obtener más información sobre la administración de la configuración de complementos de Amazon EKS, consulte Administración de campos de Kubernetes.

Eliminar el complemento CSI de Amazon EBS

Tiene dos opciones al eliminar un complemento de Amazon EKS.

- Conservar el software del complemento en el clúster: esta opción elimina la administración de Amazon EKS de cualquier configuración. También elimina la capacidad de Amazon EKS de notificarle las actualizaciones y actualizar de forma automática el complemento de Amazon EKS después de iniciar una actualización. Sin embargo, conserva el software del complemento en el clúster. Esta opción hace que la instalación sea autoadministrada, en lugar de un complemento de Amazon EKS. Con esta opción, no hay tiempo de inactividad para el complemento. Los comandos de este procedimiento utilizan esta opción.
- Eliminar por completo el software del complemento del clúster: recomendamos que elimine el complemento de Amazon EKS del clúster solo si no hay recursos en el clúster que dependan de él. Para hacer esta opción, elimine `--preserve` del comando que utiliza en este procedimiento.

Si el complemento tiene una cuenta de IAM asociada, esta no se elimina.

Puede usar `eksctl`, la AWS Management Console o la AWS CLI para eliminar el complemento CSI de Amazon EBS.

eksctl

Para eliminar el complemento CSI de Amazon EBS con **`eksctl`**

Reemplace *my-cluster* por el nombre del clúster y, a continuación, ejecute el siguiente comando.

```
eksctl delete addon --cluster my-cluster --name aws-ebs-csi-driver --preserve
```

AWS Management Console

Para eliminar el complemento CSI de Amazon EBS con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Elija el nombre del clúster para el que desea eliminar el complemento CSI de Amazon EBS.
4. Elija la pestaña Complementos.
5. Elija Controlador CSI de Amazon EBS.
6. Elija Eliminar.
7. En el cuadro de diálogo de confirmación Eliminar: aws-ebs-csi-driver, haga lo siguiente:
   a. Si desea que Amazon EKS deje de administrar la configuración del complemento, seleccione Conservar en clúster. Haga esto si desea retener el software del complemento en el clúster. Esto es para que pueda administrar todas las configuraciones del complemento por su cuenta.
   b. Escriba **aws-ebs-csi-driver**.
   c. Seleccione Remove (Eliminar).

AWS CLI

Para eliminar el complemento CSI de Amazon EBS con la AWS CLI

Reemplace *my-cluster* por el nombre del clúster y, a continuación, ejecute el siguiente comando.

```
aws eks delete-addon --cluster-name my-cluster --addon-name aws-ebs-csi-driver --
preserve
```

Implemente una aplicación de muestra y verifique que el controlador CSI funciona

Puede probar la funcionalidad del controlador CSI con una aplicación de ejemplo. En este tema se muestra un ejemplo, pero también puede hacer lo siguiente:

- Implemente una aplicación de ejemplo que utiliza el instantáneas de instantáneas de externas para crear instantáneas de volumen. Para obtener más información, consulte Instantáneas de volumen en GitHub.
- Implemente una aplicación de ejemplo que utiliza el cambio de tamaño de volúmenes. Para obtener más información, consulte Cambio de tamaño de volumen en GitHub.

En este procedimiento, se utiliza el ejemplo de aprovisionamiento de volúmenes dinámico del repositorio de GitHub del controlador de Amazon EBS Container Storage Interface (CSI) para utilizar un volumen de Amazon EBS aprovisionado de manera dinámica.

1. Clone el repositorio de GitHub del controlador de Amazon EBS Container Storage Interface (CSI) en su sistema local.

   ```
   git clone https://github.com/kubernetes-sigs/aws-ebs-csi-driver.git
   ```

2. Vaya al directorio de ejemplo `dynamic-provisioning`.

   ```
   cd aws-ebs-csi-driver/examples/kubernetes/dynamic-provisioning/
   ```

3. (Opcional) El archivo `manifests/storageclass.yaml` aprovisiona los volúmenes gp2 de Amazon EBS de forma predeterminada. Para usar volúmenes `gp3` en su lugar, agregue `type: gp3` a `manifests/storageclass.yaml`.

   ```
   echo "parameters:
     type: gp3" >> manifests/storageclass.yaml
   ```

4. Implemente la clase de almacenamiento `ebs-sc`, la notificación de volumen `ebs-claim` persistente y la aplicación de muestra `app` desde el directorio `manifests`.

   ```
   kubectl apply -f manifests/
   ```

5. Describa la clase de almacenamiento `ebs-sc`.

```
kubectl describe storageclass ebs-sc
```

Un ejemplo de salida sería el siguiente.

```
Name:            ebs-sc
IsDefaultClass:  No
Annotations:     kubectl.kubernetes.io/last-applied-
configuration={"apiVersion":"storage.k8s.io/v1","kind":"StorageClass","metadata":
{"annotations":{},"name":"ebs-
sc"},"provisioner":"ebs.csi.aws.com","volumeBindingMode":"WaitForFirstConsumer"}

Provisioner:           ebs.csi.aws.com
Parameters:            <none>
AllowVolumeExpansion:  <unset>
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     WaitForFirstConsumer
Events:                <none>
```

> **Note**
>
> La clase de almacenamiento utiliza el modo de enlace de volumen `WaitForFirstConsumer`. Esto quiere decir que los volúmenes no se aprovisionan de forma dinámica hasta que un Pod realiza una afirmación de volumen persistente. Para obtener más información, consulte Volume Binding Mode en la documentación de Kubernetes.

6. Vea los Pods en el espacio de nombres predeterminado. Transcurridos unos minutos, el estado del Pod de la `app` cambia a `Running`.

```
kubectl get pods --watch
```

Ingrese `Ctrl+C` para obtener un símbolo de sistema del shell.

7. Enumere los volúmenes persistentes en el espacio de nombres predeterminado. Busque un volumen persistente con la notificación `default/ebs-claim`.

```
kubectl get pv
```

Un ejemplo de salida sería el siguiente.

```
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY
   STATUS   CLAIM               STORAGECLASS   REASON   AGE
pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a   4Gi        RWO            Delete
   Bound    default/ebs-claim   ebs-sc                  30s
```

8. Describa el volumen persistente. Reemplace `pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a` por el valor de la salida del paso anterior.

```
kubectl describe pv pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a
```

Un ejemplo de salida sería el siguiente.

```
Name:              pvc-37717cd6-d0dc-11e9-b17f-06fad4858a5a
Labels:            <none>
Annotations:       pv.kubernetes.io/provisioned-by: ebs.csi.aws.com
Finalizers:        [kubernetes.io/pv-protection external-attacher/ebs-csi-aws-com]
StorageClass:      ebs-sc
Status:            Bound
Claim:             default/ebs-claim
Reclaim Policy:    Delete
Access Modes:      RWO
VolumeMode:        Filesystem
Capacity:          4Gi
Node Affinity:
  Required Terms:
    Term 0:        topology.ebs.csi.aws.com/zone in [region-code]
Message:
Source:
    Type:              CSI (a Container Storage Interface (CSI) volume source)
    Driver:            ebs.csi.aws.com
    VolumeHandle:      vol-0d651e157c6d93445
    ReadOnly:          false
    VolumeAttributes:      storage.kubernetes.io/
csiProvisionerIdentity=1567792483192-8081-ebs.csi.aws.com
Events:                <none>
```

El ID de volumen de Amazon EBS es el valor de `VolumeHandle` en la salida anterior.

9. Compruebe que el Pod escribe los datos en el volumen.

```
kubectl exec -it app -- cat /data/out.txt
```

Un ejemplo de salida sería el siguiente.

```
Wed May 5 16:17:03 UTC 2021
Wed May 5 16:17:08 UTC 2021
Wed May 5 16:17:13 UTC 2021
Wed May 5 16:17:18 UTC 2021
[...]
```

10. Una vez terminado, elimine los recursos de esta aplicación de muestra.

```
kubectl delete -f manifests/
```

Preguntas frecuentes sobre migración de CSI de Amazon EBS

> **Important**
>
> Si tiene Pods ejecutándose en una versión `1.22` o un clúster anterior, debe instalar el Controlador de Amazon EBS CSI antes de actualizar su clúster a la versión `1.23` para evitar la interrupción del servicio.

La característica de migración de la interfaz de almacenamiento en contenedores (CSI) de Amazon EBS transfiere la responsabilidad de administrar las operaciones de almacenamiento del aprovisionador de almacenamiento EBS integrado en árbol de Amazon EBS al controlador CSI de Amazon EBS.

¿Qué son los controladores CSI?

Los controladores CSI:

- sustituyen los controladores de almacenamiento “en árbol” de Kubernetes que existen en el código de origen del proyecto de Kubernetes.
- funcionan con proveedores de almacenamiento, como Amazon EBS.

- proporcionan un modelo de complemento simplificado que facilita las cosas a los proveedores de almacenamiento como AWS para lanzar características y mantener la compatibilidad sin depender del ciclo de lanzamiento de Kubernetes.

Para obtener más información, consulte Introducción en la documentación de CSI de Kubernetes.

¿Qué es la migración de CSI?

La característica de migración de CSI de Kubernetes traslada la responsabilidad de administrar las operaciones de almacenamiento de los complementos de almacenamiento existentes en árbol, como `kubernetes.io/aws-ebs`, a los controladores CSI correspondientes. Los objetos `StorageClass`, `PersistentVolume` y `PersistentVolumeClaim` (PVC) existentes siguen funcionando siempre que esté instalado el controlador CSI correspondiente. Cuando la característica está habilitada:

- Las cargas de trabajo existentes que utilizan PVC siguen funcionando como siempre lo han hecho.
- Kubernetes pasa el control de todas las operaciones de administración de almacenamiento a los controladores de CSI.

Para obtener más información, consulte Kubernetes1.23: Actualización del estado de migración del volumen de en árbol a CSI de Kubernetes en el blog Kubernetes.

Para ayudarle a migrar del complemento en árbol a los controladores de CSI, los marcadores `CSIMigration` y `CSIMigrationAWS` se encuentran habilitadas de forma predeterminada en Amazon EKS versión `1.23` y clústeres posteriores. Estos marcadores permiten que el clúster traduzca las API del árbol a sus API de CSI equivalentes. Estos marcadores están colocados en el plano de control de Kubernetes administrado por Amazon EKS y en los ajustes de `kubelet` configurados en AMI optimizadas para Amazon EKS. Si tiene Pods con volúmenes de Amazon EBS en su clúster, debe instalar el controlador CSI de Amazon EBS antes de actualizar el clúster a la versión **`1.23`**. Si no lo hace, es posible que las operaciones de volumen, como el aprovisionamiento y el montaje, no funcionen según lo esperado. Para obtener más información, consulte Controlador CSI de Amazon EBS.

> **Note**
>
> El aprovisionador `StorageClass` en árbol se llama `kubernetes.io/aws-ebs`. El aprovisionador `StorageClass` de CSI de Amazon EBS se llama `ebs.csi.aws.com`.

¿Puedo montar volúmenes **kubernetes.io/aws-ebs StorageClass** en la versión **1.23** y clústeres posteriores?

Sí, siempre y cuando Controlador de CSI de Amazon EBSesté instalado. Para la versión recién creada `1.23` y los clústeres posteriores, se recomienda instalar el controlador CSI de Amazon EBS como parte del proceso de creación del clúster. También recomendamos usar únicamente `StorageClasses` basado en el aprovisionador `ebs.csi.aws.com`.

Si ha actualizado el plano de control del clúster a la versión `1.23` y aún no ha actualizado sus nodos a `1.23`, a continuación, los marcadores `CSIMigration` y `CSIMigrationAWS` de `kubelet` no están habilitados. En este caso, el controlador integrado en el árbol se utiliza para montar los volúmenes basados en `kubernetes.io/aws-ebs`. Sin embargo, el controlador CSI de Amazon EBS debe estar instalado para garantizar que los Pods con volúmenes basados en `kubernetes.io/aws-ebs` se puedan programar. El controlador también es necesario para que otras operaciones de volumen se realicen correctamente.

¿Puedo aprovisionar volumen **kubernetes.io/aws-ebs StorageClass** en Amazon EKS **1.23** y los clústeres posteriores?

Sí, siempre y cuando el controlador de CSI de Amazon EBS esté instalado.

¿El aprovisionador **kubernetes.io/aws-ebs StorageClass** se ha eliminado de Amazon EKS?

El aprovisionador `StorageClass` de `kubernetes.io/aws-ebs` y el tipo de volumen `awsElasticBlockStore` ya no reciben soporte, pero no hay planes de eliminarlos. Estos recursos se tratan como parte de la API de Kubernetes.

¿Cómo instalo el controlador de CSI de Amazon EBS?

Le recomendamos que instale el controlador CSI de Amazon EBS del complemento Amazon EKS. Cuando se requiere una actualización para el complemento de Amazon EKS, inicia la actualización y Amazon EKS actualiza el complemento en su nombre. Si quiere administrar el controlador, puede instalarlo mediante el gráfico de Helm de código abierto.

> **Important**
>
> El controlador de Amazon EBS en árbol de Kubernetes se encuentra en el plano de control de Kubernetes. Utiliza los permisos de IAM asignados al Rol de IAM del clúster de Amazon

EKS para aprovisionar los volúmenes de Amazon EBS. El controlador de CSI de Amazon EBS se ejecuta en nodos. El controlador necesita permisos de IAM para aprovisionar volúmenes. Para obtener más información, consulte Creación del rol de IAM del controlador de CSI de Amazon EBS.

¿Cómo puedo comprobar si el controlador CSI de Amazon EBS está instalado en mi clúster?

Para determinar si el controlador está instalado en su clúster, ejecute el siguiente comando:

```
kubectl get csidriver ebs.csi.aws.com
```

Para comprobar si la instalación está gestionada por Amazon EKS, ejecute el siguiente comando:

```
aws eks list-addons --cluster-name my-cluster
```

¿Impedirá Amazon EKS una actualización del clúster a la versión **1.23** si aún no he instalado el controlador CSI de Amazon EBS?

No.

¿Qué pasa si me olvido de instalar el controlador CSI de Amazon EBS antes de actualizar mi clúster a la versión 1.23? ¿Puedo instalar el controlador después de actualizar el clúster?

Sí, pero las operaciones de volumen que requieren el controlador CSI de Amazon EBS fallarán después de la actualización del clúster hasta que se instale el controlador.

¿Cuál es el **StorageClass** predeterminado aplicado en la versión **1.23** recién creada de Amazon EKS y clústeres posteriores?

El comportamiento `StorageClass` predeterminado permanece sin cambios. Con cada clúster nuevo, Amazon EKS aplica un `StorageClass` basado en `kubernetes.io/aws-ebs` llamado gp2. No planeamos quitar nunca este `StorageClass` de los clústeres recién creados. Separado del clúster predeterminado `StorageClass`, si crea un `StorageClass` basado en `ebs.csi.aws.com`

sin especificar un tipo de volumen, el controlador CSI de Amazon EBS volverá de forma predeterminada a usar `gp3`.

¿Amazon EKS realizará cambios en las **`StorageClasses`** ya presentes en mi clúster existente cuando actualice mi clúster a la versión **`1.23`**?

No.

¿Cómo puedo migrar un volumen persistente desde **`kubernetes.io/aws-ebsStorageClass`** a **`ebs.csi.aws.com`** usando instantáneas?

Para migrar un volumen persistente, consulte Migración de clústeres de Amazon EKS de volúmenes de EBS gp2 a gp3 en el blog de AWS.

¿Cómo modifico un volumen de Amazon EBS mediante anotaciones?

Para empezar `aws-ebs-csi-driverv1.19.0-eksbuild.2`, puede modificar los volúmenes de Amazon EBS mediante anotaciones dentro de sus `PersistentVolumeClaim` (PVC). La nueva característica de modificación de volumen se implementa como un sidecar adicional, llamado `volumemodifier`. Para obtener más información, consulte Simplificación de la migración y modificación de volúmenes de Amazon EBS en Kubernetes mediante el controlador CSI de EBS en el blog de AWS.

¿Se admite la migración de cargas de trabajo de Windows?

Sí. Si va a instalar el controlador CSI de Amazon EBS mediante el gráfico de Helm de código abierto, establezca `node.enableWindows` a `true`. Este se encuentra establecido de forma predeterminada si se encuentra instalado el controlador CSI de Amazon EBS como complemento de Amazon EKS. Al crear `StorageClasses`, defina el `fsType` en un sistema de archivos Windows, como `ntfs`. Las operaciones de volumen para las cargas de trabajo de Windows se migran al controlador CSI de Amazon EBS de la misma manera que para las cargas de trabajo de Linux.

Controlador CSI de Amazon EFS

Amazon Elastic File System (Amazon EFS) proporciona un almacenamiento de archivos totalmente elástico y sin servidor para que pueda compartir datos de archivos sin aprovisionar ni administrar la capacidad de almacenamiento ni el rendimiento. El controlador de la interfaz de almacenamiento de contenedores (CSI) de Amazon EFS proporciona una interfaz CSI que permite a los clústeres

de Kubernetes que se ejecutan en AWS administrar el ciclo de vida de los sistemas de archivos de Amazon EFS. En este tema se muestra cómo implementar el controlador CSI de Amazon EFS en su clúster de Amazon EKS.

Consideraciones

- El controlador CSI de Amazon EFS no es compatible con imágenes de contenedor basadas en Windows.
- No se puede utilizar el aprovisonamiento dinámico de volúmenes persistentes con los nodos de Fargate, pero sí se puede utilizar el aprovisonamiento estático.
- El aprovisionamiento dinámico requiere el controlador `1.2` o uno posterior. Puede utilizar el aprovisonamiento estático para volúmenes persistentes con la versión `1.1` del controlador en cualquier versión de clúster de Amazon EKS compatible.
- La versión `1.3.2` y posteriores de este controlador admiten la arquitectura Arm64, incluidas las instancias basadas en Graviton de Amazon EC2.
- La versión `1.4.2` o posterior de este controlador admite el uso de FIPS para montar sistemas de archivos.
- Tome nota de las cuotas de recursos de Amazon EFS. Por ejemplo, hay una cuota de 1000 puntos de acceso que se pueden crear para cada sistema de archivos de Amazon EFS. Para obtener más información, consulte las cuotas de recursos de Amazon EFS que no puede cambiar.

Requisitos previos

- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

> **Note**
>
> Un Pod que se ejecuta en AWS Fargate monta automáticamente un sistema de archivos de Amazon EFS.

Creación de un rol de IAM

El controlador CSI de Amazon EFS requiere permisos de IAM para interactuar con el sistema de archivos. Cree un rol de IAM y adjúntelo a la política administrada de AWS requerida. Puede utilizar la `eksctl`, la AWS Management Console o la AWS CLI.

> **Note**
>
> Los pasos específicos de este procedimiento están diseñados para usar el controlador como complemento de Amazon EKS. Para obtener más información sobre las instalaciones autoadministradas, consulte Configurar el permiso del controlador en GitHub.

eksctl

Para crear el rol de IAM del controlador CSI de Amazon EFS con **`eksctl`**

Ejecute los siguientes comandos para crear el rol de IAM. Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`AmazonEKS_EFS_CSI_DriverRole`* con el nombre para su rol.

```
export cluster_name=my-cluster
export role_name=AmazonEKS_EFS_CSI_DriverRole
eksctl create iamserviceaccount \
    --name efs-csi-controller-sa \
    --namespace kube-system \
    --cluster $cluster_name \
```

```
    --role-name $role_name \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEFSCSIDriverPolicy \
    --approve
TRUST_POLICY=$(aws iam get-role --role-name $role_name --query
 'Role.AssumeRolePolicyDocument' | \
    sed -e 's/efs-csi-controller-sa/efs-csi-*/' -e 's/StringEquals/StringLike/')
aws iam update-assume-role-policy --role-name $role_name --policy-document
 "$TRUST_POLICY"
```

AWS Management Console

Cómo crear el rol de IAM del controlador CSI de Amazon EFS con la AWS Management Console

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la página Roles, elija Crear rol.
4. En la página Seleccionar entidad de confianza, haga lo siguiente:
   a. En la sección Tipo de entidad de confianza, elija Identidad web.
   b. En Identity provider (Proveedor de identidades), elija la URL del proveedor de OpenID Connect para su clúster (como se muestra en Overview [Información general] en Amazon EKS).
   c. En Audiencia, elija `sts.amazonaws.com`.
   d. Elija Siguiente.
5. En la página Agregar permisos, haga lo siguiente:
   a. En el cuadro Filtrar políticas, escriba *`AmazonEFSCSIDriverPolicy`*.
   b. Marque la casilla situada a la izquierda del nombre de la *`AmazonEFSCSIDriverPolicy`* que obtuvo en la búsqueda.
   c. Elija Siguiente.
6. En la página Nombrar, revisar y crear, haga lo siguiente:
   a. En Nombre del rol, ingrese un nombre único para su rol, por ejemplo, ***`AmazonEKS_EFS_CSI_DriverRole`***.
   b. En Agregar etiquetas (Opcional), de manera opcional, agregue metadatos al rol asociando etiquetas como pares de clave-valor. Para obtener más información sobre el

uso de etiquetas en IAM, consulte Etiquetado de recursos de IAM en la Guía de usuario de IAM.

c. Elija Crear rol.

7. Una vez creado el rol, seleccione el rol en la consola para abrirlo y editarlo.
8. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.
9. Busque la línea que se parezca a la siguiente:

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud":
 "sts.amazonaws.com"
```

Añada la siguiente línea por encima de la línea anterior. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Reemplace *EXAMPLED539D4633E53DE1B71EXAMPLE* con el ID del proveedor OIDC del clúster.

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub":
 "system:serviceaccount:kube-system:efs-csi-*",
```

10. Modifique el operador `Condition` de `"StringEquals"` a `"StringLike"`.
11. Elija Actualizar política para terminar.

AWS CLI

Para crear el rol de IAM del controlador CSI de Amazon EFS con la AWS CLI

1. Vea la URL del proveedor de OIDC de su clúster. Reemplace *my-cluster* por el nombre de su clúster. Si la salida del comando es `None`, revise los Requisitos previos.

```
aws eks describe-cluster --name my-cluster --query
 "cluster.identity.oidc.issuer" --output text
```

Un ejemplo de salida sería el siguiente.

```
https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
```

2. Cree el rol de IAM que concede la acción `AssumeRoleWithWebIdentity`.

a. Copie el siguiente contenido en un archivo denominado *aws-efs-csi-driver-trust-policy*`.json`. Reemplace *111122223333* por su ID de cuenta. Reemplace

EXAMPLED539D4633E53DE1B71EXAMPLE y *region-code* por los valores que se devolvieron en el paso anterior. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringLike": {
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:efs-csi-*",
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
        }
      }
    }
  ]
}
```

b. Cree el rol. Puede cambiar el nombre de *AmazonEKS_EFS_CSI_DriverRole*, pero si lo hace, asegúrese de cambiarlo también en los pasos posteriores.

```
aws iam create-role \
  --role-name AmazonEKS_EFS_CSI_DriverRole \
  --assume-role-policy-document file://"aws-efs-csi-driver-trust-
policy.json"
```

3. Asocie el comando requerido administrado por AWS al rol con el siguiente comando. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEFSCSIDriverPolicy \
```

```
    --role-name AmazonEKS_EFS_CSI_DriverRole
```

Instalación del controlador CSI de Amazon EFS

Le recomendamos que instale el controlador CSI de Amazon EFS a través del complemento Amazon EKS. Para agregar un complemento de Amazon EKS al clúster, consulte Creación de un complemento. Para obtener más información sobre los complementos, consulte Complementos de Amazon EKS. Si no puede usar el complemento de Amazon EKS, le recomendamos que envíe una pregunta sobre los motivos por los que no puede hacerlo al repositorio de GitHub de la hoja de ruta de contenedores.

Como alternativa, si desea una instalación autogestionada del controlador CSI de Amazon EFS, consulte Instalación en GitHub.

Creación de un sistema de archivos de Amazon EFS

Para crear un sistema de archivos de Amazon EFS, consulte Crear un sistema de archivos de Amazon EFS para Amazon EKS en GitHub.

Implementación de una aplicación de muestra

Puede implementar una variedad de aplicaciones de muestra y modificarlas según sea necesario. Para obtener más información, consulte Ejemplos en GitHub.

Controlador CSI de Amazon FSx para Lustre

El controlador de la interfaz de almacenamiento de contenedores (CSI) de FSx para Lustre proporciona una interfaz CSI que permite a los clústeres de Amazon EKS administrar el ciclo de vida de los sistemas de archivos de FSx para Lustre. Para obtener más información, consulte la Guía del usuario de FSx para Lustre.

En este tema se muestra cómo implementar el controlador de CSI de FSx for Lustre en el clúster de Amazon EKS y verificar que funcione. Siempre recomendamos usar la versión más reciente del controlador. Para ver las versiones disponibles, consulte CSI Specification Compatibility Matrix (Matriz de compatibilidad de especificaciones de CSI) en GitHub.

> **Note**
>
> El controlador no es compatible con Fargate.

Para obtener descripciones detalladas de los parámetros disponibles y ejemplos completos que demuestran las características del controlador, consulte el proyecto Controlador de la interfaz de almacenamiento de contenedores (CSI) de FSx para Lustre en GitHub.

Requisitos previos

Debe tener:

- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para Instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

Los siguientes procedimientos ayudan a crear un clúster de prueba sencillo con el controlador de CSI de FSx para Lustre para que pueda ver cómo funciona. No recomendamos utilizar el clúster de pruebas para cargas de trabajo de producción. Para este tutorial, le recomendamos utilizar el *`example values`*, excepto en los casos en que se indique que los reemplace. Puede reemplazar cualquier *`example value`* al completar los pasos del clúster de producción. Recomendamos

completar todos los pasos del mismo terminal, ya que las variables se establecen y utilizan en todos los pasos y no existirán en terminales diferentes.

Para implementar el controlador de CSI de FSx para Lustre en un clúster de Amazon EKS

1. Establezca algunas variables para utilizarlas en los pasos restantes. Reemplace *my-csi-fsx-cluster* con el nombre del clúster de pruebas que desea crear y *region-code* con la Región de AWS en el que desea crear su clúster de pruebas.

```
export cluster_name=my-csi-fsx-cluster
export region_code=region-code
```

2. Cree un clúster de pruebas.

```
eksctl create cluster \
  --name $cluster_name \
  --region $region_code \
  --with-oidc \
  --ssh-access \
  --ssh-public-key my-key
```

   El aprovisionamiento de clústeres tarda varios minutos. Durante la creación del clúster, verá varias líneas de salida. La última línea de salida es similar a la siguiente línea de ejemplo.

```
[#]  EKS cluster "my-csi-fsx-cluster" in "region-code" region is ready
```

3. Cree una cuenta de servicio de Kubernetes para el controlador y asigne la política administrada de AWS `AmazonFSxFullAccess` a la cuenta de servicio con el siguiente comando. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
eksctl create iamserviceaccount \
  --name fsx-csi-controller-sa \
  --namespace kube-system \
  --cluster $cluster_name \
  --attach-policy-arn arn:aws:iam::aws:policy/AmazonFSxFullAccess \
  --approve \
  --role-name AmazonEKSFSxLustreCSIDriverFullAccess \
  --region $region_code
```

Verá varias líneas de salida a medida que se crea la cuenta de servicio. Las últimas líneas de salida es similar a la siguiente línea de ejemplo.

```
[#]  1 task: {
    2 sequential sub-tasks: {
        create IAM role for serviceaccount "kube-system/fsx-csi-controller-sa",
        create serviceaccount "kube-system/fsx-csi-controller-sa",
    } }
[#]  building iamserviceaccount stack "eksctl-my-csi-fsx-cluster-addon-
iamserviceaccount-kube-system-fsx-csi-controller-sa"
[#]  deploying stack "eksctl-my-csi-fsx-cluster-addon-iamserviceaccount-kube-
system-fsx-csi-controller-sa"
[#]  waiting for CloudFormation stack "eksctl-my-csi-fsx-cluster-addon-
iamserviceaccount-kube-system-fsx-csi-controller-sa"
[#]  created serviceaccount "kube-system/fsx-csi-controller-sa"
```

Apunte el nombre de la pila de AWS CloudFormation que se implementó. En la salida de ejemplo anterior, la pila se denomina `eksctl-my-csi-fsx-cluster-addon-iamserviceaccount-kube-system-fsx-csi-controller-sa`.

4. Implemente el controlador con el siguiente comando. Sustituya *release-X.XX* por la rama que desee. La rama maestra no es compatible porque puede contener características futuras que no sean compatibles con la versión estable del controlador publicada actualmente. Le recomendamos que utilice la versión lanzada más reciente. Para obtener una lista de las ramas activas, consulte aws-fsx-csi-driver en GitHub.

> **Note**
>
> Puede ver el contenido que se está aplicando en aws-fsx-csi-driver en GitHub.

```
kubectl apply -k "github.com/kubernetes-sigs/aws-fsx-csi-driver/deploy/kubernetes/
overlays/stable/?ref=release-X.XX"
```

Un ejemplo de salida sería el siguiente.

```
serviceaccount/fsx-csi-controller-sa created
serviceaccount/fsx-csi-node-sa created
clusterrole.rbac.authorization.k8s.io/fsx-csi-external-provisioner-role created
```

```
clusterrole.rbac.authorization.k8s.io/fsx-external-resizer-role created
clusterrolebinding.rbac.authorization.k8s.io/fsx-csi-external-provisioner-binding
 created
clusterrolebinding.rbac.authorization.k8s.io/fsx-csi-resizer-binding created
deployment.apps/fsx-csi-controller created
daemonset.apps/fsx-csi-node created
csidriver.storage.k8s.io/fsx.csi.aws.com created
```

5. Apunte el ARN para el rol que se creó. Si no lo anotó antes y ya no lo tiene disponible en la salida AWS CLI, puede realizar lo siguiente para verlo en la AWS Management Console.

   a. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.

   b. Asegúrese de que la consola esté configurada en la Región de AWS en la que creó el rol de IAM y, a continuación, seleccione Stacks (Pilas).

   c. Seleccione la pila denominada `eksctl-`*`my-csi-fsx-cluster`*`-addon-iamserviceaccount-kube-system-`*`fsx-csi-controller-sa`*.

   d. Seleccione la pestaña Salidas. El ARN de Role1 aparece en la página Outputs (1) (Salida) (1).

6. Revise la implementación del controlador a fin de agregar la cuenta de servicio que creó anteriormente con el siguiente comando. Reemplace el ARN por el ARN que anotó. Reemplace *`111122223333`* por su ID de cuenta. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`.

   ```
   kubectl annotate serviceaccount -n kube-system fsx-csi-controller-sa \
     eks.amazonaws.com/role-
   arn=arn:aws:iam::111122223333:role/AmazonEKSFSxLustreCSIDriverFullAccess --
   overwrite=true
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   serviceaccount/fsx-csi-controller-sa annotated
   ```

Para implementar un tipo de almacenamiento de Kubernetes, una notificación de volumen persistente y una aplicación de muestra para verificar que el controlador CSI está funcionando

Este procedimiento utiliza el repositorio GitHub del controlador de la interfaz de almacenamiento de contenedores (CSI) de FSx para Lustre para utilizar un volumen de FSx para Lustre aprovisionado dinámicamente.

1. Anote el grupo de seguridad del clúster. Puede verlo en la AWS Management Console en la sección Redes o utilizando el siguiente comando AWS CLI.

   ```
   aws eks describe-cluster --name $cluster_name --query
    cluster.resourcesVpcConfig.clusterSecurityGroupId
   ```

2. Cree un grupo de seguridad para su sistema de archivos Amazon FSx de acuerdo con los criterios que se muestran en Grupos de seguridad de Amazon VPC en la Guía del usuario de Amazon FSx para Lustre. Para la VPC, seleccione la VPC de su clúster tal como se muestra en la sección Networking (Redes). Para “los grupos de seguridad asociados a los clientes de Lustre”, utilice el grupo de seguridad de clúster. Puede dejar solo las reglas de salida para permitir All traffic (Todo el tráfico).
3. Descargue el manifiesto de con el siguiente comando.

   ```
   curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-fsx-csi-driver/
   master/examples/kubernetes/dynamic_provisioning/specs/storageclass.yaml
   ```

4. Edite la sección de parámetros del archivo `storageclass.yaml`. Reemplace cada *example value* con valores propios.

   ```
   parameters:
     subnetId: subnet-0eabfaa81fb22bcaf
     securityGroupIds: sg-068000ccf82dfba88
     deploymentType: PERSISTENT_1
     automaticBackupRetentionDays: "1"
     dailyAutomaticBackupStartTime: "00:00"
     copyTagsToBackups: "true"
     perUnitStorageThroughput: "200"
     dataCompressionType: "NONE"
     weeklyMaintenanceStartTime: "7:09:00"
     fileSystemTypeVersion: "2.12"
   ```

- **`subnetId`**: el ID de subred en el que se debe crear el sistema de archivos de Amazon FSx para Lustre. Amazon FSx para Lustre no se admite en todas las zonas de disponibilidad. Abra la consola de Amazon FSx para Lustre en https://console.aws.amazon.com/fsx/ para confirmar que la subred que desea utilizar se encuentra en una zona de disponibilidad compatible. La subred puede incluir sus nodos o puede ser una subred o VPC diferente:
  - Puede comprobar si hay subredes de nodos en la AWS Management Console seleccionando el grupo de nodos en la sección Compute (Informática).
  - Si la subred que especifica no es la misma en la que tiene los nodos, las VPC deben estar conectadas y debe asegurarse de que tiene abiertos los puertos necesarios en los grupos de seguridad.
- **`securityGroupIds`**: el ID del grupo de seguridad que ha creado para el sistema de archivos.
- **`deploymentType`** (opcional): el tipo de implementación del sistema de archivos. Los valores válidos son `SCRATCH_1`, `SCRATCH_2`, `PERSISTENT_1` y `PERSISTENT_2`. Para obtener más información sobre los tipos de implementación, consulte cómo crear su sistema de archivos de Amazon FSx para Lustre.
- Otros parámetros (opcionales): para obtener información acerca del resto de parámetros, consulte Edit StorageClass (Editar StorageClass) en GitHub.

5. Cree el manifiesto de clase de almacenamiento.

```
kubectl apply -f storageclass.yaml
```

Un ejemplo de salida sería el siguiente.

```
storageclass.storage.k8s.io/fsx-sc created
```

6. Descargue el manifiesto de notificación de volumen persistente.

```
curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-fsx-csi-driver/
master/examples/kubernetes/dynamic_provisioning/specs/claim.yaml
```

7. (Opcional) Edite el archivo `claim.yaml`. Cambie *`1200Gi`* por uno de los valores de incremento que se indican a continuación, en función de los requisitos de almacenamiento y del `deploymentType` que seleccionó en los pasos anteriores.

```
storage: 1200Gi
```

- SCRATCH_2 y PERSISTENT: **1.2 TiB**, **2.4 TiB** o incrementos de 2.4 TiB sobre 2.4 TiB.
- SCRATCH_1: **1.2 TiB**, **2.4 TiB**, **3.6 TiB** o incrementos de 3.6 TiB sobre 3.6 TiB.

8. Cree la notificación de volumen persistente.

```
kubectl apply -f claim.yaml
```

Un ejemplo de salida sería el siguiente.

```
persistentvolumeclaim/fsx-claim created
```

9. Confirme que el sistema de archivos está aprovisionado.

```
kubectl describe pvc
```

Un ejemplo de salida sería el siguiente.

```
Name:          fsx-claim
Namespace:     default
StorageClass:  fsx-sc
Status:        Bound
[...]
```

> **Note**
>
> El `Status` puede aparecer como `Pending` durante 5-10 minutos, antes de cambiar a `Bound`. No continúe con el siguiente paso hasta que el `Status` sea `Bound`. Si el `Status` muestra `Pending` durante más de 10 minutos, utilice los mensajes de advertencia en los `Events` como referencia para abordar cualquier problema.

10. Implemente la aplicación de muestra.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-fsx-csi-driver/master/examples/kubernetes/dynamic_provisioning/specs/pod.yaml
```

11. Verifique que la aplicación de muestra se está ejecutando.

```
kubectl get pods
```

Un ejemplo de salida sería el siguiente.

```
NAME      READY   STATUS              RESTARTS   AGE
fsx-app   1/1     Running             0          8s
```

12. Verifique que la aplicación haya montado correctamente el sistema de archivos.

```
kubectl exec -ti fsx-app -- df -h
```

Un ejemplo de salida sería el siguiente.

```
Filesystem                  Size  Used Avail Use% Mounted on
overlay                      80G  4.0G   77G   5% /
tmpfs                        64M     0   64M   0% /dev
tmpfs                       3.8G     0  3.8G   0% /sys/fs/cgroup
192.0.2.0@tcp:/abcdef01     1.1T  7.8M  1.1T   1% /data
/dev/nvme0n1p1               80G  4.0G   77G   5% /etc/hosts
shm                          64M     0   64M   0% /dev/shm
tmpfs                       6.9G   12K  6.9G   1% /run/secrets/kubernetes.io/
serviceaccount
tmpfs                       3.8G     0  3.8G   0% /proc/acpi
tmpfs                       3.8G     0  3.8G   0% /sys/firmware
```

13. Verifique que la aplicación de muestra haya escrito los datos en el sistema de archivos de FSx para Lustre.

```
kubectl exec -it fsx-app -- ls /data
```

Un ejemplo de salida sería el siguiente.

```
out.txt
```

En este resultado de ejemplo se muestra que la aplicación de ejemplo escribió correctamente el archivo `out.txt` en el sistema de archivos.

> **Note**
>
> Antes de eliminar el clúster, asegúrese de eliminar el sistema de archivos de FSx para Lustre. Para obtener más información, consulte Limpiar recursos en la Guía de usuario de FSx para Lustre.

Controlador de CSI de Amazon FSx para ONTAP de NetApp

NetApp's Astra Trident proporciona una orquestación dinámica del almacenamiento mediante un controlador compatible con la interfaz de almacenamiento de contenedores (CSI). Esto permite a los clústeres de Amazon EKS administrar el ciclo de vida de los volúmenes persistentes (PV) respaldados por sistemas de archivos Amazon FSx para NetApp ONTAP. Para comenzar, consulte Use Astra Trident with Amazon FSx for NetApp ONTAP en la documentación de Astra Trident.

Amazon FSx para NetApp ONTAP es un servicio de almacenamiento que le permite lanzar y ejecutar sistemas de archivos ONTAP completamente administrados en la nube. ONTAP es una tecnología de sistema de archivos de NetApp's que proporciona un conjunto ampliamente adoptado de capacidades de administración y de acceso a datos. FSx para ONTAP proporciona las características, el rendimiento y las API de los sistemas de archivos de NetApp en las instalaciones con la agilidad, escalabilidad y simplicidad de un servicio de completamente administrado de AWS. Para obtener más información, consulte la Guía del usuario de FSx para ONTAP.

Controlador de CSI de Amazon FSx para OpenZFS

Amazon FSx para OpenZFS es un servicio de almacenamiento de archivos completamente administrado que facilita el traslado de datos a AWS desde ZFS en las instalaciones u otros servidores de archivos basados en Linux. Puede hacerlo sin cambiar el código de la aplicación ni la forma en que administra los datos. Ofrece almacenamiento de archivos altamente confiable, escalable, eficiente y rico en características construido en el sistema de archivos OpenZFS de código abierto. Combina estas capacidades con la agilidad, la escalabilidad y la simplicidad de un servicio de AWS totalmente gestionado. Para más información, consulte la Guía del usuario de Amazon FSx para OpenZFS.

El controlador de la interfaz de almacenamiento de contenedores (CSI) de Amazon FSx para OpenZFS proporciona una interfaz CSI que permite que los clústeres de Amazon EKS administren el ciclo de vida de los volúmenes de Amazon FSx para OpenZFS. Para implementar el controlador CSI

de Amazon FSx para OpenZFS en su clúster de Amazon EKS, consulte `aws-fsx-openzfs-csi-driver` en GitHub.

Controlador CSI de Amazon File Cache

Amazon File Cache es una memoria caché de alta velocidad totalmente administrada en AWS que se utiliza para procesar datos de archivos, independientemente de dónde estén almacenados los datos. Amazon File Cache carga automáticamente los datos en la memoria caché cuando se accede a ella por primera vez y publica los datos cuando no se utilizan. Para obtener más información, consulte la Guía del usuario de Amazon File Cache.

El controlador Container Storage Interface (CSI) de Amazon File Cache proporciona una interfaz CSI que permite a los clústeres de Amazon EKS administrar el ciclo de vida del almacenamiento caché de archivos de Amazon. Para implementar el controlador CSI de Amazon File Cache en su clúster de Amazon EKS, consulte `aws-file-cache-csi-driver` en GitHub.

Mountpoint para el controlador CSI de Amazon S3

Con el Mountpoint para el controlador Container Storage Interface (CSI) de Amazon S3, las aplicaciones de Kubernetes pueden acceder a los objetos de Amazon S3 a través de una interfaz de sistema de archivos, lo que permite lograr un rendimiento total alto sin cambiar códigos de aplicaciones. Basado en Mountpoint para Amazon S3, el controlador CSI presenta un bucket de Amazon S3 como un volumen al que se puede acceder mediante contenedores de Amazon EKS y clústeres autoadministrados de Kubernetes. En este tema se muestra cómo implementar el Mountpoint para el controlador CSI para Amazon S3 en su clúster de Amazon EKS.

Consideraciones

- El controlador de CSI del Mountpoint para Amazon S3 no es compatible con imágenes de contenedor basadas en Windows.
- El controlador CSI del Mountpoint para Amazon S3 no es compatible con AWS Fargate. Sin embargo, se admiten los contenedores que se ejecutan en Amazon EC2 (ya sea con Amazon EKS o con una instalación personalizada de Kubernetes).
- El controlador CSI del Mountpoint para Amazon S3 solo admite el aprovisionamiento estático. No se admite el aprovisionamiento dinámico o la creación de nuevos buckets.

> **Note**
>
> El aprovisionamiento estático se refiere al uso de un bucket de Amazon S3 existente que se especifica como `bucketName` en el `volumeAttributes` del objeto `PersistentVolume`. Para obtener más información, consulte Aprovisionamiento estático en GitHub.

- Los volúmenes montados con el controlador CSI del Mountpoint para Amazon S3 no admiten todas las características del sistema de archivos POSIX. Para obtener más información sobre el comportamiento del sistema de archivos, consulte Comportamiento del sistema de archivos del Mountpoint para Amazon S3 en GitHub.

Requisitos previos

- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- La versión 2.12.3 o posterior del AWS CLI instalada y configurada en su dispositivo o AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

Creación de una política de IAM

El controlador CSI del Mountpoint para Amazon S3 requiere permisos de Amazon S3 para interactuar con el sistema de archivos. En esta sección se muestra cómo crear una política de IAM que conceda los permisos necesarios.

El siguiente ejemplo de política sigue las recomendaciones de permisos de IAM para Mountpoint. Como alternativa, puede usar la política administrada de AWS AmazonS3FullAccess, pero esta política administrada otorga más permisos de los necesarios por Mountpoint.

Para obtener más información sobre los permisos recomendados por Mountpoint, consulte Permisos de IAM de Mountpoint en GitHub.

Creación de una política de IAM mediante la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Policies (Políticas).
3. En la página Políticas, seleccione Crear una política.
4. En el editor de políticas, seleccione JSON.
5. En el editor de políticas, copie y pegue lo siguiente:

> **Important**
>
> Reemplace `DOC-EXAMPLE-BUCKET1` por el nombre de su bucket de Amazon S3.

```
{
   "Version": "2012-10-17",
   "Statement": [
        {
            "Sid": "MountpointFullBucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
            ]
        },
        {
            "Sid": "MountpointFullObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:AbortMultipartUpload",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
```

```
            ]
        }
    ]
}
```

Los buckets de directorio, introducidos con la clase de almacenamiento Amazon S3 Express One Zone, utilizan un mecanismo de autenticación diferente al de los buckets de uso general. En lugar de utilizar acciones `s3:*`, debe utilizar la acción `s3express:CreateSession`. Para más información sobre los buckets de directorio, consulte Buckets de directorio en la Guía de usuario de Amazon S3.

A continuación, se muestra un ejemplo de política de privilegios mínimos que utilizaría para un bucket de directorios.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3express:CreateSession",
            "Resource": "arn:aws:s3express:aws-region:111122223333:bucket/DOC-
EXAMPLE-BUCKET1--az_id--x-s3"
        }
    ]
}
```

6. Elija Siguiente.
7. En la página Revisar y crear, asigne un nombre a la política. En este tutorial de ejemplo se utiliza el nombre `AmazonS3CSIDriverPolicy`.
8. Elija Crear política.

Creación de un rol de IAM

El controlador CSI del Mountpoint para Amazon S3 requiere permisos de Amazon S3 para interactuar con el sistema de archivos. En esta sección se muestra cómo crear un rol de IAM para delegar estos permisos. Puede crear este rol, puede utilizar `eksctl`, la consola de IAM o la AWS CLI.

> **Note**
>
> La política de IAM `AmazonS3CSIDriverPolicy` se creó en la sección anterior.

eksctl

Para crear el Mountpoint para rol de IAM del controlador CSI de Amazon S3 con **`eksctl`**

Para crear el rol de IAM y la cuenta de servicio de Kubernetes ejecute los siguientes comandos. Estos comandos también asocian la política de IAM `AmazonS3CSIDriverPolicy` al rol, anotan la cuentas de servicio de Kubernetes (`s3-csi-controller-sa`) con el Nombre de recurso de Amazon (ARN) del rol de IAM y agregan el nombre de la cuenta de servicio de Kubernetes a la política de confianza correspondiente al rol de IAM.

```
CLUSTER_NAME=my-cluster
REGION=region-code
ROLE_NAME=AmazonEKS_S3_CSI_DriverRole
POLICY_ARN=AmazonEKS_S3_CSI_DriverRole_ARN
eksctl create iamserviceaccount \
    --name s3-csi-driver-sa \
    --namespace kube-system \
    --cluster $CLUSTER_NAME \
    --attach-policy-arn $POLICY_ARN \
    --approve \
    --role-name $ROLE_NAME \
    --region $REGION \
    --role-only
```

IAM console

Para crear el Mountpoint para rol de IAM del controlador CSI de Amazon S3 con la AWS Management Console

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la página Roles, elija Crear rol.
4. En la página Seleccionar entidad de confianza, haga lo siguiente:

    a. En la sección Tipo de entidad de confianza, elija Identidad web.

b. En Proveedor de identidades, elija la URL del proveedor OpenID Connect para su clúster (como se muestra en Información general en Amazon EKS).

   Si no se muestra ninguna URL, consulte la sección Requisitos previos.

c. En Audiencia, elija `sts.amazonaws.com`.

d. Elija Siguiente.

5. En la página Agregar permisos, haga lo siguiente:

   a. En el cuadro Filtrar políticas, escriba **`AmazonS3CSIDriverPolicy`**.

   > **Note**
   >
   > Esta política se creó en la sección anterior.

   b. Marque la casilla situada a la izquierda del resultado de `AmazonS3CSIDriverPolicy` que obtuvo en la búsqueda.

   c. Elija Siguiente.

6. En la página Nombrar, revisar y crear, haga lo siguiente:

   a. En Nombre del rol, ingrese un nombre único para su rol, por ejemplo, **`AmazonEKS_S3_CSI_DriverRole`**.

   b. En Agregar etiquetas (Opcional), de manera opcional, agregue metadatos al rol asociando etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetado de recursos de IAM en la Guía de usuario de IAM.

   c. Elija Crear rol.

7. Una vez creado el rol, seleccione el rol en la consola para abrirlo y editarlo.

8. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.

9. Busque la línea que se parezca a la siguiente:

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud":
 "sts.amazonaws.com"
```

   Agregue una coma al final de la línea anterior y, luego, agregue la siguiente línea después de esta. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster.

Reemplace *EXAMPLED539D4633E53DE1B71EXAMPLE* con el ID del proveedor OIDC del clúster.

```
"oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub":
 "system:serviceaccount:kube-system:s3-csi-*"
```

10. Cambie el operador `Condition` de `"StringEquals"` a `"StringLike"`.
11. Elija Actualizar política para terminar.

AWS CLI

Cómo crear el Mountpoint para el rol de IAM del controlador CSI de Amazon S3 con la AWS CLI

1. Vea la URL del proveedor de OIDC para su clúster. Reemplace *my-cluster* por el nombre del clúster. Si la salida del comando es `None`, revise los Requisitos previos.

```
aws eks describe-cluster --name my-cluster --query
 "cluster.identity.oidc.issuer" --output text
```

Un ejemplo de salida sería el siguiente.

```
https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
```

2. Cree el rol de IAM otorgándole a la cuenta de servicio de Kubernetes la acción `AssumeRoleWithWebIdentity`.

   a. Copie el siguiente contenido en un archivo denominado *aws-s3-csi-driver-trust-policy*`.json`. Reemplace *111122223333* por su ID de cuenta. Reemplace *EXAMPLED539D4633E53DE1B71EXAMPLE* y *region-code* por los valores que se devolvieron en el paso anterior.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
```

```
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringLike": {
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:s3-csi-*",
          "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
        }
      }
    }
  ]
}
```

b. Cree el rol. Puede cambiar el nombre de *AmazonEKS_S3_CSI_DriverRole*, pero si lo hace, asegúrese de cambiarlo también en los pasos posteriores.

```
aws iam create-role \
  --role-name AmazonEKS_S3_CSI_DriverRole \
  --assume-role-policy-document file://"aws-s3-csi-driver-trust-policy.json"
```

3. Cree un rol de IAM y adjunte la política de IAM creada previamente al rol con el siguiente comando.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonS3CSIDriverPolicy \
  --role-name AmazonEKS_S3_CSI_DriverRole
```

> **Note**
>
> La política de IAM `AmazonS3CSIDriverPolicy` se creó en la sección anterior.

4. Omita este paso si va a instalar el controlador como complemento de Amazon EKS. Para las instalaciones autogestionadas del controlador, cree cuentas de servicio de Kubernetes que estén anotadas con el ARN del rol de IAM que creó.

a. Guarde los siguientes contenidos en un archivo llamado *mountpoint-s3-service-account*`.yaml`. Reemplace *111122223333* por su ID de cuenta.

```
---
apiVersion: v1
```

```
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/name: aws-mountpoint-s3-csi-driver
  name: mountpoint-s3-csi-controller-sa
  namespace: kube-system
  annotations:
    eks.amazonaws.com/role-arn:
 arn:aws:iam::111122223333:role/AmazonEKS_S3_CSI_DriverRole
```

b. Cree la cuenta de servicio Kubernetes en el clúster. La cuenta de servicio de Kubernetes (`mountpoint-s3-csi-controller-sa`) está anotada con el rol de IAM que creó con el nombre *AmazonEKS_S3_CSI_DriverRole*.

```
kubectl apply -f mountpoint-s3-service-account.yaml
```

> **Note**
>
> Cuando implementa el complemento en este procedimiento, crea una cuenta de servicio que se llama `s3-csi-driver-sa` y se configura para utilizarla.

Instalación del Mountpoint para el controlador CSI de Amazon S3

Puede instalar el Mountpoint para el controlador CSI de Amazon S3 a través del complemento Amazon EKS. Puede usar `eksctl`, el AWS Management Console o el AWS CLI para agregar el complemento al clúster.

Si lo desea, puede instalar Mountpoint para el controlador CSI de Amazon S3 como una instalación autoadministrada. Para obtener instrucciones sobre cómo realizar una instalación autoadministrada, consulte Instalación en GitHub.

eksctl

Agregar el complemento CSI de Amazon S3 con **`eksctl`**

Ejecute el siguiente comando de la . Reemplace *my-cluster* por el nombre del clúster, *111122223333* por el ID de cuenta y *AmazonEKS_S3_CSI_DriverRole* por el nombre del rol de IAM creado anteriormente.

```
eksctl create addon --name aws-mountpoint-s3-csi-driver --cluster my-cluster --
service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKS_S3_CSI_DriverRole
 --force
```

Si quita la opción **--force** y cualquiera de las configuraciones del complemento de Amazon EKS entran en conflicto con la configuración existente, se produce un error al actualizar el complemento de Amazon EKS y recibe un mensaje de error para ayudarlo a resolver el conflicto. Antes de especificar esta opción, asegúrese de que el complemento de Amazon EKS no administra la configuración que necesita administrar, ya que dicha configuración se sobrescribe con esta opción. Para obtener más información acerca de otras opciones para este ajuste, consulte Addons (Complementos) en la documentación de `eksctl`. Para obtener más información sobre la administración de campos de Amazon EKS deKubernetes, consulte Administración de campos de Kubernetes.

AWS Management Console

Agregar el complemento Mountpoint para Amazon S3 CSI con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Seleccione el nombre del clúster para el cual desea configurar el complemento Mountpoint para Amazon S3 CSI.
4. Elija la pestaña Complementos.
5. Escoja Obtener más complementos.
6. En la página Seleccionar complementos, haga lo siguiente:
   a. En la sección Complementos de Amazon EKS, seleccione la casilla de verificación Mountpoint para el controlador de Amazon S3 CSI.
   b. Elija Siguiente.
7. En la página Configurar las opciones de complementos seleccionados, haga lo siguiente:
   a. Seleccione la Version (Versión) que desea utilizar.
   b. En Seleccionar rol de IAM, seleccione el nombre de un rol de IAM al que le haya adjuntado la política de IAM de Mountpoint para el controlador de Amazon S3 CSI.

c. (Opcional) Puede ampliar las Valores de configuración opcionales. Si selecciona Anular en Método de resolución de conflictos, una o varias de las configuraciones del complemento existente pueden sobrescribirse con la configuración del complemento de Amazon EKS. Si no habilita esta opción y hay un conflicto con la configuración existente, la operación falla. Puede utilizar el mensaje de error resultante para solucionar el conflicto. Antes de seleccionar esta opción, asegúrese de que el complemento de Amazon EKS no administra las configuraciones que se necesitan autoadministrar.

d. Elija Siguiente.

8. En la página Revisar y añadir, elija Crear. Una vez finalizada la instalación del complemento, verá el complemento instalado.

AWS CLI

Agregar el complemento Mountpoint para Amazon S3 CSI con la AWS CLI

Ejecute el siguiente comando de la . Reemplace *my-cluster* por el nombre del clúster, *111122223333* por el ID de cuenta y *AmazonEKS_S3_CSI_DriverRole* por el nombre del rol creado anteriormente.

```
aws eks create-addon --cluster-name my-cluster --addon-name aws-mountpoint-s3-csi-
driver \
  --service-account-role-arn
 arn:aws:iam::111122223333:role/AmazonEKS_S3_CSI_DriverRole
```

Configuración de Mountpoint para Amazon S3

En la mayoría de los casos, puede configurar Mountpoint para Amazon S3 solo con un nombre de bucket. Para obtener instrucciones sobre la configuración de Mountpoint para Amazon S3, consulte Configuración de Mountpoint para Amazon S3 en GitHub.

Implementación de una aplicación de muestra

Puede implementar el aprovisionamiento estático en el controlador de un bucket de Amazon S3 existente. Para obtener más información, consulte Static provisioning en GitHub.

Eliminar el controlador Mountpoint para Amazon S3 CSI

Tiene dos opciones al eliminar un complemento de Amazon EKS.

- Conservar el software del complemento en el clúster: esta opción elimina la administración de Amazon EKS de cualquier configuración. También elimina la capacidad de Amazon EKS de notificarle las actualizaciones y actualizar de forma automática el complemento de Amazon EKS después de iniciar una actualización. Sin embargo, conserva el software del complemento en el clúster. Esta opción hace que la instalación sea autoadministrada, en lugar de un complemento de Amazon EKS. Con esta opción, no hay tiempo de inactividad para el complemento. Los comandos de este procedimiento utilizan esta opción.
- Eliminar por completo el software del complemento del clúster: recomendamos que elimine el complemento de Amazon EKS del clúster solo si no hay recursos en el clúster que dependan de él. Para hacer esta opción, elimine `--preserve` del comando que utiliza en este procedimiento.

Si el complemento tiene una cuenta de IAM asociada, esta no se elimina.

Puede usar `eksctl`, la AWS Management Console o la AWS CLI para eliminar el complemento de Amazon S3 CSI.

eksctl

Eliminar el complemento Amazon S3 CSI con **`eksctl`**

Reemplace *`my-cluster`* por el nombre del clúster y, a continuación, ejecute el siguiente comando.

```
eksctl delete addon --cluster my-cluster --name aws-mountpoint-s3-csi-driver --
preserve
```

AWS Management Console

Eliminar el complemento Amazon S3 CSI con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Elija el nombre del clúster para el que desea eliminar el complemento CSI de Amazon EBS.
4. Elija la pestaña Complementos.
5. Seleccione Mountpoint para el controlador de Amazon S3 CSI.
6. Elija Eliminar.
7. En el cuadro de diálogo de confirmación Eliminar: aws-mountpoint-s3-csi-driver, haga lo siguiente:

a. Si desea que Amazon EKS deje de administrar la configuración del complemento, seleccione Conservar en clúster. Haga esto si desea retener el software del complemento en el clúster. Esto es para que pueda administrar todas las configuraciones del complemento por su cuenta.
b. Escriba **`aws-mountpoint-s3-csi-driver`**.
c. Seleccione Eliminar.

AWS CLI

Eliminar el complemento Amazon S3 CSI con la AWS CLI

Reemplace *`my-cluster`* por el nombre del clúster y, a continuación, ejecute el siguiente comando.

```
aws eks delete-addon --cluster-name my-cluster --addon-name aws-mountpoint-s3-csi-driver --preserve
```

Controlador de instantáneas CSI

El controlador de instantáneas de la interfaz de almacenamiento de contenedores (CSI) permite el uso de la funcionalidad de capturas de instantáneas en controladores CSI compatibles, como el controlador CSI de Amazon EBS.

Aquí se incluyen algunos aspectos que debe tener en cuenta cuando se utiliza el controlador de instantáneas CSI.

- El controlador de instantáneas debe instalarse junto con un controlador CSI con funcionalidad de capturas de instantáneas. El controlador CSI de Amazon EBS permite crear instantáneas de Amazon EBS de los volúmenes gestionados por CSI de Amazon EBS. Para obtener las instrucciones de instalación, consulte Controlador CSI de Amazon EBS.
- Kubernetes no admite instantáneas de volúmenes que se entreguen mediante la migración de CSI, como los volúmenes de Amazon EBS que utilizan un `StorageClass` con aprovisionador `kubernetes.io/aws-ebs`. Los volúmenes deben crearse con un `StorageClass` que haga referencia al aprovisionador de controladores CSI `ebs.csi.aws.com`. Para obtener más información sobre la migración de CSI, consulte Preguntas frecuentes sobre migración de CSI de Amazon EBS.

Le recomendamos que instale el controlador de instantáneas CSI a través del complemento gestionado de Amazon EKS. Para agregar un complemento de Amazon EKS al clúster, consulte Creación de un complemento. Para obtener más información sobre los complementos, consulte Complementos de Amazon EKS.

Como alternativa, si prefiere una instalación autogestionada del controlador de instantáneas CSI de Amazon EBS, consulte Usage en la versión anterior Kubernetes `external-snapshotter` en GitHub.

Redes de Amazon EKS

El clúster de Amazon EKS se crea en una VPC. El complemento de interfaz de red de contenedores (CNI) de Amazon VPC proporciona redes de pod. En este capítulo, se detallan los siguientes temas para obtener más información sobre redes para el clúster.

Temas

Requisitos y consideraciones de Amazon EKS VPC y subred

Al crear un clúster, especifica una VPC y al menos dos subredes que se encuentran en diferentes zonas de disponibilidad. En este tema se proporciona una descripción general de los requisitos y consideraciones específicos de Amazon EKS para la VPC y las subredes que utiliza con el clúster. Si no dispone de una VPC para usar con Amazon EKS, puede crear una utilizando una plantilla de AWS CloudFormation proporcionada por Amazon EKS. Si está creando un clúster local o extendido en AWS Outposts, consulte Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS en lugar de este tema.

Requisitos y consideraciones de la VPC

Al crear un clúster, la VPC que especifique debe cumplir los siguientes requisitos y consideraciones:

- La VPC debe tener un número suficiente de direcciones IP disponibles para el clúster, los nodos y otros recursos de Kubernetes que desee crear. Si la VPC que desea utilizar no tiene un número suficiente de direcciones IP, intente aumentar el número de direcciones IP disponibles.

  Para ello, puede actualizar la configuración del clúster para cambiar las subredes y los grupos de seguridad que utiliza el clúster. Puede actualizar desde la AWS Management Console, la última versión de la AWS CLI, AWS CloudFormation, y `eksctl` versión `v0.164.0-rc.0` o posterior. Es

posible que tenga que hacer esto para proporcionar a las subredes más direcciones IP disponibles con las que poder actualizar correctamente la versión de un clúster.

> **Important**
>
> Todas las subredes que agregue deben estar en el mismo conjunto de zonas de disponibilidad proporcionadas originalmente cuando creó el clúster. Las nuevas subredes deben cumplir todos los demás requisitos; por ejemplo, deben tener suficientes direcciones IP.
>
> Por ejemplo, suponga que creó un clúster y especificó cuatro subredes. En el orden en que las especificó, la primera subred está en la zona de disponibilidad `us-west-2a`, la segunda y tercera subredes están en la zona de disponibilidad `us-west-2b`, y la cuarta subred está en la zona de disponibilidad `us-west-2c`. Si desea cambiar las subredes, debe proporcionar al menos una subred en cada una de las tres zonas de disponibilidad, y las subredes deben estar en la misma VPC que las subredes originales.

Si necesita más direcciones IP de las que tienen los bloques de CIDR de la VPC, puede agregar bloques de CIDR adicionales asociando bloques de enrutamiento entre dominios sin clases (CIDR) adicionales a la VPC. Puede asociar bloques CIDR privados (RFC 1918) y públicos (RFC 1918) a su VPC antes o después de crear el clúster. Un clúster puede tardar hasta cinco horas en reconocer un bloque CIDR asociado a una VPC.

Puede conservar el uso de direcciones IP usando una puerta de enlace de tránsito con una VPC de servicios compartidos. Para obtener más información, consulte VPC aisladas con servicios compartidos y Patrones de conservación de direcciones IP enrutables de VPC de Amazon EKS en una red híbrida.

- Si quiere que Kubernetes asigne direcciones `IPv6` a Pods y servicios, asocie un bloque de CIDR `IPv6` con su VPC. Para obtener más información, consulte Asociar un bloque de CIDR `IPv6` a su VPC en la Guía del usuario de Amazon VPC.
- La VPC debe tener un nombre de host `DNS` y admitir la resolución `DNS`. De lo contrario, los nodos no podrán unirse al clúster. A fin de obtener más información, consulte Atributos de DNS para su VPC en la Guía del usuario de Amazon VPC.
- Es posible que la VPC requiera puntos de conexión de VPC mediante AWS PrivateLink. Para obtener más información, consulte Requisitos y consideraciones de la subred.

Si ha creado un clúster con Kubernetes `1.14` o anterior, Amazon EKS agregó la siguiente etiqueta a la VPC:

Clave	Valor
`kubernetes.io/cluster/` *`my-cluster`*	`owned`

Esta etiqueta solo la utilizó Amazon EKS. Puede quitar la etiqueta sin afectar a sus servicios. No se utiliza con clústeres que son versión `1.15` o posterior.

Requisitos y consideraciones de la subred

Al crear un clúster, Amazon EKS crea de 2 a 4 interfaces de red elásticas en las subredes que usted especifique. Estas interfaces de red permiten la comunicación entre el clúster y la VPC. Estas interfaces de red también habilitan características de Kubernetes como `kubectl exec` y `kubectl logs`. Cada interfaz de red creada con Amazon EKS cuenta con el texto `Amazon EKS` *`cluster-name`* en su descripción.

Amazon EKS puede crear sus interfaces de red en cualquier subred que especifique al crear un clúster. Puede cambiar en qué subredes Amazon EKS crea sus interfaces de red después de crear el clúster. Al actualizar la versión de Kubernetes de un clúster, Amazon EKS elimina las interfaces de red originales que creó y crea nuevas interfaces de red. Estas interfaces de red se pueden crear en las mismas subredes que las interfaces de red originales o en subredes distintas de las interfaces de red originales. Para controlar en qué subredes se crean las interfaces de red, puede limitar el número de subredes especificadas a solo dos al crear un clúster, o bien actualizar las subredes después de crear el clúster.

Requisitos de la subred para los clústeres

Las subredes que especifique al crear o actualizar un clúster deben cumplir los siguientes requisitos:

- Las subredes deben tener al menos seis direcciones IP para que Amazon EKS las utilice. Sin embargo, recomendamos al menos 16 direcciones IP.
- Las subredes no pueden residir en AWS Outposts, AWS Wavelength o una AWS Local Zone. Sin embargo, si los tiene en la VPC, puede implementar nodos autoadministrados y recursos de Kubernetes para este tipo de subredes.
- Las subredes pueden ser públicas o privadas. Sin embargo, le recomendamos que especifique subredes privadas, si es posible. Una subred pública es una subred con una tabla de enrutamiento

que incluye una ruta a una puerta de enlace de Internet, mientras que una subred privada es una subred con tabla de enrutamiento que no incluye ruta a la puerta de enlace de internet.

- Las subredes no pueden residir en las siguientes zonas de disponibilidad:

Región de AWS	Nombres de las regiones	ID de zona de disponibilidad que no están permitidos
`us-east-1`	Este de EE. UU. (Norte de Virginia)	`use1-az3`
`us-west-1`	Oeste de EE. UU. (Norte de California)	`usw1-az2`
`ca-central-1`	Canadá (centro)	`cac1-az3`

Uso de familias de direcciones IP por componente

La siguiente tabla contiene la familia de direcciones IP que utiliza cada componente de Amazon EKS. Puede utilizar una traducción de direcciones de red (NAT) u otro sistema de compatibilidad para conectarse a estos componentes desde las direcciones IP de origen en familias con el valor "No" para una entrada de la tabla.

La funcionalidad puede variar según la configuración IP family (`ipFamily`) del clúster. Esta configuración cambia el tipo de direcciones IP utilizadas para el bloque CIDR que Kubernetes le asigna a Services. Un clúster con el valor de configuración de IPv4 se denomina IPv4 cluster y un clúster con el valor de configuración de IPv6 se denomina IPv6 cluster.

Componente	Solo direcciones **`IPv4`**	Solo direcciones **`IPv6`**	Direcciones de pila doble
Punto de conexión público de la API de EKS	Sí	No	No
Punto de conexión de VPC de la API de EKS	Sí	No	No

Componente	Solo direcciones **IPv4**	Solo direcciones **IPv6**	Direcciones de pila doble
Punto de conexión público de la API de autenticación de EKS	Sí[1]	Sí[1]	Sí[1]
Punto de conexión de VPC de la API de autenticación de EKS	Sí[1]	Sí[1]	Sí[1]
Punto de conexión público del clúster de EKS	Sí	No	No
Punto de conexión privado del clúster de EKS	Sí[2]	Sí[2]	No
Subredes del clúster de EKS	Sí[2]	No	Sí[2]
Direcciones IP principales del nodo	Sí[2]	No	Sí[2]
Rango de CIDR del clúster para direcciones IP de Service	Sí[2]	Sí[2]	No
Direcciones IP del Pod del CNI de VPC	Sí[2]	Sí[2]	No

Note

[1] El punto de conexión es de pila doble con ambas direcciones, `IPv4` y `IPv6`. Las aplicaciones fuera de AWS, los nodos del clúster y los pods dentro del clúster pueden alcanzar este punto de conexión mediante `IPv4` o `IPv6`.

[2] Cuando se crea un clúster, se elige entre un clúster `IPv4` y un clúster `IPv6` en la configuración IP family (`ipFamily`) del clúster, y esto no se puede cambiar. En su lugar, es necesario elegir una configuración diferente al momento de crear otro clúster y migrar las cargas de trabajo.

Requisitos de la subred para los nodos

Puede implementar nodos y recursos de Kubernetes en las mismas subredes que especifique al crear el clúster. Sin embargo, esto no es necesario. Esto se debe a que también puede implementar nodos y recursos de Kubernetes en subredes que no especificó al crear el clúster. Si implementa nodos en subredes diferentes, Amazon EKS no crea interfaces de red de clúster en esas subredes. Cualquier subred en la que implemente nodos y recursos de Kubernetes debe cumplir los siguientes requisitos:

- Las subredes deben tener suficientes direcciones IP disponibles para implementar todos sus nodos y recursos de Kubernetes.
- Si quiere que Kubernetes asigne direcciones `IPv6` a Pods y servicios, entonces debe tener un bloque de CIDR `IPv6` y un bloque de CIDR `IPv4` asociado a su subred. Para obtener más información, consulte Asociar un bloque de CIDR `IPv6` a su subred en la Guía del usuario de Amazon VPC. Las tablas de enrutamiento asociadas a las subredes deben incluir rutas a direcciones `IPv4` y `IPv6`. Para obtener más información, consulte Rutas en la Guía del usuario de Amazon VPC. A los pods solo se les asigna una dirección `IPv6`. Sin embargo, a las interfaces de red que Amazon EKS crea para el clúster y los nodos se les asigna una dirección `IPv4` y `IPv6`.
- Si necesita acceso entrante desde Internet a sus Pods, asegúrese de tener al menos una subred pública con suficientes direcciones IP disponibles para implementar equilibradores de carga e ingresar a ellos. Puede implementar un equilibrador de carga en una subred pública. Los equilibradores de carga pueden equilibrar la carga de los Pods en subredes privadas o públicas. Recomendamos implementar los nodos en subredes privadas, si es posible.
- Si planea implementar nodos en una subred pública, la subred debe asignar automáticamente direcciones `IPv4` públicas o direcciones `IPv6`. Si implementa nodos en una subred privada que tiene un bloque de CIDR `IPv6` asociado, la subred privada también debe asignar automáticamente direcciones `IPv6`. Si utilizó una Plantilla de AWS CloudFormation Amazon EKS para implementar la VPC después del 26 de marzo de 2020, esta configuración está habilitada. Si ha utilizado las plantillas para implementar la VPC antes de esta fecha o utiliza su propia VPC, debe habilitar esta configuración manualmente. Para obtener más información, consulte Modificar el atributo de

direcciones IPv4 públicas para su subred y Modificar laIPv6atributo de direcciones de su subred en la Guía de usuario de Amazon VPC.

- Si la subred en la que implementa un nodo es una subred privada y su tabla de enrutamiento no incluye una ruta a un dispositivo de traducción de direcciones de red (NAT) (IPv4) o una puerta de enlace solo de salida (IPv6), agregue puntos de conexión de la VPC mediante AWS PrivateLink a su VPC. Se necesitan puntos de conexión de VPC para todos los Servicios de AWS con los que sus nodos y Pods necesitan comunicarse. Algunos ejemplos incluyen Amazon ECR, equilibradores de carga elástica, Amazon CloudWatch, AWS Security Token Service y Amazon Simple Storage Service (Amazon S3). El punto de conexión debe incluir la subred en la que se encuentran los nodos. No todos los Servicios de AWS admiten los puntos de conexión de VPC. Para obtener más información, consulte ¿Qué es AWS PrivateLink¿ y Servicios de AWS que se integran con AWS PrivateLink. Para obtener una lista de más requisitos de Amazon EKS, consulte Requisitos del clúster privado.
- Si desea implementar equilibradores de carga en una subred, la subred debe tener la siguiente etiqueta:
  - Subredes privadas

Clave	Valor
kubernetes.io/role/internal-elb	1

  - Subredes públicas

Clave	Valor
kubernetes.io/role/elb	1

Cuando se crea un clúster de Kubernetes que es una versión 1.18 y anterior, Amazon EKS agrega la siguiente etiqueta a todas las subredes especificadas.

Clave	Valor
kubernetes.io/cluster/ *my-cluster*	shared

Cuando crea un nuevo clúster de Kubernetes ahora, Amazon EKS no agrega la etiqueta a sus subredes. Si la etiqueta estaba en subredes usadas por un clúster que anteriormente tenía una versión anterior a la `1.19`, la etiqueta no se eliminó automáticamente de las subredes cuando el clúster se actualizó a una versión más reciente. La versión `2.1.1` o anterior del AWS Load Balancer Controller requiere esta etiqueta. Si está usando una versión más reciente del controlador de equilibrador de carga, puede eliminar la etiqueta sin interrumpir sus servicios.

Si ha implementado una VPC mediante `eksctl` o con alguna de las plantillas de VPC de AWS CloudFormation de Amazon EKS, aplica lo siguiente:

- A partir del 26 de marzo de 2020: las subredes públicas asignan de manera automática direcciones `IPv4` públicas a los nodos nuevos implementados en subredes públicas.
- Antes del 26 de marzo de 2020: las subredes públicas no asignan de forma automática las direcciones `IPv4` públicas a los nodos nuevos implementados en subredes públicas.

Este cambio afecta a los nuevos grupos de nodos implementados en subredes públicas de las siguientes formas:

- Grupos de nodos administrados: si el grupo de nodos se implementa en una subred pública a partir del 22 de abril de 2020, la subred pública debe tener habilitada la asignación automática de direcciones IP públicas. Para obtener más información, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred.
- Grupos de nodos autoadministrados Linux, Windows o Arm: si el grupo de nodos se implementa en una subred pública a partir del 26 de marzo de 2020, la asignación automática de direcciones IP públicas deben estar habilitadas para la subred pública. De lo contrario, los nodos deben iniciarse con una dirección IP pública en su lugar. Para obtener más información, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred o Asignación de una dirección `IPv4` pública durante el lanzamiento de la instancia.

Requisitos y consideraciones de la subred compartida

Puede usar Uso compartido de VPC para compartir subredes con otras cuentas de AWS dentro de la misma AWS Organizations. Puede crear clústeres de Amazon EKS en subredes compartidas, teniendo en cuenta las siguientes consideraciones:

- El propietario de la subred de VPC debe compartir una subred con una cuenta participante antes de que esa cuenta pueda crear un clúster de Amazon EKS en ella.

- No puede lanzar recursos mediante el grupo de seguridad predeterminado de la VPC porque pertenece al propietario. Además, los participantes no pueden lanzar recursos mediante grupos de seguridad que sean propiedad de otros participantes o del propietario.
- En una subred compartida, el participante y el propietario controlan por separado los grupos de seguridad de cada cuenta respectiva. El propietario de la subred puede ver estos grupos de seguridad creados por los participantes, pero no puede realizar ninguna acción en ellos. Si el propietario de la subred quiere eliminar o modificar estos grupos de seguridad, el participante que ha creado el grupo de seguridad debe realizar la acción.
- Si un participante crea un clúster, se deben tener en cuenta las siguientes consideraciones:
  - El rol de IAM de clúster y los roles de IAM de nodo deben crearse en esa cuenta. Para obtener más información, consulte Rol de IAM del clúster de Amazon EKS y Rol de IAM de nodo de Amazon EKS.
  - Todos los nodos debe crearlos el mismo participante, incluidos los grupos de nodos administrados.
- El propietario de la VPC compartida no puede ver, actualizar ni eliminar un clúster que un participante cree en la subred compartida. Esto se suma a los recursos de VPC a los que cada cuenta tiene un acceso diferente. Para obtener más información, consulte Responsabilidades y permisos de los propietarios y los participantes en la Guía del usuario de Amazon VPC.
- Si usa la característica de redes personalizadas del Amazon VPC CNI plugin for Kubernetes, debe utilizar las asignaciones de ID de zona de disponibilidad que figuran en la cuenta del propietario para crear cada `ENIConfig`. Para obtener más información, consulte Redes personalizadas para los pods.

Para obtener más información sobre el uso compartido de la subred de VPC, consulte Compartir su VPC con otras cuentas en la Guía del usuario de Amazon VPC.

Creación de una VPC para su clúster de Amazon EKS

Puede usar Amazon Virtual Private Cloud (Amazon VPC) para lanzar recursos de AWS en una red virtual que haya definido. Esta red virtual es prácticamente idéntica a una red tradicional que podría operar en su propio centro de datos. Sin embargo, incluye los beneficios que supone utilizar la infraestructura escalable de Amazon Web Services. Le recomendamos que conozca a fondo el servicio Amazon VPC antes de implementar clústeres de Amazon EKS en producción. Para obtener más información, consulte la Guía del usuario de Amazon VPC.

Un clúster de Amazon EKS, nodos y recursos de Kubernetes se implementan en una VPC. Si desea utilizar una VPC existente con Amazon EKS, dicha VPC debe cumplir los requisitos que se describen en Requisitos y consideraciones de Amazon EKS VPC y subred. En este tema se describe cómo crear una VPC que cumpla los requisitos de Amazon EKS mediante una plantilla AWS CloudFormation proporcionada por Amazon EKS. Una vez que haya implementado una plantilla, podrá ver los recursos creados por la plantilla para saber exactamente qué recursos creó y la configuración de esos recursos.

Requisito previo

Para crear una VPC para Amazon EKS, debe tener los permisos de IAM necesarios para crear recursos de Amazon VPC. Estos recursos son VPC, subredes, grupos de seguridad, tablas de enrutamiento y rutas y puertas de enlace de Internet y NAT. Para obtener más información, consulte Crear una VPC con una política de ejemplo de subred pública en la Guía del usuario de Amazon VPC y en la lista completa de Acciones, recursos y claves de condición de Amazon EC2 en la Referencia de autorizaciones de servicio.

Puede crear una VPC con subredes públicas y privadas, solo subredes públicas o solo subredes privadas.

Public and private subnets

Esta VPC tiene dos subredespúblicas y dos privadas. La tabla de enrutamiento asociada de una subred pública tiene una ruta a una puerta de enlace de internet. Sin embargo, la tabla de enrutamiento de una subred privada no tiene ninguna ruta a ninguna puerta de enlace de Internet. Una subred pública y una subred privada se implementan en la misma zona de disponibilidad. Las otras subredes públicas y privadas se implementan en una segunda zona de disponibilidad en la misma Región de AWS. Recomendamos esta opción para la mayoría de las implementaciones.

Con esta opción, puede implementar los nodos en subredes privadas. Esta opción permite a Kubernetes implementar los equilibradores de carga en las subredes públicas que pueden equilibrar la carga de tráfico a Pods que se ejecuta en los nodos en las subredes privadas. Las direcciones `IPv4` públicas se asignan de forma automática a nodos implementados en subredes públicas, pero las direcciones `IPv4` públicas no se asignan a nodos implementados en subredes privadas.

También puede asignar direcciones `IPv6` a nodos en subredes públicas y privadas. Los nodos de las subredes privadas pueden comunicarse con el clúster y otros Servicios de AWS. Los Pods

pueden comunicarse por Internet a través de una puerta de enlace NAT mediante direcciones IPv4 o una puerta de enlace de Internet de solo salida con direcciones IPv6 que se implementa en cada zona de disponibilidad. Se implementa un grupo de seguridad que dispone de reglas que deniegan todo el tráfico entrante de fuentes distintas del clúster o los nodos, pero permite todo el tráfico saliente. Las subredes están etiquetadas para que Kubernetes pueda implementar equilibradores de carga en ellas.

Crear la VPC

1. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
2. En la barra de navegación, seleccione una Región de AWS compatible con Amazon EKS.
3. Elija Create stack (Crear pila), With new resources (standard) (Con nuevos recursos [estándar]).
4. En Prerequisite - Prepare template (Requisito previo - Preparar plantilla), asegúrese de que esté seleccionada la opción Template is ready (La plantilla está lista) y, a continuación, en Specify template (Especificar plantilla), seleccione Amazon S3 URL (URL de Amazon S3).
5. Puede crear una VPC que admita únicamente IPv4 o una VPC que admita IPv4 y IPv6. Pegue una de las siguientes URL en el área de texto de URL de Amazon S3 y elija Siguiente:

   - IPv4

     ```
     https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml
     ```

   - IPv4 y IPv6

     ```
     https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-ipv6-vpc-public-private-subnets.yaml
     ```

6. En la página Especificar detalles de pila, modifique los parámetros y, a continuación, elija Siguiente.

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation. Por ejemplo, puede usar el nombre de la plantilla que usó en el paso anterior. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.

- VpcBlock: elija un rango de CIDR IPv4 para su VPC. A cada nodo, Pod y equilibrador de carga que implemente se le asigna una dirección IPv4 de este bloque. Los valores predeterminados de IPv4 proporcionan suficientes direcciones IP para la mayoría de las implementaciones, pero, si no lo hacen, puede cambiarlos. Para obtener más información, consulte Tamaño de la subred y VPC en la Guía del usuario de Amazon VPC. También puede agregar bloques de CIDR adicionales a la VPC una vez creada. Si crea una VPC IPv6, los rangos de CIDR IPv6 se asignan de forma automática desde el espacio de direcciones de unidifusión global de Amazon.
- PublicSubnet01Block: especifique un bloque de CIDR IPv4 para la subred pública 1. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo. Si crea una VPC IPv6, este bloque se especifica en su nombre dentro de la plantilla.
- PublicSubnet02Block: especifique un bloque de CIDR IPv4 para la subred pública 2. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo. Si crea una VPC IPv6, este bloque se especifica en su nombre dentro de la plantilla.
- PrivateSubnet01Block: especifique un bloque de CIDR IPv4 para la subred privada 1. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo. Si crea una VPC IPv6, este bloque se especifica en su nombre dentro de la plantilla.
- PrivateSubnet02Block: especifique un bloque de CIDR IPv4 para la subred privada 2. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo. Si crea una VPC IPv6, este bloque se especifica en su nombre dentro de la plantilla.

7. (Opcional) En la página Configurar las opciones de la pila, etiquete los recursos de la pila y, luego, elija Siguiente.
8. En la página Review (Revisar), elija Create stack (Crear pila).
9. Una vez creada la pila, selecciónela en la consola y elija Outputs (Salidas).
10. Registre el VpcId de la VPC que ha creado. Necesita esto cuando al crear su clúster y nodos.
11. Registre los SubnetIds de las subredes que se crearon y si las creó como subredes públicas o privadas. Necesita al menos dos de ellos al crear el clúster y los nodos.
12. Si has creado un VPC IPv4, omita este paso. Si ha creado una VPC IPv6, debe habilitar la opción de asignación automática de dirección IPv6 para las subredes públicas que creó

la plantilla. Esta configuración ya está habilitada para las subredes privadas. Complete los siguientes pasos para habilitar la configuración:

a. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.

b. En el panel de navegación izquierdo, elija Subnets (Subredes).

c. Seleccione una de las subredes públicas (***stack-name***/SubnetPublic01 o ***stack-name***/SubnetPublic02, que contienen la palabra public [pública]), elija Actions (Acciones) y, luego, Edit subnet settings (Editar la configuración de la subred).

d. Marque la casilla de verificación Habilitar asignación automática de direcciones **IPv6** y, luego, elija Guardas.

e. Complete los pasos anteriores de nuevo para la otra subred pública.

Only public subnets

Esta VPC tiene tres subredes públicas que se implementan en diferentes zonas en una Región de AWS. A todos los nodos se les asignan de forma automática direcciones `IPv4` públicas y pueden enviar y recibir tráfico de Internet a través de una puerta de enlace de Internet. Se implementa un grupo de seguridad que deniega todo el tráfico entrante y permite todo el tráfico saliente. Las subredes están etiquetadas para que Kubernetes pueda implementar equilibradores de carga en ellas.

Crear la VPC

1. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
2. En la barra de navegación, seleccione una Región de AWS compatible con Amazon EKS.
3. Elija Create stack (Crear pila), With new resources (standard) (Con nuevos recursos [estándar]).
4. En Prepare template (Preparar la plantilla), asegúrese de que se seleccione Template is ready (La plantilla está lista) y, a continuación, en Template source (Origen de la plantilla), seleccione Amazon S3 URL (URL de Amazon S3).
5. Pegue la siguiente URL en el área de texto de URL de Amazon S3 y elija Siguiente:

```
https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-sample.yaml
```

6. En la página Especificar detalles, rellene los parámetros y, luego, elija Siguiente.

- Nombre de pila: elija un nombre para la pila de AWS CloudFormation. Por ejemplo, puede llamarla ***amazon-eks-vpc-sample***. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
- VpcBlock: elija un bloque de CIDR para la VPC. A cada nodo, Pod y equilibrador de carga que implemente se le asigna una dirección `IPv4` de este bloque. Los valores predeterminados de `IPv4` proporcionan suficientes direcciones IP para la mayoría de las implementaciones, pero, si no lo hacen, puede cambiarlos. Para obtener más información, consulte Tamaño de la subred y VPC en la Guía del usuario de Amazon VPC. También puede agregar bloques de CIDR adicionales a la VPC una vez creada.
- Subnet01Block: especifique un bloque de CIDR para la subred 1. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo.
- Subnet02Block: especifique un bloque de CIDR para la subred 2. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo.
- Subnet03Block: especifique un bloque de CIDR para la subred 3. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo.

7. (Opcional) En la página Options (Opciones), marque los recursos de la pila. Elija Siguiente.
8. En la página Review (Revisar), elija Create (Crear).
9. Una vez creada la pila, selecciónela en la consola y elija Outputs (Salidas).
10. Registre el VpcId de la VPC que ha creado. Necesita esto cuando al crear su clúster y nodos.
11. Anote el valor de SubnetIds de las subredes que se han creado. Necesita al menos dos de ellos al crear el clúster y los nodos.
12. (Opcional) Cualquier clúster que implemente en esta VPC puede asignar direcciones `IPv4` privadas a sus Pods y services. Si desea implementar clústeres en esta VPC para asignar direcciones `IPv6` privadas a sus Pods y services, debe actualizar la VPC, la subred, las tablas de enrutamiento y los grupos de seguridad. Para obtener más información, consulte Migrar VPC existentes de `IPv4` a `IPv6` en la Guía del usuario de Amazon VPC. Amazon EKS requiere que sus subredes tengan la opción de direcciones `Auto-assign IPv6` habilitada. De forma predeterminada, está deshabilitada.

Only private subnets

Esta VPC tiene tres subredes privadas que se implementan en diferentes zonas de disponibilidad de la Región de AWS. Los recursos que se implementan en las subredes no pueden acceder a Internet ni Internet acceder a los recursos de las subredes. La plantilla crea Puntos de conexión de VPC con AWS PrivateLink para varios Servicios de AWS a los que normalmente deben acceder los nodos. Si los nodos necesitan acceso a Internet saliente, puede agregar una puerta de enlace NATpública en la zona de disponibilidad de cada subred después de crear la VPC. Se crea un grupo de seguridad que niega todo el tráfico entrante, excepto los recursos desplegados en las subredes. Un grupo de seguridad también permite todo el tráfico saliente. Las subredes están etiquetadas para que Kubernetes pueda implementar equilibradores de carga internos en ellas. Si va a crear una VPC con esta configuración, consulte Requisitos del clúster privado para obtener requisitos y consideraciones adicionales.

Crear la VPC

1. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
2. En la barra de navegación, seleccione una Región de AWS compatible con Amazon EKS.
3. Elija Create stack (Crear pila), With new resources (standard) (Con nuevos recursos [estándar]).
4. En Prepare template (Preparar la plantilla), asegúrese de que se seleccione Template is ready (La plantilla está lista) y, a continuación, en Template source (Origen de la plantilla), seleccione Amazon S3 URL (URL de Amazon S3).
5. Pegue la siguiente URL en el área de texto de URL de Amazon S3 y elija Siguiente:

   ```
   https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-fully-private-vpc.yaml
   ```

6. En la página Especificar detalles, rellene los parámetros y, luego, elija Siguiente.

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation. Por ejemplo, puede llamarla ***amazon-eks-fully-private-vpc***. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
   - VpcBlock: elija un bloque de CIDR para la VPC. A cada nodo, Pod y equilibrador de carga que implemente se le asigna una dirección `IPv4` de este bloque. Los valores predeterminados de `IPv4` proporcionan suficientes direcciones IP para la mayoría de las

implementaciones, pero, si no lo hacen, puede cambiarlos. Para obtener más información, consulte Tamaño de la subred y VPC en la Guía del usuario de Amazon VPC. También puede agregar bloques de CIDR adicionales a la VPC una vez creada.

- PrivateSubnet01Block: especifique un bloque de CIDR para la subred 1. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo.
- PrivateSubnet02Block: especifique un bloque de CIDR para la subred 2. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo.
- PrivateSubnet03Block: especifique un bloque de CIDR para la subred 3. El valor predeterminado proporciona suficientes direcciones IP para la mayoría de las implementaciones, pero si no lo hace, puede cambiarlo.

7. (Opcional) En la página Options (Opciones), marque los recursos de la pila. Elija Siguiente.
8. En la página Review (Revisar), elija Create (Crear).
9. Una vez creada la pila, selecciónela en la consola y elija Outputs (Salidas).
10. Registre el VpcId de la VPC que ha creado. Necesita esto cuando al crear su clúster y nodos.
11. Anote el valor de SubnetIds de las subredes que se han creado. Necesita al menos dos de ellos al crear el clúster y los nodos.
12. (Opcional) Cualquier clúster que implemente en esta VPC puede asignar direcciones `IPv4` privadas a sus Pods y services. Si desea implementar clústeres en esta VPC para asignar direcciones `IPv6` privadas a sus Pods y services, debe actualizar la VPC, la subred, las tablas de enrutamiento y los grupos de seguridad. Para obtener más información, consulte Migrar VPC existentes de `IPv4` a `IPv6` en la Guía del usuario de Amazon VPC. Amazon EKS requiere que sus subredes tengan la opción de direcciones `Auto-assign IPv6` habilitada (está desactivada de forma predeterminada).

Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS

En este tema se describen los requisitos del grupo de seguridad de un clúster de Amazon EKS.

Al crear un clúster, Amazon EKS crea un grupo de seguridad denominado `eks-cluster-sg-`*`my-cluster-uniqueID`*. Este grupo de seguridad tiene las siguientes reglas de forma predeterminada:

Tipo de regla	Protocolo	Puertos	Origen	Destino
Entrada	Todos	Todos	Auto	
Salida	Todos	Todos		0.0.0.0/0 (IPv4) o ::/0 (IPv6)

> **Important**
>
> Si su clúster no necesita la regla de salida, puede eliminarla. Si la elimina, debe seguir teniendo las reglas mínimas enumeradas en Restricción del tráfico del clúster. Si elimina la regla de entrada, Amazon EKS la vuelve a crear cada vez que se actualice el clúster.

Amazon EKS agrega las siguientes etiquetas al grupo de seguridad. Si elimina las etiquetas, Amazon EKS las vuelve a agregar al grupo de seguridad cada vez que se actualice el clúster.

Clave	Valor
`kubernetes.io/cluster/ my-cluster`	`owned`
`aws:eks:cluster-name`	*my-cluster*
`Name`	`eks-cluster-sg-` *my-cluster* *-uniqueid*

Amazon EKS asocia automáticamente este grupo de seguridad a los siguientes recursos que también crea:

- De 2 a 4 interfaces de red elásticas (denominadas para el resto de este documento como interfaz de red) que se crean al crear el clúster.
- Interfaces de redes de los nodos de cualquier grupo de nodos administrado que cree.

Las reglas predeterminadas permiten que todo el tráfico fluya libremente entre el clúster y los nodos, y permite que todo el tráfico saliente llegue a cualquier destino. Al crear un clúster, puede especificar (opcionalmente) sus propios grupos de seguridad. Si lo hace, Amazon EKS también asocia los

grupos de seguridad especificados a las interfaces de red que crea para su clúster. Sin embargo, no los asocia a ningún grupo de nodos que cree.

Puede determinar el ID del grupo de seguridad de clúster en la AWS Management Console bajo la sección Networking (Redes) del clúster. O, puede hacerlo ejecutando el siguiente comando AWS CLI.

```
aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.clusterSecurityGroupId
```

Restringir el tráfico del clúster

Si debe limitar los puertos abiertos entre el clúster y los nodos, puede eliminar la regla de salida predeterminada y agregar las siguientes reglas mínimas requeridas para el clúster. Si elimina la regla de entrada predeterminada, Amazon EKS la vuelve a crear cada vez que se actualice el clúster.

Tipo de regla	Protocolo	Puerto	Destino
Salida	TCP	443	Grupo de seguridad de clúster
Salida	TCP	10250	Grupo de seguridad de clúster
Saliente (DNS)	TCP y UDP	53	Grupo de seguridad de clúster

También debe agregar reglas para el siguiente tráfico:

- Cualquier protocolo que los puertos que espera que sus nodos usen para la comunicación entre nodos
- Acceso de salida de Internet para que los nodos puedan acceder a las API de Amazon EKS a fin de realizar la introspección del clúster y el registro de nodos en el momento del lanzamiento. Si los nodos no tienen acceso a Internet, revise Requisitos del clúster privado para obtener consideraciones adicionales.

- Acceso a los nodos para extraer imágenes de contenedor de Amazon ECR u otras API de registros de contenedor del que necesiten extraer imágenes, como DockerHub. Para obtener más información, consulte Rangos de direcciones IP de AWS en la Referencia general de AWS.
- Acceso de nodo a Amazon S3.
- Se requieren reglas separadas para las direcciones `IPv4` y `IPv6`.

Si está considerando limitar las reglas, le recomendamos que pruebe detenidamente todos los Pods antes de aplicar las reglas modificadas a un clúster de producción.

Si implementó originalmente un clúster con Kubernetes `1.14` y una versión de plataforma de `eks.3` o anterior, tenga en cuenta lo siguiente:

- Puede que también tenga grupos de seguridad de nodo y plano de control. Cuando se crearon estos grupos, incluyeron las reglas restringidas enumeradas en la tabla anterior. Estos grupos de seguridad ya no son necesarios y se pueden quitar. Sin embargo, debe asegurarse de que el grupo de seguridad del clúster contiene las reglas que contienen esos grupos.
- Si ha implementado el clúster utilizando la API directamente o ha utilizado una herramienta como la AWS CLI o AWS CloudFormation para crear el clúster y no especificó un grupo de seguridad al crear el clúster, el grupo de seguridad predeterminado para la VPC se aplicó a las interfaces de red del clúster que creó Amazon EKS.

Complementos de red de Amazon EKS

Varios complementos de red están disponibles para el clúster de Amazon EKS.

Complementos incorporados

> **Note**
>
> Si crea clústeres de cualquier forma, excepto mediante la consola, cada clúster incluye las versiones autogestionadas de los complementos integrados. Las versiones autogestionadas no se pueden administrar desde AWS Management Console, AWS Command Line Interface o SDK. Usted administra la configuración y las actualizaciones de los complementos autogestionados.

Recomendamos agregar el tipo de complemento de Amazon EKS al clúster en lugar de utilizar el tipo de complemento autoadministrado. Si crea clústeres en la consola, está instalada el tipo de Amazon EKS de estos complementos.

Amazon VPC CNI plugin for Kubernetes

Este CNI crea interfaces de red elástica y las adjunta a los nodos de Amazon EC2. El complemento también asigna una dirección `IPv4` o `IPv6` privada de la VPC a cada Pod y servicio. De forma predeterminada, este complemento se instala en el clúster. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.

CoreDNS

CoreDNS es un servidor de DNS flexible y extensible que puede servir como el DNS del clúster de Kubernetes. CoreDNS proporciona resolución de nombres para todos los Pods del clúster. De forma predeterminada, este complemento se instala en el clúster. Para obtener más información, consulte Trabajar con el complemento CoreDNS de Amazon EKS.

`kube-proxy`

Este complemento mantiene las reglas de red en los nodos de Amazon EC2 y permite la comunicación de red con los Pods. De forma predeterminada, este complemento se instala en el clúster. Para obtener más información, consulte Trabajar con el complemento `kube-proxy` Kubernetes.

Complementos de red de AWS opcionales

AWS Load Balancer Controller

Al implementar objetos de servicio de Kubernetes del tipo `loadbalancer`, el controlador crea equilibradores de carga de red de AWS. Al crear objetos de entrada de Kubernetes, el controlador crea equilibradores de carga de aplicaciones de AWS. Recomendamos usar este controlador para aprovisionar equilibradores de carga de red, en lugar de usar el controlador Cloud Provider antiguo integrado en Kubernetes. Para obtener más información, consulte la documentación de AWS Load Balancer Controller.

Controlador API Gateway de AWS

Este controlador le permite conectar servicios en varios clústeres de Kubernetes utilizando la API de puerta de enlace de Kubernetes. El controlador conecta los servicios de Kubernetes que se ejecutan en instancias, contenedores y funciones sin servidor de Amazon EC2 utilizando el servicio de Amazon VPC Lattice. Para obtener más información, consulte la documentación del controlador AWS de la API de puerta de enlace.

Para obtener más información sobre los complementos, consulte Complementos de Amazon EKS.

Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS

El complemento Amazon VPC CNI plugin for Kubernetes se implementa en cada nodo de Amazon EC2 del clúster de Amazon EKS. El complemento crea interfaces de red elástica y las adjunta a los nodos de Amazon EC2. El complemento también asigna una dirección `IPv4` o `IPv6` privada de la VPC a cada Pod y servicio.

Se implementa una versión del complemento con cada nodo de Fargate del clúster, pero no se actualiza en los nodos de Fargate. Hay otros complementos de CNI compatibles disponibles para su uso en los clústeres de Amazon EKS, pero este es el único complemento de CNI compatible con Amazon EKS.

En la siguiente tabla se muestra la versión más reciente disponible del tipo de complemento de Amazon EKS para cada versión de Kubernetes.

Versión de Kubernetes	**1.30**	**1.29**	**1.28**	**1.27**	**1.26**	**1.25**	**1.24**	**1.23**
Tipo de versión CNI de VPC de Amazon EKS	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild	v1.18.2-eksbuild.1

> **Important**
>
> Si administra este complemento, es posible que las versiones de la tabla no sean las mismas que las versiones autoadministradas disponibles. Para obtener más información acerca de la actualización de complementos autoadministrados, consulte Actualizar el complemento autoadministrado.

> **Important**
>
> Para actualizar a VPC CNI v1.12.0 o superior, primero debe actualizar a VPC CNI v1.7.0. Le recomendamos que actualice una versión secundaria a la vez.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.
- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- Un rol de IAM con la política de IAM AmazonEKS_CNI_Policy (si el clúster utiliza la familia `IPv4`) o una política de IPv6 (si el clúster utiliza la familia `IPv6`) adjuntas. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).
- Si utiliza la versión `1.7.0` o posterior del Amazon VPC CNI plugin for Kubernetes y utiliza políticas de seguridad del Pod personalizadas, consulte Eliminar la política de seguridad predeterminada del Pod de Amazon EKSPolítica de seguridad del pod.
- > **Important**
  >
  > Las versiones `v1.16.0` a `v1.16.1` de Amazon VPC CNI plugin for Kubernetes eliminaron la compatibilidad con versiones `1.23` y anteriores de Kubernetes. La versión `v1.16.2` de VPC CNI restaura la compatibilidad con las versiones `1.23` y anteriores de Kubernetes y las especificaciones `v0.4.0` de CNI.

Las versiones `v1.16.0` a `v1.16.1` de Amazon VPC CNI plugin for Kubernetes implementan la versión `v1.0.0` de la especificación CNI. La especificación `v1.0.0` de CNI es compatibles con los clústeres de EKS que ejecutan las versiones `v1.24` o posteriores de Kubernetes. Las versiones `v1.16.0` a `v1.16.1` de VPC CNI y la especificación `v1.0.0` de CNI no son compatibles en las versiones `v1.23` o anteriores de Kubernetes. Para obtener más información sobre la especificación `v1.0.0` de CNI, consulte Especificación de Container Network Interface (CNI) en

Consideraciones

- Las versiones se especifican como `major-version.minor-version.patch-version-eksbuild.build-number`.
- Comprobar la compatibilidad de versiones para cada característica

  Algunas características de cada versión de Amazon VPC CNI plugin for Kubernetes requieren determinadas versiones de Kubernetes. Cuando se utilizan distintas funciones de Amazon EKS, si se requiere una versión específica del complemento, se indica en la documentación de características. A menos que tenga un motivo específico para ejecutar una versión anterior, le recomendamos elegir la versión más reciente.

Creación del complemento de Amazon EKS

Cree el tipo de Amazon EKS del complemento.

1. Consulte qué versión del complemento está instalada en el clúster.

   ```
   kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
    | cut -d : -f 3
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v1.16.4-eksbuild.2
   ```

2. Consulte qué tipo del complemento está instalado en el clúster. Según la herramienta con la que haya creado el clúster, es posible que actualmente no tenga instalado el tipo de complemento Amazon EKS en el clúster. Reemplace *my-cluster* por el nombre de su clúster.

```
$ aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 addon.addonVersion --output text
```

Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster y no es necesario que complete los pasos restantes del procedimiento. Si se devuelve un error, no tiene el tipo de complemento de Amazon EKS instalado en el clúster. Complete los pasos restantes de este procedimiento para instalarlo.

3. Guarde la configuración del complemento instalado actualmente.

```
kubectl get daemonset aws-node -n kube-system -o yaml > aws-k8s-cni-old.yaml
```

4. Cree el complemento mediante el AWS CLI. Si desea utilizar AWS Management Console o `eksctl` para crear el complemento, consulteCreación de un complemento y especifique`vpc-cni` para el nombre del complemento. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado.

   - Reemplace *my-cluster* por el nombre del clúster.
   - Reemplace *v1.18.2-eksbuild.1* por la versión más reciente que aparece en la tabla de versiones más recientes de la versión de su clúster.
   - Reemplace *111122223333* por el ID de su cuenta y *AmazonEKSVPCCNIRole* por el nombre del rol de IAM existente que creó. Para especificar un rol, es necesario disponer de un proveedor de IAM OpenID Connect (OIDC) para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.

```
aws eks create-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.18.2-eksbuild.1 \
    --service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
```

Si ha aplicado una configuración personalizada al complemento actual que entra en conflicto con la configuración predeterminada del complemento de Amazon EKS, es posible que se produzca un error en la creación. Si se produce un error en la creación, recibe un error que puede serle de utilidad para resolver el problema. Como alternativa, puede añadir **`--resolve-conflicts OVERWRITE`** al comando anterior. Esto permite que el complemento sobrescriba

cualquier configuración personalizada existente. Una vez que haya creado el complemento, puede actualizarlo con la configuración personalizada.

5. Confirme que la versión más reciente del complemento de la Kubernetes versión de su clúster se haya agregado al clúster. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 addon.addonVersion --output text
```

Es posible que la creación del complemento tarde varios segundos en completarse.

Un ejemplo de salida sería el siguiente.

```
v1.18.2-eksbuild.1
```

6. Si ha realizado ajustes personalizados en el complemento original, antes de crear el complemento de Amazon EKS, utilice la configuración que guardó en el paso anterior para actualizar el complemento de Amazon EKS con su configuración personalizada.
7. (Opcional) Instale el `cni-metrics-helper` en su clúster. Extrae información de la interfaz de red elástica y la dirección IP, agrega métricas en todo el clúster y publica las métricas en Amazon CloudWatch. Para obtener más información, consulte cni-metrics-helper en GitHub.

Actualizar el complemento de Amazon EKS

Actualice el tipo de Amazon EKS del complemento. Si no ha agregado el tipo Amazon EKS del complemento al clúster, agréguelo o consulte Actualizar el complemento autoadministrado, en lugar de completar este procedimiento.

1. Consulte qué versión del complemento está instalada en el clúster. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 "addon.addonVersion" --output text
```

Un ejemplo de salida sería el siguiente.

```
v1.16.4-eksbuild.2
```

Si la versión devuelta es la misma que la versión Kubernetes del clúster en la tabla de versiones más recientes, significa que ya tiene la última versión instalada en el clúster y no necesita completar el resto de este procedimiento. Si recibe un error, en lugar de un número de versión en la salida, significa que no tiene el tipo de versión de Amazon EKS en el clúster. Debe crear el complemento antes de poder actualizarlo mediante este procedimiento.

2. Guarde la configuración del complemento instalado actualmente.

```
kubectl get daemonset aws-node -n kube-system -o yaml > aws-k8s-cni-old.yaml
```

3. Actualice el complemento con la AWS CLI. Si desea utilizar AWS Management Console o `eksctl` para actualizar el complemento, consulte Actualización de un complemento. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado.

   - Reemplace *`my-cluster`* por el nombre del clúster.
   - Reemplace *`v1.18.2-eksbuild.1`* por la versión más reciente que aparece en la tabla de versiones más recientes de la versión de su clúster.
   - Reemplace *`111122223333`* por el ID de su cuenta y *`AmazonEKSVPCCNIRole`* por el nombre del rol de IAM existente que creó. Para especificar un rol, es necesario disponer de un proveedor de IAM OpenID Connect (OIDC) para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.
   - La opción *`CONSERVAR`* de **`--resolve-conflicts`** conserva los valores de configuración existentes del complemento. Si ha establecido valores personalizados para la configuración del complemento y no utiliza esta opción, Amazon EKS sobrescribe los valores con los valores predeterminados. Si utiliza esta opción, le recomendamos que pruebe cualquier cambio de campo y valor en un clúster que no sea de producción antes de actualizar el complemento del clúster de producción. Si cambia este valor a `OVERWRITE`, todas las configuraciones cambiarán a los valores predeterminados de Amazon EKS. Si ha establecido valores personalizados para cualquier configuración, es posible que se sobrescriban con los valores predeterminados de Amazon EKS. Si cambia este valor a `none`, Amazon EKS no cambia el valor de ninguna configuración, pero la actualización podría fallar. Si se produce un error en la actualización, recibe un mensaje de error que lo ayuda a resolver el conflicto.
   - Si no va a actualizar un ajuste de configuración, elimine **`--configuration-values '{`***`"env":{"AWS_VPC_K8S_CNI_EXTERNALSNAT":"true"}`***`}'`** del comando. Si va a actualizar una configuración, sustituya *`“env”:`*

{“AWS_VPC_K8S_CNI_EXTERNALSNAT”:“true”} por la configuración que desee establecer. En este ejemplo, la variable de entorno AWS_VPC_K8S_CNI_EXTERNALSNAT se establece en `true`. El valor que especifique debe ser válido para el esquema de configuración. Si no conoce el esquema de configuración, ejecute **aws eks describe-addon-configuration --addon-name vpc-cni --addon-version *v1.18.2-eksbuild.1*** y reemplace *v1.18.2-eksbuild.1* por el número de versión del complemento cuya configuración desea ver. El esquema se devuelve en la salida. Si ya tiene alguna configuración personalizada, quiere eliminarla toda y volver a establecer los valores de todos los ajustes en los valores predeterminados de Amazon EKS, elimine *“env”:{“AWS_VPC_K8S_CNI_EXTERNALSNAT”:“true”}* del comando para que quede vacío **{}**. Para obtener una explicación de cada configuración, consulte las Variables de configuración de CNI en GitHub.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.18.2-eksbuild.1 \
    --service-account-role-arn arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
 \
    --resolve-conflicts PRESERVE --configuration-values '{"env":
{"AWS_VPC_K8S_CNI_EXTERNALSNAT":"true"}}'
```

La actualización puede tardar varios segundos en completarse.

4. Confirme que la versión del complemento se ha actualizado. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni
```

La actualización puede tardar varios segundos en completarse.

Un ejemplo de salida sería el siguiente.

```
{
    "addon": {
        "addonName": "vpc-cni",
        "clusterName": "my-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.18.2-eksbuild.1",
        "health": {
            "issues": []
        },
```

```
        "addonArn": "arn:aws:eks:region:111122223333:addon/my-cluster/vpc-
cni/74c33d2f-b4dc-8718-56e7-9fdfa65d14a9",
        "createdAt": "2023-04-12T18:25:19.319000+00:00",
        "modifiedAt": "2023-04-12T18:40:28.683000+00:00",
        "serviceAccountRoleArn":
 "arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole",
        "tags": {},
        "configurationValues": "{\"env\":{\"AWS_VPC_K8S_CNI_EXTERNALSNAT\":\"true
\"}}"
    }
}
```

Actualizar el complemento autoadministrado

> **Important**
>
> Recomendamos agregar el tipo de complemento de Amazon EKS al clúster en lugar de utilizar el tipo de complemento autoadministrado. Si no está familiarizado con la diferencia entre los tipos, consulte the section called “Complementos de Amazon EKS”. Para obtener más información acerca de cómo agregar un complemento de Amazon EKS al clúster, consulte the section called “Creación de un complemento”. Si no puede usar el complemento de Amazon EKS, le recomendamos que envíe una pregunta sobre los motivos por los que no puede hacerlo al repositorio de GitHub de la hoja de ruta de contenedores.

1. Confirme que no tiene instalado en el clúster el tipo Amazon EKS del complemento autoadministrado. Reemplace *my-cluster* por el nombre de su clúster.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
    addon.addonVersion --output text
   ```

   Si se devuelve un error, no tiene el tipo de complemento de Amazon EKS instalado en el clúster. Para autoadministrar el complemento, complete los pasos restantes de este procedimiento para actualizar el complemento. Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster. Para actualizarlo, siga el procedimiento que aparece en Actualización de un complemento, en lugar de este procedimiento. Si no está familiarizado con las diferencias entre los tipos de complementos, consulte Complementos de Amazon EKS.
2. Consulte qué versión de la imagen del contenedor está instalada actualmente en el clúster.

```
kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
 | cut -d : -f 3
```

Un ejemplo de salida sería el siguiente.

```
v1.16.4-eksbuild.2
```

Es posible que su resultado no incluya el número de compilación.

3. Haga una copia de seguridad de la configuración actual para poder aplicar la misma configuración una vez que haya actualizado la versión.

```
kubectl get daemonset aws-node -n kube-system -o yaml > aws-k8s-cni-old.yaml
```

4. Consulte releases en GitHub para revisar las versiones disponibles y familiarizarse con los cambios efectuados en la versión a la que desea actualizar. Tenga en cuenta que le recomendamos que actualice a la misma versión `major.minor.patch` que aparece en la tabla de versiones más recientes disponibles, incluso si hay versiones posteriores disponibles en GitHub. Las versiones de compilación que aparecen en la tabla no se especifican en las versiones autoadministradas que aparecen en GitHub. Actualice su versión al completar las tareas de una de las siguientes opciones:

   - Si no tiene ninguna configuración personalizada para el complemento, ejecute el comando que aparece debajo del encabezado `To apply this release:` en GitHub para la versión a la que desea actualizar.
   - Si tiene una configuración personalizada, descargue el archivo de manifiesto con el siguiente comando. Cambie *https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/v1.18.2/config/master/aws-k8s-cni.yaml* por la URL de la versión de GitHub a la que está actualizando.

     ```
     curl -O https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/v1.18.2/config/master/aws-k8s-cni.yaml
     ```

     Si es necesario, modifique el archivo de manifiesto con la configuración personalizada de la copia de seguridad que hizo en un paso anterior y, a continuación, aplique el archivo modificado al clúster. Si los nodos no tienen acceso a los repositorios privados de Amazon ECR de Amazon EKS de donde se extraen las imágenes (consulte las líneas que

comienzan con `image:` en el manifiesto), tendrá que descargar las imágenes, copiarlas en su propio repositorio y modificar el manifiesto para que extraiga las imágenes de su repositorio. Para obtener más información, consulte Copiar una imagen de contenedor de un repositorio en otro repositorio.

```
kubectl apply -f aws-k8s-cni.yaml
```

5. Confirme que la nueva versión ya esté instalada en el clúster.

```
kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
 | cut -d : -f 3
```

Un ejemplo de salida sería el siguiente.

```
v1.18.2
```

6. (Opcional) Instale el `cni-metrics-helper` en su clúster. Extrae información de la interfaz de red elástica y la dirección IP, agrega métricas en todo el clúster y publica las métricas en Amazon CloudWatch. Para obtener más información, consulte cni-metrics-helper en GitHub.

Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA)

El Amazon VPC CNI plugin for Kubernetes es el complemento de red para redes de Pod en clústeres de Amazon EKS. El complemento se encarga de asignar direcciones IP de la VPC a los nodos de Kubernetes y establecer la configuración de red necesaria para los Pods de cada nodo. El complemento:

- Requiere permisos de AWS Identity and Access Management (IAM). Si el clúster usa la familia IPv4, los permisos se especifican en la política administrada `AmazonEKS_CNI_Policy` AWS. Si el clúster utiliza la familia IPv6, entonces se deben agregar los permisos a una Política de IAM que crea.Puede asociar la política al rol de IAM del nodo de Amazon EKS o a un rol de IAM independiente. Le recomendamos que lo adjunte a un rol independiente, tal y como se detalla en este tema.
- Crea y está configurado para utilizar una cuenta de servicio de Kubernetes denominada `aws-node`, cuando se implementa. La cuenta de servicio está vinculada a un Kubernetes `clusterrole` denominado `aws-node`, al que se le asignan los permisos de Kubernetes necesarios.

> **Note**
>
> Los Pods correspondientes al Amazon VPC CNI plugin for Kubernetes tienen acceso a los permisos asignados al rol de IAM del nodo de Amazon EKS, a menos que se bloquee el acceso a IMDS. Para obtener más información, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.
- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.

Paso 1: Crear el rol de IAM Amazon VPC CNI plugin for Kubernetes

Cómo crear el rol de IAM

1. Determine la familia de IP del clúster.

   ```
   aws eks describe-cluster --name my-cluster | grep ipFamily
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   "ipFamily": "ipv4"
   ```

   La salida puede devolver `ipv6` en cambio.

2. Cree el rol de IAM. Puede utilizar `eksctl` o `kubectl` y el AWS CLI para crear el rol de IAM.

   eksctl

   Cree un rol de IAM y adjunte la política de IAM al rol con el comando que coincida con la familia IP del clúster. Este comando crea e implementa una pila de AWS CloudFormation que crea un rol de IAM, adjunta la política que especifica para el rol y anota la cuenta de servicio `aws-node` Kubernetes existente con el ARN del rol de IAM que se crea.

- IPv4

  Reemplace *my-cluster* por su propio valor.

  ```
  eksctl create iamserviceaccount \
      --name aws-node \
      --namespace kube-system \
      --cluster my-cluster \
      --role-name AmazonEKSVPCCNIRole \
      --attach-policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
      --override-existing-serviceaccounts \
      --approve
  ```

- IPv6

  Reemplace *my-cluster* por su propio valor. Reemplace *111122223333* por el ID de su cuenta y reemplace *AmazonEKS_CNI_IPv6_Policy*por el nombre de su política IPv6. Si no dispone de una política IPv6, consulte Cree una política de IAM para clústeres que utilizan la familia IPv6 para crear uno. Para utilizar IPv6 con su clúster, debe cumplir varios requisitos. Para obtener más información, consulte Direcciones IPv6 de clústeres, Pods y services.

  ```
  eksctl create iamserviceaccount \
      --name aws-node \
      --namespace kube-system \
      --cluster my-cluster \
      --role-name AmazonEKSVPCCNIRole \
      --attach-policy-arn
   arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy \
      --override-existing-serviceaccounts \
      --approve
  ```

kubectl and the AWS CLI

1. Vea la URL del proveedor de OIDC de su clúster.

   ```
   aws eks describe-cluster --name my-cluster --query
    "cluster.identity.oidc.issuer" --output text
   ```

Un ejemplo de salida sería el siguiente.

```
https://oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE
```

Si no se devuelve ninguna salida, debe crear un proveedor de OIDC de IAM para el clúster.

2. Copie el siguiente contenido en un archivo denominado *vpc-cni-trust-policy.json*. Reemplace *111122223333* por su ID de cuenta y *EXAMPLED539D4633E53DE1B71EXAMPLE* por la salida que obtuvo en el paso anterior. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:aws-node"
                }
            }
        }
    ]
}
```

3. Cree el rol. Puede reemplazar *AmazonEKSVPCCNIRole* con el nombre que usted elija.

```
aws iam create-role \
  --role-name AmazonEKSVPCCNIRole \
  --assume-role-policy-document file://"vpc-cni-trust-policy.json"
```

4. Adjunte la política de IAM necesaria al rol de IAM Ejecute el comando que coincida con la familia IP del clúster.

   - IPv4

     ```
     aws iam attach-role-policy \
       --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
       --role-name AmazonEKSVPCCNIRole
     ```

   - IPv6

     Reemplace *111122223333* por el ID de su cuenta y *AmazonEKS_CNI_IPv6_Policy* por el nombre de su política de IPv6. Si no dispone de una política IPv6, consulte Cree una política de IAM para clústeres que utilizan la familia IPv6 para crear uno. Para utilizar IPv6 con su clúster, debe cumplir varios requisitos. Para obtener más información, consulte Direcciones IPv6 de clústeres, Pods y services.

     ```
     aws iam attach-role-policy \
       --policy-arn arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy \
       --role-name AmazonEKSVPCCNIRole
     ```

5. Ejecute el siguiente comando para anotar la cuenta de servicio del aws-node con el ARN del rol de IAM que creó anteriormente. Sustituya *example values* con valores propios.

   ```
   kubectl annotate serviceaccount \
       -n kube-system aws-node \
       eks.amazonaws.com/role-
   arn=arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
   ```

3. (Opcional) Configure el tipo de punto de conexión AWS Security Token Service utilizado por su cuenta de servicio de Kubernetes. Para obtener más información, consulte Configure el punto de conexión AWS Security Token Service de una cuenta de servicio.

Paso 2: Nueva implementación de Amazon VPC CNI plugin for KubernetesPods

1. Elimine y vuelva a crear todos los Pods existentes asociados a la cuenta de servicio para aplicar las variables de entorno de credenciales. La anotación no se aplica a Pods que se están ejecutando actualmente sin la anotación. El siguiente comando elimina los aws-node DaemonSet Pods existentes y los implementa con la anotación de cuenta de servicio.

```
kubectl delete Pods -n kube-system -l k8s-app=aws-node
```

2. Confirme que todos los Pods se reiniciaron.

```
kubectl get pods -n kube-system -l k8s-app=aws-node
```

3. Describa uno de los Pods y verifique que existen las variables de entorno `AWS_WEB_IDENTITY_TOKEN_FILE` y `AWS_ROLE_ARN`. Reemplace *cpjw7* por el nombre de uno de los Pods que obtuvo en la salida del paso anterior.

```
kubectl describe pod -n kube-system aws-node-cpjw7 | grep 'AWS_ROLE_ARN:\|
AWS_WEB_IDENTITY_TOKEN_FILE:'
```

Un ejemplo de salida sería el siguiente.

```
AWS_ROLE_ARN:                 arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
      AWS_WEB_IDENTITY_TOKEN_FILE:  /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
      AWS_ROLE_ARN:
 arn:aws:iam::111122223333:role/AmazonEKSVPCCNIRole
      AWS_WEB_IDENTITY_TOKEN_FILE:           /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
```

Se devuelven dos conjuntos de resultados duplicados porque el Pod contiene dos contenedores. Ambos contenedores tienen los mismos valores.

Si su Pod está utilizando el punto de conexión de la Región de AWS, entonces la siguiente línea también se devuelve en la salida anterior.

```
AWS_STS_REGIONAL_ENDPOINTS=regional
```

Paso 3: Eliminar la política de CNI del rol de IAM del nodo

Si actualmente su rol de IAM del nodo de Amazon EKS tiene asociada la política (`IPv4`) de IAM `AmazonEKS_CNI_Policy` o una política `IPv6` , y ha creado un rol de IAM independiente, le ha asociado la política y lo ha asignado a la cuenta de servicio de Kubernetes `aws-node`, se recomienda eliminar la política del rol del nodo con el comando de la AWS CLI que coincida con la familia de IP de su clúster. Reemplace *AmazonEKSNodeRole* por el nombre de rol de su nodo.

- IPv4

```
aws iam detach-role-policy --role-name AmazonEKSNodeRole --policy-arn
 arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
```

- IPv6

  Reemplace *111122223333* por el ID de su cuenta y *AmazonEKS_CNI_IPv6_Policy* por el nombre de su política de IPv6.

```
aws iam detach-role-policy --role-name AmazonEKSNodeRole --policy-arn
 arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy
```

Cree una política de IAM para clústeres que utilizan la familia **IPv6**

Si creó un clúster que utiliza la familia IPv6 y el clúster tiene configurada la versión 1.10.1 o posterior del complemento Amazon VPC CNI plugin for Kubernetes, debe crear una política de IAM que pueda asignar a un rol de IAM en un paso posterior. Si tiene un clúster existente que no configuró con la familia IPv6 cuando lo creó, deberá crear un clúster nuevo para poder utilizar IPv6. Para obtener más información acerca del uso de IPv6 con su clúster, consulte Direcciones IPv6 de clústeres, Pods y services.

1. Copie el siguiente texto y guárdelo en un archivo llamado *vpc-cni-ipv6-policy*.json.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssignIpv6Addresses",
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
```

2. Cree la política de IAM.

```
aws iam create-policy --policy-name AmazonEKS_CNI_IPv6_Policy --policy-document
 file://vpc-cni-ipv6-policy.json
```

Elección de casos de uso de redes de Pod

Amazon VPC CNI plugin for Kubernetes proporciona redes para Pods. La siguiente tabla lo ayudará a comprender qué casos de uso de redes se pueden utilizar juntos, así como las capacidades y la configuración del Amazon VPC CNI plugin for Kubernetes que puede utilizar con diferentes tipos de nodos de Amazon EKS. Toda la información de la tabla se aplica únicamente a los nodos de Linux `IPv4`.

Tipo de nodo de Amazon EKS	Amazon EC2			Fargate
Caso de uso	Direcciones IP individuales asignadas a la interfaz de red	Prefijos de IP asignados a la interfaz de red	Grupos de seguridad de Pods	
Redes personalizadas para los pods: asignar direcciones IP desde una subred distinta a la subred del nodo	Sí	Sí	Sí	Sí (subredes controladas a través del perfil de Fargate)

Tipo de nodo de Amazon EKS	Amazon EC2			Fargate
Caso de uso	Direcciones IP individuales asignadas a la interfaz de red	Prefijos de IP asignados a la interfaz de red	Grupos de seguridad de Pods	
SNAT para Pods	Sí (el valor predeterminado es `false`)	Sí (el valor predeterminado es `false`)	Sí (solo `true`)	Sí (solo `true`)
Capacidades				
Ámbito de grupo de seguridad	Nodo	Nodo	Pod (si configuró `POD_SECURITY_GROUP_ENFORCING_MODE = standard` y `AWS_VPC_K8S_CNI_EXTERNALSNAT =false`, el tráfico destinado a puntos finales fuera de la VPC utiliza los grupos de seguridad del nodo, no los grupos de seguridad de Pod's)	Pod

Tipo de nodo de Amazon EKS	Amazon EC2			Fargate
Caso de uso	Direcciones IP individuales asignadas a la interfaz de red	Prefijos de IP asignados a la interfaz de red	Grupos de seguridad de Pods	
Tipos de subred de Amazon VPC	Pública y privada	Pública y privada	Solo privada	Solo privada
Política de red (CNI de VPC)	Compatible	Compatible	Compatible Solo con la versión `1.14.0` o posterior del complemento CNI de Amazon VPC	No compatible
Densidad de pods por nodo	Medio	Alta	Baja	Uno
Hora de lanzamiento del pod	Mejor	Óptima	Buena	Moderado
Configuración del complemento CNI de Amazon VPC (para obtener más información acerca de cada opción, consulte amazon-vpc-cni-k8s en GitHub)				
`WARM_ENI_TARGET`	Sí	No aplicable	No aplicable	No aplicable
`WARM_IP_TARGET`	Sí	Sí	No aplicable	No aplicable
`MINIMUM_IP_TARGET`	Sí	Sí	No aplicable	No aplicable

Tipo de nodo de Amazon EKS	Amazon EC2			Fargate
Caso de uso	Direcciones IP individuales asignadas a la interfaz de red	Prefijos de IP asignados a la interfaz de red	Grupos de seguridad de Pods	
`WARM_PREFIX_TARGET`	No aplicable	Sí	No aplicable	No aplicable

Note

- No puede utilizar `IPv6` con las redes personalizadas.
- Las direcciones `IPv6` no se traducen, por lo que no se aplica SNAT.
- El flujo de tráfico hacia y desde los Pods con grupos de seguridad asociados no está sujeto a la aplicación de políticas de red de Calico y solo se limita a la aplicación de grupos de seguridad de Amazon VPC.
- Si utiliza la aplicación de políticas de red de Calico, le recomendamos que configure la variable de entorno `ANNOTATE_POD_IP` en `true` para evitar un problema conocido con Kubernetes. Para utilizar esta característica, debe añadir permisos de `patch` para los pods a `aws-node` ClusterRole. Tenga en cuenta que añadir permisos de parche a `aws-node` DaemonSet aumenta el alcance de seguridad del plugin. Para obtener más información, consulte ANNOTATE_POD_IP en el repositorio de CNI de la VPC en GitHub.
- Los prefijos IP y las direcciones IP están asociados a las interfaces de red elásticas estándar de Amazon EC2. A los pods que requieren grupos de seguridad específicos se les asigna la dirección IP principal de una interfaz de red de ramificación. Puede mezclar Pods que obtengan direcciones IP o direcciones IP de prefijos IP con Pods que obtengan interfaces de red de ramificación en el mismo nodo.

Nodos del Windows

Cada nodo solo admite una interfaz de red. Puede utilizar direcciones `IPv4` y prefijos `IPv4` secundarios. De forma predeterminada, la cantidad de direcciones `IPv4` disponibles

en el nodo es igual a la cantidad de direcciones IPv4 secundarias que puede asignar a cada interfaz de red elástica, menos una. Sin embargo, puede aumentar las direcciones IPv4 disponibles y la densidad Pod en el nodo habilitando los prefijos IP. Para obtener más información, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.

Las políticas de red de Calico son compatibles con Windows. No puede usar grupos de seguridad Pods ni redes personalizadas en Windows.

Direcciones **IPv6** de clústeres, Pods y services

De forma predeterminada, Kubernetes asigna direcciones IPv4 a sus Pods y services. En lugar de asignar direcciones IPv4 a sus Pods y services, puede configurar el clúster para que les asigne direcciones IPv6. Amazon EKS no admite Pods o services de doble pila, aunque Kubernetes sí lo hace en versión 1.23 y posterior. Como resultado, no puede asignar ambos tipos de direcciones, IPv4 e IPv6, a sus Pods y services.

Selecciona qué familia IP desea utilizar para su clúster cuando lo cree. Una vez creado el clúster, no se puede cambiar la familia.

Consideraciones para utilizar la familia **IPv6** para el clúster

- Debe crear un clúster nuevo y especificar que desea utilizar la familia IPv6 para ese clúster. No puede habilitar la familia IPv6 para un clúster que haya actualizado desde una versión anterior. Para obtener instrucciones sobre cómo crear un clúster nuevo, consulte Creación de un clúster de Amazon EKS.
- La versión del complemento Amazon VPC CNI que implementa en el clúster debe ser la versión 1.10.1 o posterior. Esta versión o posterior se implementa de forma predeterminada. Una vez que implemente el complemento, no podrá revertir la versión del complemento Amazon VPC CNI a una versión inferior a 1.10.1 sin eliminar primero todos los nodos de todos los grupos de nodos del clúster.
- No se admiten Pods y services de Windows.
- Si utiliza nodos de Amazon EC2, debe configurar el complemento Amazon VPC CNI con delegación de prefijos IP e IPv6. Si elige la familia IPv6 cuando cree el clúster, la versión 1.10.1 del complemento tendrá esta configuración como predeterminada. Este es el caso de un complemento autoadministrado o de Amazon EKS. Para obtener más información acerca de la delegación de prefijos IP, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2.

- Cuando crea un clúster, la VPC y las subredes que especifique deben tener un bloque de CIDR IPv6 asignado a la VPC y las subredes que usted especifica. También deben tener un bloque de CIDR IPv4 asignado. Esto se debe a que, incluso si solo desea utilizar IPv6, una VPC necesita un bloque de CIDR IPv4 para funcionar. Para obtener más información, consulte Asociar un bloque de CIDR IPv6 a su VPC en la Guía del usuario de Amazon VPC.
- Cuando crea el clúster y los nodos, debe especificar subredes que estén configuradas para asignar de forma automática direcciones IPv6. De lo contrario, no podrá implementar el clúster ni los nodos. Esta configuración está desactivada de forma predeterminada. Para obtener más información, consulte Modificar el atributo de direccionamiento IPv6 de su subred en la Guía del usuario de Amazon VPC.
- Las tablas de enrutamiento que se asignen a las subredes deben tener rutas para direcciones IPv6. Para obtener más información, consulte Migración a IPv6 en la Guía del usuario de Amazon VPC.
- Los grupos de seguridad deben permitir el uso de direcciones IPv6. Para obtener más información, consulte Migración a IPv6 en la Guía del usuario de Amazon VPC.
- Solo puede utilizar IPv6 con nodos de Amazon EC2 o de Fargate basados en AWS Nitro.
- No puede utilizar IPv6 con Grupos de seguridad de Pods con nodos de Amazon EC2. Sin embargo, puede utilizarse con nodos de Fargate. Si necesita grupos de seguridad independientes para Pods individuales, continúe utilizando la familia IPv4 con nodos de Amazon EC2 o utilice nodos de Fargate en su lugar.
- Si anteriormente utilizó redes personalizadas para ayudar a aliviar el agotamiento de las direcciones IP, puede utilizar IPv6 en su lugar. No puede utilizar redes personalizadas con IPv6. Si utiliza redes personalizadas para el aislamiento de red, es posible que deba continuar utilizando redes personalizadas y la familia IPv4 para sus clústeres.
- No se puede utilizar IPv6 con AWS Outposts.
- A Pods y services solo se les asigna una dirección IPv6. No se les asigna una dirección IPv4. Debido a que los Pods pueden comunicarse con puntos de conexión IPv4 a través de NAT en la propia instancia, no se necesita DNS64 ni NAT64. Si el tráfico necesita una dirección IP pública, el tráfico es la dirección de red de origen convertida en una IP pública.
- La dirección IPv6 de origen de un Pod no es la dirección de red de origen convertida en la dirección IPv6 del nodo cuando se comunica fuera de la VPC. Se enruta mediante una puerta de enlace de Internet o una puerta de enlace de Internet de solo salida.
- A todos los nodos se les asignan una dirección IPv4 e IPv6.
- No es compatible con la Controlador CSI de Amazon FSx para Lustre.

- Puede utilizar la versión `2.3.1` o posterior del Controlador del equilibrador de carga de AWS para equilibrar la carga de tráfico de la aplicación o la red a los Pods `IPv6` en modo IP, pero no en el modo de instancia. Para obtener más información, consulte ¿Qué es el AWS Load Balancer Controller?.
- Debe adjuntar una política de IAM de `IPv6` al IAM de su rol de IAM de CNI o nodo de IAM. Entre los dos, le recomendamos que lo adjunte a un rol de IAM de CNI. Para obtener más información, consulte Cree una política de IAM para clústeres que utilizan la familia `IPv6` y Paso 1: Crear el rol de IAM Amazon VPC CNI plugin for Kubernetes.
- Cada Pod de Fargate recibe una dirección `IPv6` del CIDR que se especifica para la subred en la que se implementa. La unidad de hardware subyacente que ejecuta Pods de Fargate obtiene una dirección `IPv4` e `IPv6` única de los CIDR asignados a la subred en la que se implementa la unidad de hardware.
- Le recomendamos que realice una evaluación exhaustiva de las aplicaciones, los complementos de Amazon EKS y los servicios de AWS con los que se integra antes de implementar clústeres `IPv6`. Esto es para garantizar que todo funcione según lo esperado con `IPv6`.
- El uso del punto de conexión `IPv6` del servicio de metadatos de instancia de Amazon EC2 no es compatible con Amazon EKS.
- Al crear un grupo de nodos autoadministrado en un clúster que utiliza la familia `IPv6`, los datos del usuario deben incluir los siguientes `BootstrapArguments` para el archivo `bootstrap.sh` que se ejecuta al iniciar el nodo. Reemplace *`your-cidr`* con el rango `IPv6` CIDR de la VPC del clúster.

  ```
  --ip-family ipv6 --service-ipv6-cidr your-cidr
  ```

  Si no sabe cuál es el rango `IPv6` `CIDR` de su clúster, puede verlo con el siguiente comando (requiere el AWS CLI versión `2.4.9` o posterior).

  ```
  aws eks describe-cluster --name my-cluster --query
   cluster.kubernetesNetworkConfig.serviceIpv6Cidr --output text
  ```

Implemente un clúster **`IPv6`** y nodos de Amazon Linux administrados

En este tutorial, implementa una nube de Amazon VPC `IPv6`, un clúster de Amazon EKS con la familia `IPv6` y un grupo de nodos administrado con nodos de Amazon Linux de Amazon EC2. No se pueden implementar nodos de Windows de Amazon EC2 en un clúster `IPv6`. También puede

implementar nodos de Fargate en su clúster, aunque esas instrucciones no se proporcionarán en este tema por motivos de simplicidad.

Antes de crear un clúster para su uso en producción, recomendamos que conozca toda la configuración e implemente un clúster con la configuración que satisfaga sus requisitos. Para obtener más información, consulte Creación de un clúster de Amazon EKS, Grupos de nodos administrados y las consideraciones sobre este tema. Solo puede habilitar algunos ajustes de la configuración cuando cree su clúster.

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesitará para crear y administrar un clúster de Amazon EKS.

- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- La entidad principal de seguridad de IAM que está utilizando debe contar con permisos para trabajar con los roles de IAM de Amazon EKS y los roles vinculados al servicio, AWS CloudFormation, además de una VPC y recursos relacionados. Para obtener más información, consulte Acciones, recursos y claves de condición de Amazon Elastic Kubernetes Service y Uso de roles vinculados a servicios en la Guía del usuario de IAM.

Se proporcionan procedimientos para crear los recursos con `eksctl` o la AWS CLI. También puede implementar los recursos mediante la AWS Management Console, pero esas instrucciones no se proporcionan en este tema por una cuestión de simplicidad.

eksctl

Requisito previo

Versión `eksctl` `0.183.0` o posterior instalada en el equipo. Para instalar o actualizar esta versión, consulte la sección de Instalación en la documentación de `eksctl`.

Para implementar un clúster **IPv6** con **eksctl**

1. Cree el archivo `ipv6-cluster.yaml`. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado:

   - Reemplace *my-cluster* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
   - Reemplace *region-code* por cualquier Región de AWS en la que se admita Amazon EKS. Para ver una lista de Regiones de AWS, consulte Puntos de conexión y cuotas de Amazon EKS en la Guía de referencia general de AWS.
   - El valor de `version` con la versión de su clúster. Para obtener más información, consulte la versión de Kubernetes compatible de Amazon EKS.
   - Reemplace *my-nodegroup* por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.
   - Reemplace *t3.medium* por cualquier tipo de instancia de AWS Nitro System.

```
cat >ipv6-cluster.yaml <<EOF
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: "X.XX"

kubernetesNetworkConfig:
  ipFamily: IPv6

addons:
  - name: vpc-cni
    version: latest
  - name: coredns
```

```
    version: latest
  - name: kube-proxy
    version: latest

iam:
  withOIDC: true

managedNodeGroups:
  - name: my-nodegroup
    instanceType: t3.medium
EOF
```

2. Cree su clúster.

```
eksctl create cluster -f ipv6-cluster.yaml
```

La creación del clúster tarda varios minutos. No continúe hasta que vea la última línea de salida, que se parece a la siguiente salida.

```
[...]
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

3. Confirme que a los Pods predeterminados se les asignan direcciones IPv6.

```
kubectl get pods -n kube-system -o wide
```

Un ejemplo de salida sería el siguiente.

```
NAME                       READY   STATUS    RESTARTS   AGE     IP
                             NODE
 NOMINATED NODE   READINESS GATES
aws-node-rslts             1/1     Running   1          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
aws-node-t74jh             1/1     Running   0          5m32s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
coredns-85d5b4454c-cw7w2   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::                ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
```

```
coredns-85d5b4454c-tx6n8   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::1               ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
kube-proxy-btpbk           1/1     Running   0          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
kube-proxy-jjk2g           1/1     Running   0          5m33s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
```

4. Confirme que a los servicios predeterminados se les asignan direcciones IPv6.

```
kubectl get services -n kube-system -o wide
```

Un ejemplo de salida sería el siguiente.

```
NAME       TYPE        CLUSTER-IP          EXTERNAL-IP   PORT(S)         AGE
 SELECTOR
kube-dns   ClusterIP   fd30:3087:b6c2::a   <none>        53/UDP,53/TCP   57m
 k8s-app=kube-dns
```

5. (Opcional) Implemente una aplicación de muestra o implemente el AWS Load Balancer Controller y una aplicación de muestra para equilibrar la carga del tráfico de la aplicación o la red en los Pods IPv6.
6. Cuando haya terminado con el clúster y los nodos que creó para este tutorial, debería limpiar los recursos que creó con el siguiente comando.

```
eksctl delete cluster my-cluster
```

AWS CLI

Requisito previo

La versión 2.12.3 o posterior, o bien, la versión 1.27.160 o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **aws --version | cut -d / -f2 | cut -d ' ' -f1**. Los administradores de paquetes tales como yum, apt-get o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración

rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell. Si utiliza AWS CloudShell, es posible que deba instalar la versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS CLI, ya que la versión predeterminada de la AWS CLI instalada en AWS CloudShell puede ser una versión anterior.

⚠ Important

- Debe completar todos los pasos de este procedimiento como el mismo usuario. Ejecute el siguiente comando para comprobar el usuario actual:

  ```
  aws sts get-caller-identity
  ```

- Debe completar todos los pasos de este procedimiento en el mismo intérprete de comandos. Varios pasos utilizan variables establecidas en pasos anteriores. Los pasos que utilizan variables no funcionarán de forma adecuada si los valores de las variables se establecen en un intérprete de comandos diferente. Si utiliza AWS CloudShell para completar el siguiente procedimiento, recuerde que, si no interactúa con él a través del teclado o el puntero durante 20 o 30 minutos aproximadamente, se cerrará la sesión del intérprete de comandos. Los procesos en ejecución no cuentan como interacciones.
- Las instrucciones están escritas para el intérprete de comandos Bash y es posible que deban ajustarse para otros intérpretes de comandos.

Para crear el clúster con la AWS CLI

Reemplace todos los *`example values`* en los pasos de este procedimiento por sus propios valores.

1. Ejecute los siguientes comandos para establecer algunas variables utilizadas en pasos posteriores. Reemplace *`region-code`* por la Región de AWS en la que desea implementar sus recursos. El valor puede ser cualquier Región de AWS que sea compatible con Amazon EKS. Para ver una lista de Regiones de AWS, consulte Puntos de conexión y cuotas de Amazon EKS en la Guía de referencia general de AWS. Reemplace *`my-cluster`* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y

no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. Reemplace *my-nodegroup* por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Reemplace *111122223333* por su ID de cuenta.

```
export region_code=region-code
export cluster_name=my-cluster
export nodegroup_name=my-nodegroup
export account_id=111122223333
```

2. Cree una Amazon VPC con subredes privadas y públicas que cumplan con los requisitos de Amazon EKS y IPv6.

   a. Ejecute el siguiente comando para establecer una variable para el nombre de su pila de AWS CloudFormation. Puede reemplazar *my-eks-ipv6-vpc* con cualquier nombre que elija.

   ```
   export vpc_stack_name=my-eks-ipv6-vpc
   ```

   b. Cree una VPC IPv6 a partir de una plantilla de AWS CloudFormation.

   ```
   aws cloudformation create-stack --region $region_code --stack-name
    $vpc_stack_name \
     --template-url https://s3.us-west-2.amazonaws.com/amazon-
   eks/cloudformation/2020-10-29/amazon-eks-ipv6-vpc-public-private-
   subnets.yaml
   ```

   La pila tarda unos minutos en crearse. Ejecute el siguiente comando de la . No continúe con el siguiente paso hasta que la salida del comando sea CREATE_COMPLETE.

   ```
   aws cloudformation describe-stacks --region $region_code --stack-name
    $vpc_stack_name --query Stacks[].StackStatus --output text
   ```

   c. Recupere los ID de las subredes públicas que se crearon.

   ```
   aws cloudformation describe-stacks --region $region_code --stack-name
    $vpc_stack_name \
       --query='Stacks[].Outputs[?OutputKey==`SubnetsPublic`].OutputValue' --
   output text
   ```

Un ejemplo de salida sería el siguiente.

```
subnet-0a1a56c486EXAMPLE,subnet-099e6ca77aEXAMPLE
```

d. Habilite la opción de asignación automática de direcciones IPv6 para las subredes públicas que se crearon.

```
aws ec2 modify-subnet-attribute --region $region_code --
subnet-id subnet-0a1a56c486EXAMPLE --assign-ipv6-address-on-
creation
aws ec2 modify-subnet-attribute --region $region_code --subnet-id
 subnet-099e6ca77aEXAMPLE --assign-ipv6-address-on-creation
```

e. Recupere los nombres de las subredes y los grupos de seguridad creados por la plantilla desde la pila de AWS CloudFormation implementada y almacénelos en variables para utilizarlos en un paso posterior.

```
security_groups=$(aws cloudformation describe-stacks --region $region_code
 --stack-name $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SecurityGroups`].OutputValue' --
output text)

public_subnets=$(aws cloudformation describe-stacks --region $region_code --
stack-name $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SubnetsPublic`].OutputValue' --
output text)

private_subnets=$(aws cloudformation describe-stacks --region $region_code
 --stack-name $vpc_stack_name \
    --query='Stacks[].Outputs[?OutputKey==`SubnetsPrivate`].OutputValue' --
output text)

subnets=${public_subnets},${private_subnets}
```

3. Cree un rol de IAM de clúster y adjúntelo a la política administrada de IAM de Amazon EKS. Los clústeres de Kubernetes administrados por Amazon EKS realizan llamadas a otros servicios de AWS en su nombre para administrar los recursos que utiliza con el servicio.

   a. Ejecute el siguiente comando para crear un archivo `eks-cluster-role-trust-policy.json`.

```
cat >eks-cluster-role-trust-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

b. Ejecute el siguiente comando para establecer una variable para el nombre de su rol. Puede reemplazar *myAmazonEKSClusterRole* con cualquier nombre que elija.

```
export cluster_role_name=myAmazonEKSClusterRole
```

c. Cree el rol.

```
aws iam create-role --role-name $cluster_role_name --assume-role-policy-
document file://"eks-cluster-role-trust-policy.json"
```

d. Recupere el ARN del rol de IAM y almacénelo en una variable para un paso posterior.

```
cluster_iam_role=$(aws iam get-role --role-name $cluster_role_name --
query="Role.Arn" --output text)
```

e. Adjunte la política administrada de IAM por Amazon EKS requerida al rol.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name $cluster_role_name
```

4. Cree su clúster.

```
aws eks create-cluster --region $region_code --name $cluster_name --kubernetes-
version 1.XX \
   --role-arn $cluster_iam_role --resources-vpc-config subnetIds=
$subnets,securityGroupIds=$security_groups \
```

```
--kubernetes-network-config ipFamily=ipv6
```

- > **Note**
  >
  > Es posible que reciba un error que indique que una de las zonas de disponibilidad de la solicitud no tiene capacidad suficiente para crear un clúster de Amazon EKS. Si esto ocurre, el mensaje de error indicará las zonas de disponibilidad que admiten un clúster nuevo. Intente crear el clúster de nuevo con al menos dos subredes ubicadas en las zonas de disponibilidad admitidas para su cuenta. Para obtener más información, consulte Capacidad insuficiente.

  El clúster tarda varios minutos en crearse. Ejecute el siguiente comando. No continúe con el siguiente paso hasta que la salida del comando sea `ACTIVE`.

  ```
  aws eks describe-cluster --region $region_code --name $cluster_name --query
   cluster.status
  ```

5. Cree o actualice un archivo `kubeconfig` para su clúster para que pueda comunicarse con él.

   ```
   aws eks update-kubeconfig --region $region_code --name $cluster_name
   ```

   De forma predeterminada, el archivo de `config` se crea en `~/.kube` o la configuración del clúster nuevo se agrega a un archivo de `config` existente en `~/.kube`.

6. Cree un rol de IAM de nodo.

   a. Ejecute el siguiente comando para crear un archivo `vpc-cni-ipv6-policy.json`.

   ```
   cat >vpc-cni-ipv6-policy <<EOF
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": [
                   "ec2:AssignIpv6Addresses",
                   "ec2:DescribeInstances",
                   "ec2:DescribeTags",
   ```

```
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
EOF
```

b. Cree la política de IAM.

```
aws iam create-policy --policy-name AmazonEKS_CNI_IPv6_Policy --policy-
document file://vpc-cni-ipv6-policy.json
```

c. Ejecute el siguiente comando para crear un archivo `node-role-trust-relationship.json`.

```
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

d. Ejecute el siguiente comando para establecer una variable para el nombre de su rol. Puede reemplazar *AmazonEKSNodeRole* con cualquier nombre que elija.

```
export node_role_name=AmazonEKSNodeRole
```

e. Cree el rol de IAM.

```
aws iam create-role --role-name $node_role_name --assume-role-policy-
document file://"node-role-trust-relationship.json"
```

f. Adjunte la política de IAM al rol de IAM.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::
$account_id:policy/AmazonEKS_CNI_IPv6_Policy \
    --role-name $node_role_name
```

> **Important**
>
> Para simplificar este tutorial, la política se adjunta a este rol de IAM. Sin embargo, en un clúster de producción, recomendamos adjuntar la política a un rol de IAM independiente. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

g. Adjunte al rol de IAM las dos políticas administradas por IAM necesarias.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy \
  --role-name $node_role_name
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEC2ContainerRegistryReadOnly \
  --role-name $node_role_name
```

h. Recupere el ARN del rol de IAM y almacénelo en una variable para un paso posterior.

```
node_iam_role=$(aws iam get-role --role-name $node_role_name --
query="Role.Arn" --output text)
```

7. Cree un grupo de nodos administrados.

a. Vea los ID de las subredes que creó en un paso anterior.

```
echo $subnets
```

Un ejemplo de salida sería el siguiente.

```
subnet-0a1a56c486EXAMPLE,subnet-099e6ca77aEXAMPLE,subnet-
0377963d69EXAMPLE,subnet-0c05f819d5EXAMPLE
```

b. Cree el grupo de nodos. Reemplace *0a1a56c486EXAMPLE*, *099e6ca77aEXAMPLE*, *0377963d69EXAMPLE* y *0c05f819d5EXAMPLE* con los valores devueltos en el resultado del paso anterior. Asegúrese de eliminar las comas entre los ID de subredes de la salida anterior en el siguiente comando. Puede reemplazar *t3.medium* por cualquier tipo de instancia de AWS Nitro System.

```
aws eks create-nodegroup --region $region_code --cluster-name $cluster_name
 --nodegroup-name $nodegroup_name \
    --subnets subnet-0a1a56c486EXAMPLE subnet-099e6ca77aEXAMPLE
 subnet-0377963d69EXAMPLE subnet-0c05f819d5EXAMPLE \
    --instance-types t3.medium --node-role $node_iam_role
```

El grupo de nodos tarda unos minutos en crearse. Ejecute el siguiente comando. No continúe con el paso siguiente hasta que la salida devuelta sea `ACTIVE`.

```
aws eks describe-nodegroup --region $region_code --cluster-name
 $cluster_name --nodegroup-name $nodegroup_name \
    --query nodegroup.status --output text
```

8. Confirme que a los Pods predeterminados se les asignen direcciones `IPv6` en la columna `IP`.

```
kubectl get pods -n kube-system -o wide
```

Un ejemplo de salida sería el siguiente.

```
NAME                       READY   STATUS    RESTARTS   AGE     IP
                             NODE
 NOMINATED NODE   READINESS GATES
aws-node-rslts             1/1     Running   1          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
```

```
aws-node-t74jh             1/1     Running   0          5m32s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
coredns-85d5b4454c-cw7w2   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::                ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
coredns-85d5b4454c-tx6n8   1/1     Running   0          56m
  2600:1f13:b66:8203:34e5::1               ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
kube-proxy-btpbk           1/1     Running   0          5m36s
  2600:1f13:b66:8200:11a5:ade0:c590:6ac8   ip-192-168-34-75.region-
code.compute.internal   <none>           <none>
kube-proxy-jjk2g           1/1     Running   0          5m33s
  2600:1f13:b66:8203:4516:2080:8ced:1ca9   ip-192-168-253-70.region-
code.compute.internal  <none>           <none>
```

9. Confirme que a los servicios predeterminados se les asignen direcciones IPv6 en la columna IP.

   ```
   kubectl get services -n kube-system -o wide
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME       TYPE        CLUSTER-IP          EXTERNAL-IP   PORT(S)         AGE
    SELECTOR
   kube-dns   ClusterIP   fd30:3087:b6c2::a   <none>        53/UDP,53/TCP   57m
    k8s-app=kube-dns
   ```

10. (Opcional) Implemente una aplicación de muestra o implemente el AWS Load Balancer Controller y una aplicación de muestra para equilibrar la carga del tráfico de la aplicación o la red en los Pods IPv6.

11. Cuando haya terminado con el clúster y los nodos que creó para este tutorial, debería limpiar los recursos que creó con los siguientes comandos. Asegúrese de no estar utilizando ninguno de los recursos fuera de este tutorial antes de eliminarlos.

    a. Si completa este paso en un intérprete de comandos diferente al que completó los pasos anteriores, establezca los valores de todas las variables utilizadas en los pasos anteriores y reemplace los *example values* por los valores que especificó cuando completó los pasos anteriores. Si está completando este paso en el mismo intérprete de comandos en el que completó los pasos anteriores, continúe con el siguiente paso.

```
export region_code=region-code
export vpc_stack_name=my-eks-ipv6-vpc
export cluster_name=my-cluster
export nodegroup_name=my-nodegroup
export account_id=111122223333
export node_role_name=AmazonEKSNodeRole
export cluster_role_name=myAmazonEKSClusterRole
```

b. Elimine su grupo de nodos.

```
aws eks delete-nodegroup --region $region_code --cluster-name $cluster_name
 --nodegroup-name $nodegroup_name
```

La eliminación tarda unos minutos. Ejecute el siguiente comando de la . No continúe con el siguiente paso si se devuelve alguna salida.

```
aws eks list-nodegroups --region $region_code --cluster-name $cluster_name
 --query nodegroups --output text
```

c. Eliminar el clúster.

```
aws eks delete-cluster --region $region_code --name $cluster_name
```

El clúster tarda unos minutos en eliminarse. Antes de continuar, asegúrese de que el clúster se elimine con el siguiente comando.

```
aws eks describe-cluster --region $region_code --name $cluster_name
```

No continúe con el siguiente paso hasta que la salida sea similar a la siguiente.

```
An error occurred (ResourceNotFoundException) when calling the
 DescribeCluster operation: No cluster found for name: my-cluster.
```

d. Elimine los resultados de IAM que creó. Reemplace *AmazonEKS_CNI_IPv6_Policy* por el nombre que eligió si eligió un nombre diferente al que utilizó en los pasos anteriores.

```
aws iam detach-role-policy --role-name $cluster_role_name --policy-arn
 arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
```

```
aws iam detach-role-policy --role-name $node_role_name --policy-arn
 arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
aws iam detach-role-policy --role-name $node_role_name --policy-arn
 arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
aws iam detach-role-policy --role-name $node_role_name --policy-arn
 arn:aws:iam::$account_id:policy/AmazonEKS_CNI_IPv6_Policy
aws iam delete-policy --policy-arn arn:aws:iam::
$account_id:policy/AmazonEKS_CNI_IPv6_Policy
aws iam delete-role --role-name $cluster_role_name
aws iam delete-role --role-name $node_role_name
```

e. Elimine la pila de AWS CloudFormation que creó la VPC.

```
aws cloudformation delete-stack --region $region_code --stack-name
 $vpc_stack_name
```

SNAT para Pods

Si ha implementado el clúster mediante la familia `IPv6`, la información de este tema no se aplica a su clúster, porque las direcciones `IPv6` no están traducidas en red. Para obtener más información acerca del uso de `IPv6` con su clúster, consulte Direcciones `IPv6` de clústeres, Pods y services.

De forma predeterminada, a cada Pod en el clúster se le asigna una dirección privada `IPv4` de un bloque de enrutamiento entre dominios sin clase (CIDR) asociado a la VPC en la que el Pod se implementa. Los Pods en la misma VPC se comunican entre sí utilizando estas direcciones IP privadas como puntos de conexión. Cuando un Pod se comunica a cualquier dirección `IPv4` que no se encuentra dentro de un bloque CIDR asociado a la VPC, el complemento CNI de Amazon VPC (tanto para Linux como para Windows) traduce la dirección `IPv4` del Pod's a la dirección `IPv4` privada principal de la interfaz de red elástica principal del nodo en el que el Pod se está ejecutando, de forma predeterminada [*].

> **Note**
>
> En el caso de los nodos de Windows, hay detalles adicionales que se deben tener en cuenta. De forma predeterminada, el complemento CNI de VPC para Windows se define con una configuración de red en la que se excluye el tráfico a un destino dentro de la misma VPC para SNAT. Esto significa que la comunicación interna de la VPC tiene la SNAT desactivada y la dirección IP asignada a un Pod se puede enrutar dentro de la VPC. Sin embargo, el tráfico a un destino fuera de la VPC tiene la IP del Pod de origen vinculada mediante la SNAT

a la dirección IP principal de la ENI de la instancia. Esta configuración predeterminada de Windows garantiza que el pod pueda acceder a redes fuera de la VPC de la misma manera que a la instancia del host.

Debido a este comportamiento:

- Los Pods pueden comunicarse con recursos de Internet solo si el nodo en el que se están ejecutando tiene una dirección IP pública o elástica asignada y se encuentra en una subred pública. Una subred pública es una subred asociada a la tabla de enrutamiento con ruta a la puerta de enlace de internet. Recomendamos implementar nodos en subredes privadas, siempre que sea posible.
- Para versiones del complemento anteriores a la `1.8.0`, los recursos que se encuentran en redes o VPC que están conectadas a la VPC del clúster mediante un emparejamiento de VPC, una VPC de tránsito o AWS Direct Connect no pueden iniciar la comunicación con sus Pods mediante interfaces de redes elásticas secundarias. No obstante, sus Pods pueden iniciar la comunicación con esos recursos y recibir respuestas de ellos.

Si alguna de las siguientes afirmaciones es verdadera en su entorno, cambie la configuración predeterminada con el comando siguiente.

- Tiene recursos en redes o VPC que están conectados a la VPC de clúster mediante Interconexión con VPC, una VPC de tránsito o AWS Direct Connect que deben iniciar la comunicación con su Pods mediante una dirección `IPv4` y la versión del complemento es anterior a la `1.8.0`.
- Sus Pods se encuentran en una subred privada y necesitan comunicación saliente a Internet. La subred tiene una ruta hacia una puerta de enlace NAT.

```
kubectl set env daemonset -n kube-system aws-node AWS_VPC_K8S_CNI_EXTERNALSNAT=true
```

> **Note**
>
> Las variables de configuración de CNI `AWS_VPC_K8S_CNI_EXTERNALSNAT` y `AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS` no se aplican a los nodos de Windows. No se admite la desactivación de SNAT para Windows. En cuanto a la exclusión de una lista de CIDR `IPv4` de SNAT, puede definirla especificando el parámetro `ExcludedSnatCIDRs`

en el script de arranque de Windows. Para obtener más información acerca del uso de este parámetro, consulte Parámetros de configuración del script de arranque.

*Si las especificaciones de un Pod's contienen `hostNetwork=true` (el valor predeterminado es `false`), su dirección IP no se traduce a otra dirección. Este es el caso de la `kube-proxy` y Amazon VPC CNI plugin for Kubernetes Pods que se ejecutan en el clúster, de forma predeterminada. Para estos Pods, la dirección IP es la misma que la dirección IP principal del nodo, por lo que la dirección IP Pod's no está traducida. Para obtener más información acerca de la configuración de `hostNetwork` de Pod's, consulte Núcleo de PodSpec v1 en la referencia de API de Kubernetes.

Configure su clúster para las políticas de red de Kubernetes

De forma predeterminada, no hay restricciones en Kubernetes para las direcciones IP, puertos o conexiones entre cualquier Pods de su clúster o entre sus Pods y recursos en cualquier otra red. Puede usar la política de red de Kubernetes para restringir el tráfico de red que entra y sale de su Pods. Para obtener más información, consulte Políticas de red en la documentación de Kubernetes.

Si tiene una versión `1.13` o anterior del Amazon VPC CNI plugin for Kubernetes en su clúster, tiene que implementar una solución de terceros para aplicar políticas de red de Kubernetes a su clúster. La versión `1.14` o posterior del complemento puede implementar políticas de red, de modo que no es necesario utilizar una solución de terceros. En este tema, aprenderá a configurar el clúster para que utilice la política de red de Kubernetes en su clúster sin utilizar un complemento de terceros.

Las políticas de red del Amazon VPC CNI plugin for Kubernetes se admiten en las siguientes configuraciones.

- Clústeres de Amazon EKS de la versión `1.25` y posterior.
- Versión 1.14 o posterior del Amazon VPC CNI plugin for Kubernetes en su clúster.
- Clúster configurado para las direcciones `IPv4` o `IPv6`.
- Puede utilizar las políticas de red con los grupos de seguridad para Pods. Con las políticas de red, puede controlar todas las comunicaciones dentro del clúster. Con los grupos de seguridad para Pods, puede controlar el acceso a Servicios de AWS desde aplicaciones dentro de un Pod.
- Puede utilizar las políticas de red con las redes personalizadas y la delegación de prefijos.

Consideraciones

- Al aplicar las políticas de red del Amazon VPC CNI plugin for Kubernetes a su clúster con el Amazon VPC CNI plugin for Kubernetes, puede aplicar las políticas únicamente a los nodos de Linux de Amazon EC2. No puede aplicar las políticas a los nodos de Fargate o Windows.
- Si su clúster utiliza actualmente una solución de terceros para la administración de políticas de red de Kubernetes, puede usar esas mismas políticas con el Amazon VPC CNI plugin for Kubernetes. Sin embargo, debe eliminar la solución existente para que no administre las mismas políticas.
- Puede aplicar varias políticas de red al mismo Pod. Cuando se han configurado dos o más políticas que seleccionan el mismo Pod, todas las políticas se aplican al Pod.
- El número máximo de combinaciones únicas de puertos para cada protocolo en cada selector de `ingress:` o `egress:` en una política de red es 24.
- Para cualquiera de sus servicios de Kubernetes, el puerto de servicio debe ser el mismo que el puerto del contenedor. Si utiliza puertos con nombre, utilice también el mismo nombre en la especificación del servicio.
- Aplicación de políticas al inicio del Pod

  El Amazon VPC CNI plugin for Kubernetes configura las políticas de red para los pods en paralelo con el aprovisionamiento de los pods. A menos que se hayan configurado todas las políticas para el nuevo pod, los contenedores del nuevo pod se iniciarán con una política de permisos predeterminada. Esto se denomina modo estándar. Una política de permisos predeterminada significa que se permite todo el tráfico de entrada y salida desde y hacia los nuevos pods.

  Para cambiar esta política de red predeterminada, debe configurar la variable de entorno `NETWORK_POLICY_ENFORCING_MODE` en `strict` en el contenedor `aws-node` del `DaemonSet` de la CNI de la VPC.

  ```
  env:
    - name: NETWORK_POLICY_ENFORCING_MODE
      value: "strict"
  ```

  Con la variable `NETWORK_POLICY_ENFORCING_MODE` configurada en `strict`, los pods que utilizan la CNI de la VPC se inician con una política de denegación predeterminada y, a continuación, se configuran las políticas. Esto se denomina modo estricto. En el modo estricto, debe tener una política de red para cada punto de conexión del clúster al que los pods necesiten acceder. Tenga en cuenta que este requisito se aplica a los pods CoreDNS. La política de denegación predeterminada no está configurada para los pods con redes de Host.

- La característica de política de red crea y requiere una definición de recurso personalizada (CRD) de `PolicyEndpoint` llamada `policyendpoints.networking.k8s.aws`. Amazon EKS administra los objetos `PolicyEndpoint` del recurso personalizado. No debe modificar ni eliminar estos recursos.
- Si ejecuta pods que utilizan las credenciales de IAM del rol de instancia o se conectan al IMDS de EC2, compruebe si hay políticas de red que puedan bloquear el acceso al IMDS de EC2. Puede que tenga que añadir una política de red para permitir el acceso al IMDS de EC2. Para obtener más información, consulte Metadatos de instancia y datos de usuario en la guía del usuario de Amazon EC2.

  Los pods que utilizan los roles de IAM para cuentas de servicio no accedan al IMDS de EC2.
- El Amazon VPC CNI plugin for Kubernetes no aplica políticas de red a las interfaces de red adicionales de cada pod, solo a la interfaz principal de cada pod (`eth0`). Esta característica afecta a las siguientes arquitecturas:
  - Pods `IPv6` con la variable `ENABLE_V4_EGRESS` establecida en `true`. Esta variable habilita la característica de salida `IPv4` para conectar los pods IPv6 a puntos de conexión `IPv4`, como aquellos fuera del clúster. La característica de salida `IPv4` funciona mediante la creación de una interfaz de red adicional con una dirección IPv4 de bucle invertido local.
  - Cuando se utilizan complementos de red encadenados, como Multus. Debido a que estos complementos añaden interfaces de red a cada pod, las políticas de red no se aplican a los complementos de red encadenados.
- La característica de la política de red utiliza el puerto `8162` del nodo para las métricas de manera predeterminada. Además, la característica utilizaba el puerto `8163` para las sondas de estado. Si ejecuta otra aplicación en los nodos o dentro de pods que necesite utilizar estos puertos, la aplicación no se puede ejecutar. En la versión de CNI de VPC `v1.14.1` o posterior, puede cambiar estos puertos en los siguientes lugares:

  AWS Management Console

  1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
  2. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento CNI de Amazon VPC.
  3. Elija la pestaña Complementos.
  4. Seleccione la casilla situada en la parte superior derecha del cuadro y, a continuación, elija Edit (Editar).
  5. En la página Configurar el ***nombre del complemento***:

a. Seleccione la versión `v1.14.0-eksbuild.3` o posterior en la lista desplegable Versión.

b. Seleccione Ajustes de configuración opcionales.

c. Introduzca la clave JSON `"enableNetworkPolicy":` y el valor `"true"` en Valores de configuración. El texto resultante debe ser un objeto JSON válido. Si esta clave y este valor son los únicos datos del cuadro de texto, rodee la clave y el valor entre corchetes `{}`.

   El siguiente ejemplo tiene habilitada la característica de política de red y los registros de política de red habilitados. Estos se envían a Registros de Amazon CloudWatch, y las métricas y sondas de estado se configuran con los números de puertos predeterminados:

```
{
    "enableNetworkPolicy": "true",
    "nodeAgent": {
        "enablePolicyEventLogs": "true",
        "enableCloudWatchLogs": "true",
        "healthProbeBindAddr": "8163",
        "metricsBindAddr": "8162"
    }
}
```

Helm

Si ha instalado el Amazon VPC CNI plugin for Kubernetes mediante `helm`, puede actualizar la configuración para cambiar los puertos.

- Ejecute el siguiente comando para cambiar los puertos. Establezca el número de puerto en el valor para la clave `nodeAgent.metricsBindAddr` o `nodeAgent.healthProbeBindAddr`, respectivamente.

```
helm upgrade --set nodeAgent.metricsBindAddr=8162 --set
 nodeAgent.healthProbeBindAddr=8163 aws-vpc-cni --namespace kube-system eks/
aws-vpc-cni
```

kubectl

1. Abra el `DaemonSet` de `aws-node` en el editor.

```
kubectl edit daemonset -n kube-system aws-node
```

2. Reemplace los números de puertos en los siguientes argumentos de comando de `args:` del contenedor `aws-network-policy-agent` del manifiesto del daemonset de `aws-node` de CNI de VPC.

```
    - args:
            - --metrics-bind-addr=:8162
            - --health-probe-bind-addr=:8163
```

Requisitos previos

- Versión mínima del clúster

Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS. El clúster debe ser la versión `1.25` de Kubernetes o posterior. El clúster debe estar ejecutando una de las versiones de Kubernetes y versiones de la plataforma que se enumeran en la siguiente tabla. Tenga en cuenta que también se admite cualquier versión de Kubernetes y de la plataforma posterior a las enumeradas. Puede comprobar la versión actual de Kubernetes reemplazando *`my-cluster`* en el siguiente comando por el nombre del clúster y luego ejecutando el comando modificado:

```
aws eks describe-cluster
              --name my-cluster --query cluster.version --output
              text
```

Versión de Kubernetes	Versión de la plataforma
`1.27.4`	`eks.5`
`1.26.7`	`eks.6`

Versión de Kubernetes	Versión de la plataforma
1.25.12	eks.7

- Versión mínima de CNI de VPC

  Versión 1.14 o posterior del Amazon VPC CNI plugin for Kubernetes en su clúster. Puede comprobar qué versión utiliza actualmente con el siguiente comando.

  ```
  kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni: |
   cut -d : -f 3
  ```

  Si su versión es anterior a la 1.14, actualice Actualizar el complemento de Amazon EKS a la versión 1.14 o posterior.

- Versión mínima del kernel de Linux

  Sus nodos deben tener la versión del núcleo de Linux 5.10 o posterior. Puede comprobar la versión del núcleo con uname -r. Si utiliza las versiones más recientes de las AMI optimizadas para Amazon EKS de Amazon Linux, las AMI optimizadas para Amazon EKS de Amazon Linux acelerado y las AMI de Bottlerocket, estas ya tienen la versión de núcleo necesaria.

  La versión v20231116 de la AMI de Amazon Linux acelerada y optimizada de Amazon EKS o posterior tiene una versión 5.10 de núcleo.

Para configurar su clúster para que use las políticas de red de Kubernetes

1. Montar el sistema de archivos BPF

   > **Note**
   >
   > Si su clúster es la versión 1.27 o posterior, puede omitir este paso, ya que todas las AMI de Amazon Linux y Bottlerocket optimizadas para Amazon EKS para la versión 1.27 o posterior ya disponen de esta característica.
   > Para todas las demás versiones de clústeres, si actualiza Amazon Linux optimizado para Amazon EKS a la versión v20230703 o posterior, o actualiza la AMI de Bottlerocket a la versión v1.0.2 o posterior, puede omitir este paso.

a. Monte el sistema de archivos Berkeley Packet Filter (BPF) en cada uno de sus nodos.

```
sudo mount -t bpf bpffs /sys/fs/bpf
```

b. A continuación, añada el mismo comando a los datos de usuario en la plantilla de lanzamiento de sus grupos de Amazon EC2 Auto Scaling.

2. Habilitar la política de red en el CNI de VPC

a. Consulte qué tipo del complemento está instalado en el clúster. Según la herramienta con la que haya creado el clúster, es posible que actualmente no tenga instalado el tipo de complemento Amazon EKS en el clúster. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 addon.addonVersion --output text
```

Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster y no es necesario que complete los pasos restantes del procedimiento. Si se devuelve un error, no tiene el tipo de complemento de Amazon EKS instalado en el clúster.

b. • Complemento de Amazon EKS

AWS Management Console

a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

b. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento CNI de Amazon VPC.

c. Elija la pestaña Complementos.

d. Seleccione la casilla situada en la parte superior derecha del cuadro y, a continuación, elija Edit (Editar).

e. En la página Configurar el ***nombre del complemento***:

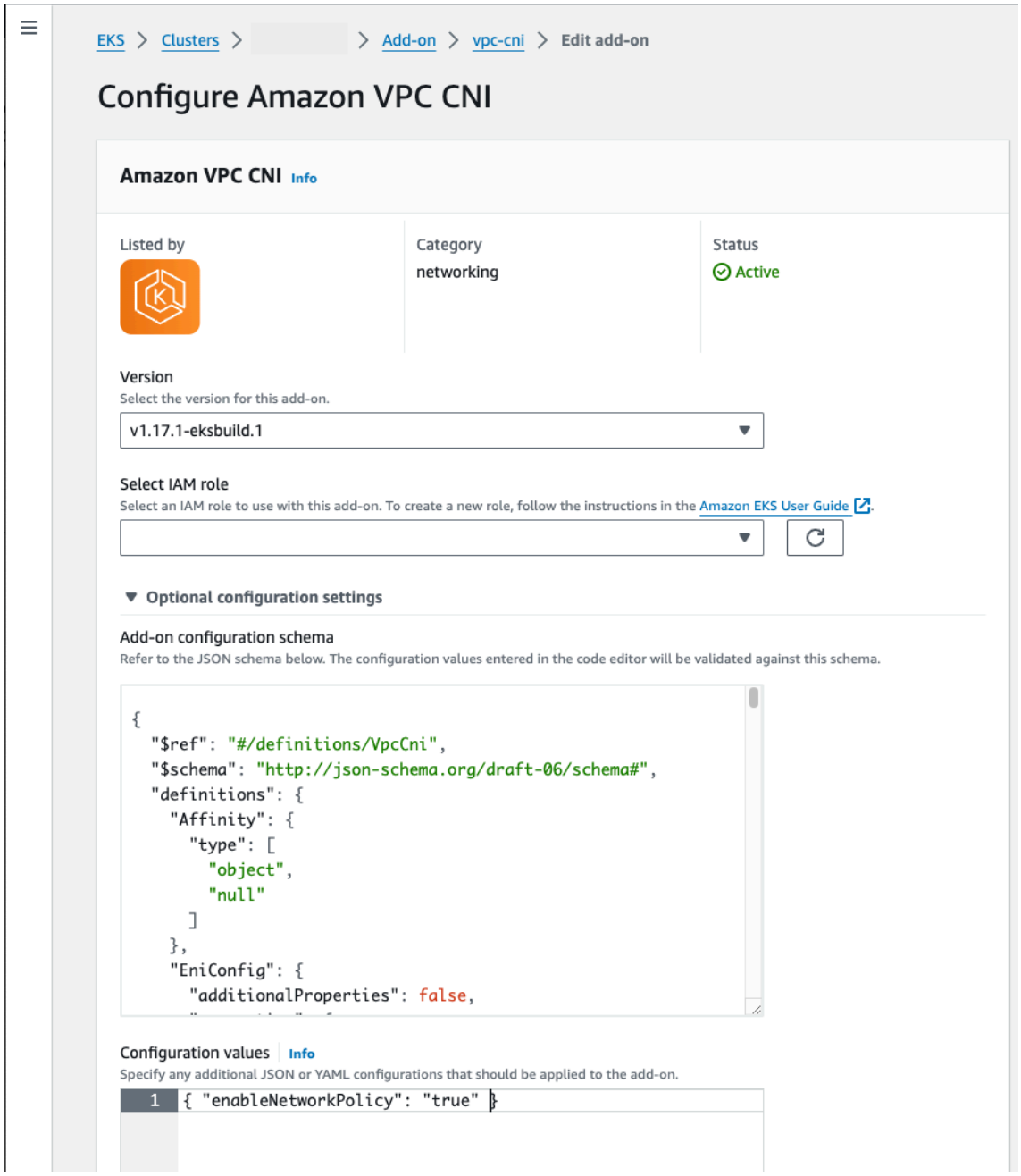
i. Seleccione la versión `v1.14.0-eksbuild.3` o posterior en la lista desplegable Versión.

ii. Seleccione Ajustes de configuración opcionales.

iii. Introduzca la clave JSON `"enableNetworkPolicy":` y el valor `"true"` en Valores de configuración. El texto resultante debe ser un objeto JSON válido. Si esta clave y este valor son los únicos datos del cuadro de texto, rodee la clave y el valor entre corchetes `{}`. En el siguiente ejemplo se muestra que la política de red está habilitada:

```
{ "enableNetworkPolicy": "true" }
```

En la siguiente captura de pantalla se muestra un ejemplo de este escenario.



AWS CLI

- Ejecute el siguiente comando de la AWS CLI. Reemplace `my-cluster` por el nombre del clúster y el ARN del rol de IAM por el rol que va a usar.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni
 --addon-version v1.14.0-eksbuild.3 \
```

```
    --service-account-role-arn arn:aws:iam::123456789012:role/
AmazonEKSVPCCNIRole \
    --resolve-conflicts PRESERVE --configuration-values
 '{"enableNetworkPolicy": "true"}'
```

- Complemento autoadministrado

  Helm

  Si ha instalado el Amazon VPC CNI plugin for Kubernetes mediante `helm`, puede actualizar la configuración para habilitar la política de red.

  - Ejecute el siguiente comando para habilitar la política de red.

    ```
    helm upgrade --set enableNetworkPolicy=true aws-vpc-cni --namespace
     kube-system eks/aws-vpc-cni
    ```

  kubectl

  a. Abra el `ConfigMap` de `amazon-vpc-cni` en el editor.

     ```
     kubectl edit configmap -n kube-system amazon-vpc-cni -o yaml
     ```

  b. Añada la línea siguiente al `data` de `ConfigMap`.

     ```
     enable-network-policy-controller: "true"
     ```

     Una vez que haya añadido la línea, el `ConfigMap` debería tener un aspecto similar al siguiente ejemplo.

     ```
     apiVersion: v1
      kind: ConfigMap
      metadata:
       name: amazon-vpc-cni
       namespace: kube-system
      data:
       enable-network-policy-controller: "true"
     ```

  c. Abra el `DaemonSet` de `aws-node` en el editor.

```
kubectl edit daemonset -n kube-system aws-node
```

d. Sustituya el `false` por `true` en el argumento del comando `--enable-network-policy=false` en el `args:` del contenedor `aws-network-policy-agent` en el manifiesto del daemonset de `aws-node` del CNI de VPC.

```
    - args:
       - --enable-network-policy=true
```

3. Confirme que los pods de `aws-node` se estén ejecutando en su clúster.

```
kubectl get pods -n kube-system | grep 'aws-node\|amazon'
```

Un ejemplo de salida sería el siguiente.

```
aws-node-gmqp7                                          2/2     Running   1 (24h
 ago)   24h
aws-node-prnsh                                          2/2     Running   1 (24h
 ago)   24h
```

Si la política de red está habilitada, hay 2 contenedores en los pods de `aws-node`. En versiones anteriores, y si la política de red está deshabilitada, solo hay un contenedor en los pods de `aws-node`.

Ahora puede implementar políticas de red de Kubernetes en su clúster. Para obtener más información, consulte Políticas de red de Kubernetes.

Demostración Stars de la política de red

Esta demostración crea un frontend, un backend y un servicio cliente en el clúster de Amazon EKS. La demostración también crea una interfaz gráfica de usuario de administración que muestra las rutas de entrada y salida disponibles entre los servicios. Le recomendamos que complete la demostración en un clúster en el que no ejecute cargas de trabajo de producción.

Cuando aún no se ha creado ninguna política de red, todos los servicios pueden comunicarse en ambas direcciones. Después de aplicar las políticas de red, puede ver que el cliente solo puede comunicarse con el servicio frontend y el backend solo puede aceptar tráfico del frontend.

Para ejecutar la demostración de política Stars

1. Aplique los servicios de frontend, backend, cliente e interfaz de usuario de administración:

```
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/namespace.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/management-ui.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/backend.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/frontend.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/client.yaml
```

2. Visualice todos los Pods en el clúster.

```
kubectl get pods -A
```

Un ejemplo de salida sería el siguiente.

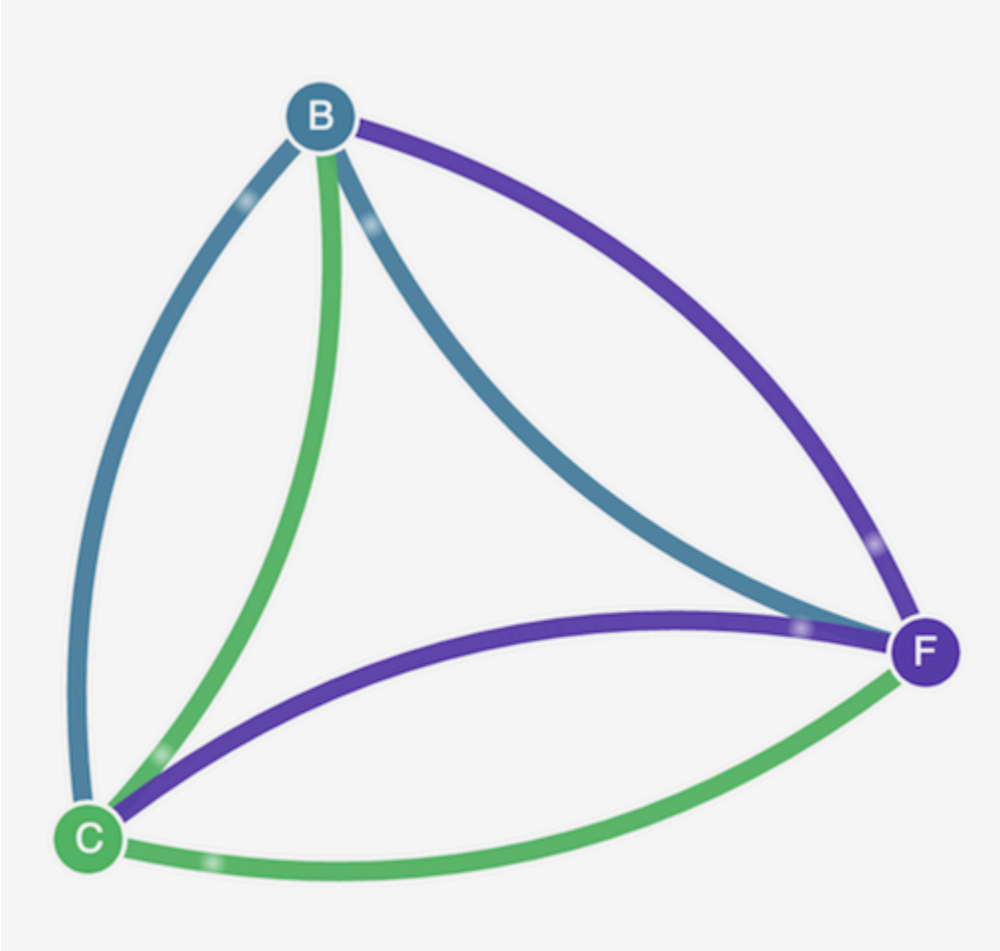
En la salida, debería ver pods en los espacios de nombres que se muestran en la siguiente salida. Los valores de *NAMES* y el número de pods de la columna READY son diferentes de los de la siguiente salida. No continúe hasta que vea pods con nombres similares y todos tengan Running en la columna STATUS.

```
NAMESPACE         NAME                                       READY   STATUS
 RESTARTS   AGE
[...]
client            client-xlffc                               1/1     Running   0
       5m19s
[...]
management-ui     management-ui-qrb2g                        1/1     Running   0
       5m24s
stars             backend-sz87q                              1/1     Running   0
       5m23s
stars             frontend-cscnf                             1/1     Running   0
       5m21s
[...]
```

3. Para conectarse a la interfaz de usuario de administración, conéctese a la EXTERNAL-IP del servicio que se ejecuta en el clúster:

```
kubectl get service/management-ui -n management-ui
```

4. Abra el navegador en la ubicación del paso anterior. Debería ver la interfaz de usuario de administración. El nodo C es el servicio cliente, el nodo F es el servicio frontend y el nodo B es el servicio backend. Cada nodo tiene acceso de comunicación completo a todos los demás nodos, como indican las líneas gruesas de colores.



5. Aplique la siguiente política de red en los espacios de nombres de `stars` y `client` para aislar los servicios entre sí:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
spec:
  podSelector:
    matchLabels: {}
```

Puede usar los siguientes comandos para aplicar la política a ambos espacios de nombres:

```
kubectl apply -n stars -f https://eksworkshop.com/beginner/120_network-policies/
calico/stars_policy_demo/apply_network_policies.files/default-deny.yaml
kubectl apply -n client -f https://eksworkshop.com/beginner/120_network-policies/
calico/stars_policy_demo/apply_network_policies.files/default-deny.yaml
```

6. Actualice su navegador. Verá que la interfaz de usuario de administración ya no tiene acceso a los nodos, que dejan de aparecer en la interfaz de usuario.
7. Aplique las siguientes políticas de red distintas para permitir el acceso de la interfaz de usuario de administración a los servicios. Aplique esta política para permitir la interfaz de usuario:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: allow-ui
spec:
  podSelector:
    matchLabels: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              role: management-ui
```

Aplique esta política para permitir el cliente:

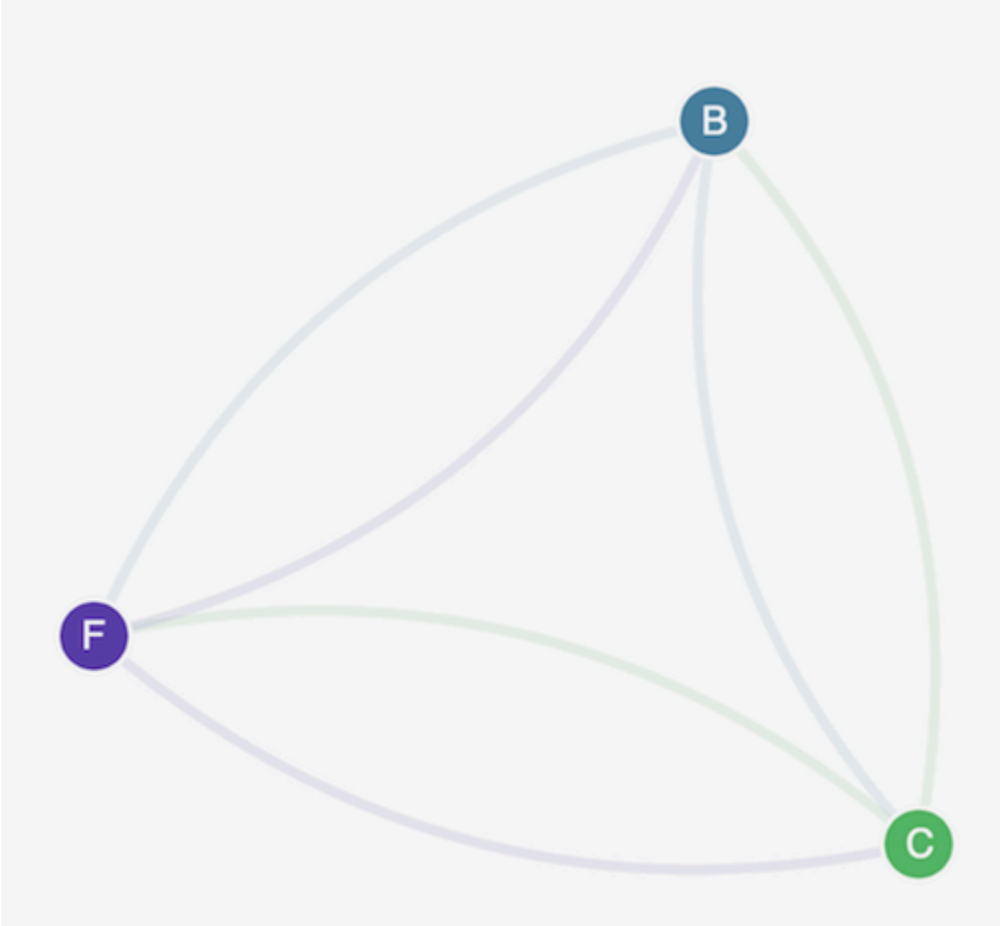
```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: client
  name: allow-ui
spec:
  podSelector:
    matchLabels: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
```

```
              role: management-ui
```

Puede usar los siguientes comandos para aplicar ambas políticas:

```
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/apply_network_policies.files/allow-ui.yaml
kubectl apply -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/apply_network_policies.files/allow-ui-client.yaml
```

8. Actualice su navegador. Verá que la interfaz de usuario de administración tiene de nuevo acceso a los nodos, pero los nodos no pueden comunicarse entre sí.

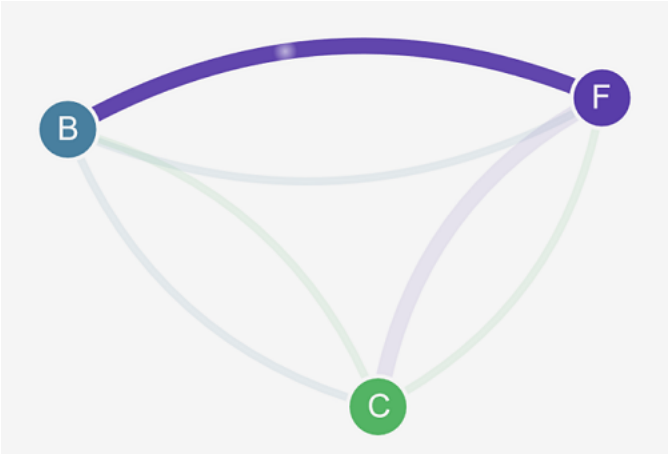


9. Aplique la siguiente política de red para permitir el tráfico desde el servicio frontend hacia el servicio backend:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
```

```
  name: backend-policy
spec:
  podSelector:
    matchLabels:
      role: backend
  ingress:
    - from:
        - podSelector:
            matchLabels:
              role: frontend
      ports:
        - protocol: TCP
          port: 6379
```

10. Actualice su navegador. Puede ver que el frontend puede comunicarse con el backend.

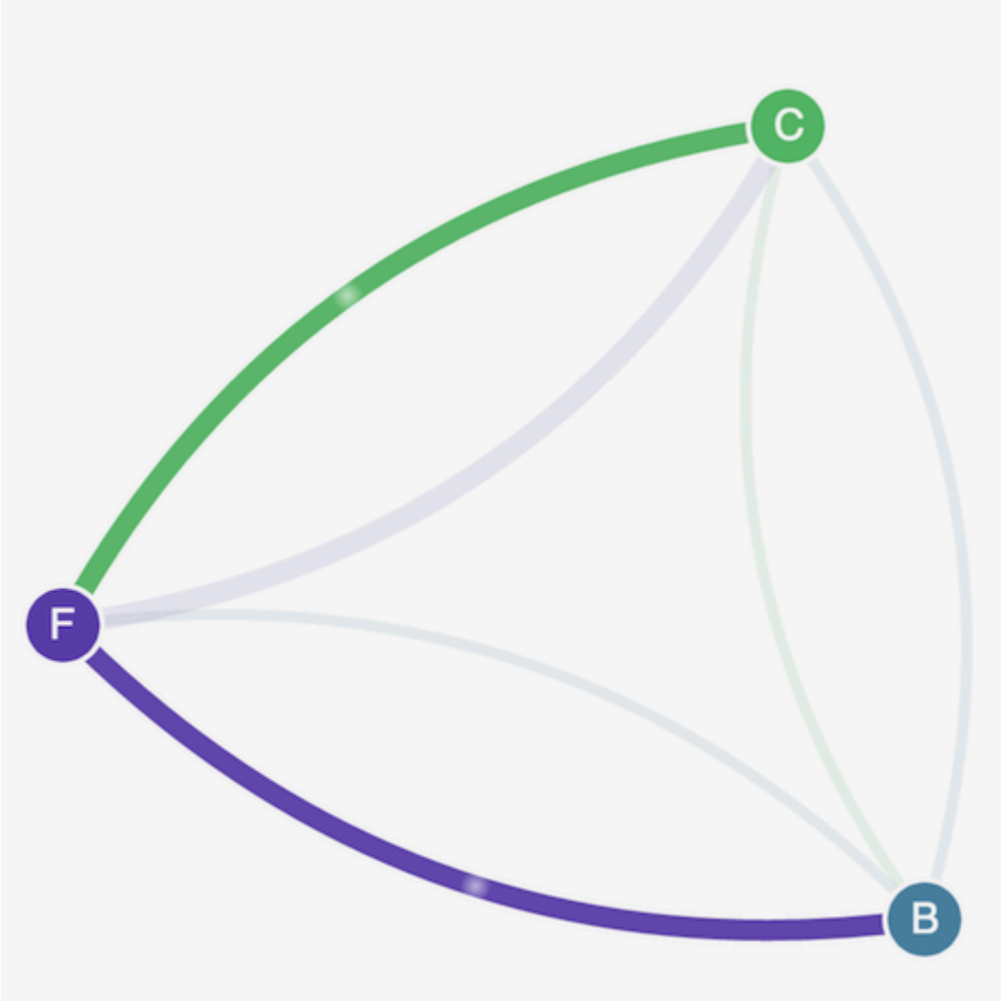


11. Aplique la siguiente política de red para permitir el tráfico desde el cliente hacia el servicio frontend:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  namespace: stars
  name: frontend-policy
spec:
  podSelector:
    matchLabels:
      role: frontend
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
```

```
            role: client
    ports:
      - protocol: TCP
        port: 80
```

12. Actualice su navegador. Verá que el cliente puede comunicarse con el servicio frontend. El servicio frontend puede seguir comunicándose con el servicio backend.



13. (Opcional) Cuando haya terminado con la demostración, puede eliminar sus recursos.

```
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/client.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/frontend.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/backend.yaml
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/management-ui.yaml
```

```
kubectl delete -f https://eksworkshop.com/beginner/120_network-policies/calico/
stars_policy_demo/create_resources.files/namespace.yaml
```

Incluso después de eliminar los recursos, todavía puede haber puntos de conexión de las políticas de red en los nodos que podrían interferir de manera inesperada con las redes en el clúster. La única forma segura de eliminar estas reglas es reiniciar los nodos o terminar todos los nodos y reciclarlos. Para terminar todos los nodos, establezca el recuento deseado del grupo de Auto Scaling en 0 y, a continuación, realice una copia de seguridad en el número deseado o simplemente termine los nodos.

Solución de problemas de las políticas de red

Puede solucionar problemas e investigar las conexiones de red que utilizan políticas de red leyendo los Registros de políticas de red y ejecutando las herramientas del eBPF SDK.

Registros de políticas de red

Se registra si una política de red permite o deniega las conexiones en los registros de flujo. Los registros de políticas de red de cada nodo incluyen los registros de flujo de cada pod que tiene una política de red. Los registros de políticas de red se almacenan en `/var/log/aws-routed-eni/network-policy-agent.log`. A continuación se muestra un ejemplo de un archivo `network-policy-agent.log`:

```
{"level":"info","timestamp":"2023-05-30T16:05:32.573Z","logger":"ebpf-
client","msg":"Flow Info: ","Src
IP":"192.168.87.155","Src Port":38971,"Dest IP":"64.6.160","Dest
Port":53,"Proto":"UDP","Verdict":"ACCEPT"}
```

Los registros de políticas de red está deshabilitados de manera predeterminada. Para habilitar los registros de políticas de red, siga estos pasos:

> **Note**
>
> Los registros de políticas de red requieren 1 vCPU adicional para el contenedor `aws-network-policy-agent` del manifiesto del daemonset de `aws-node` del CNI de la VPC.

Complemento de Amazon EKS

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento CNI de Amazon VPC.
3. Elija la pestaña Complementos.
4. Seleccione la casilla situada en la parte superior derecha del cuadro y, a continuación, elija Edit (Editar).
5. En la página Configurar el ***nombre del complemento***:

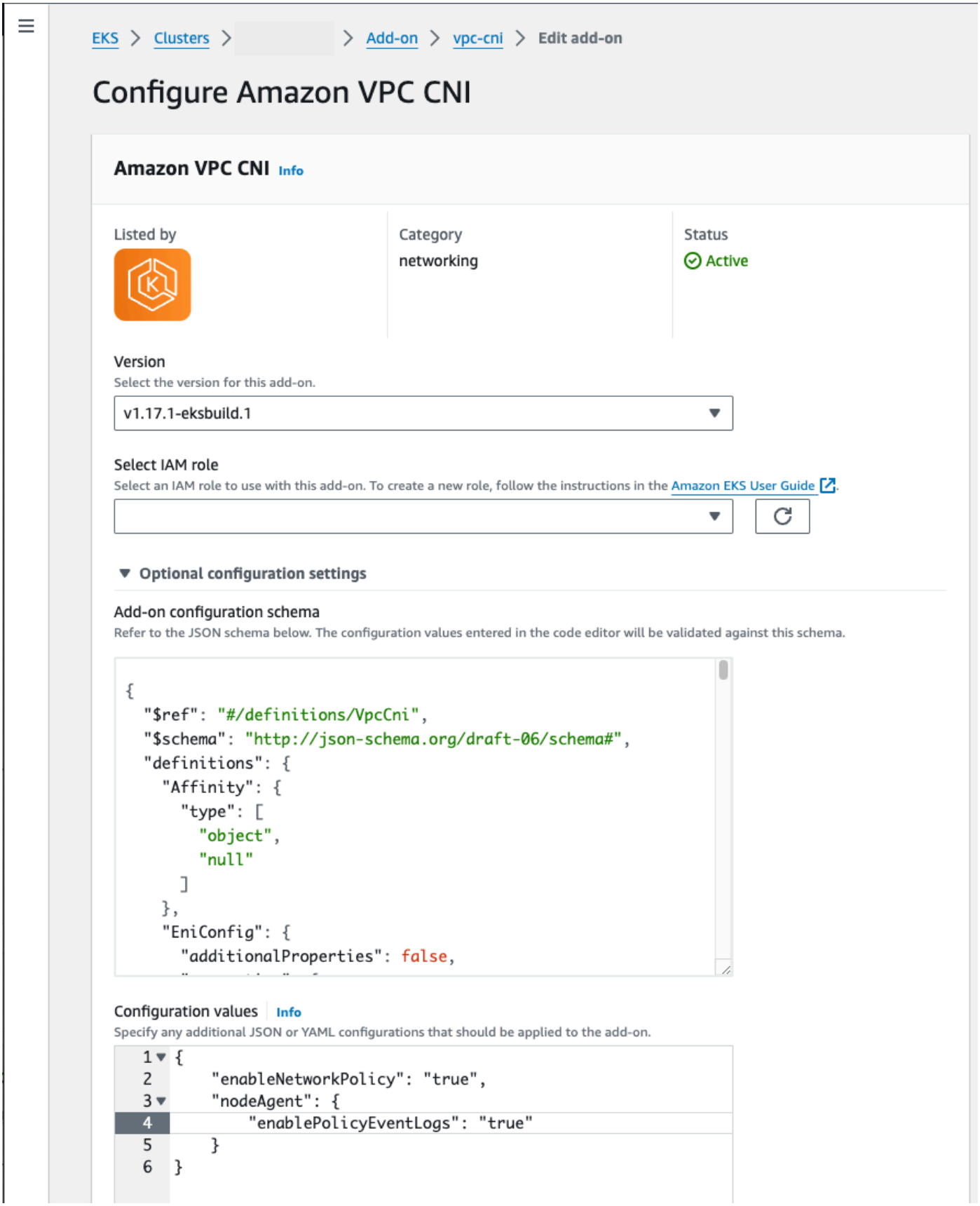
   a. Seleccione la versión `v1.14.0-eksbuild.3` o posterior en la lista desplegable Versión.

   b. Seleccione Ajustes de configuración opcionales.

   c. Introduzca la clave de JSON de nivel superior `"nodeAgent":` y el valor en un objeto con una clave `"enablePolicyEventLogs":` y un valor de `"true"` en Valores de configuración. El texto resultante debe ser un objeto JSON válido. En el siguiente ejemplo se muestra que la política de red y los registros de políticas de red están habilitados, y que estos últimos se envían a los Registros de CloudWatch:

```
{
    "enableNetworkPolicy": "true",
    "nodeAgent": {
        "enablePolicyEventLogs": "true"
    }
}
```

En la siguiente captura de pantalla se muestra un ejemplo de este escenario.


EKS > Clusters > > Add-on > vpc-cni > Edit add-on
Configure Amazon VPC CNI
Amazon VPC CNI Info
Listed by
Category
networking
Status
Active
Version
Select the version for this add-on.
v1.17.1-eksbuild.1
Select IAM role
Select an IAM role to use with this add-on. To create a new role, follow the instructions in the Amazon EKS User Guide.
Optional configuration settings
Add-on configuration schema
Refer to the JSON schema below. The configuration values entered in the code editor will be validated against this schema.
{
"$ref": "#/definitions/VpcCni",
"$schema": "http://json-schema.org/draft-06/schema#",
"definitions": {
"Affinity": {
"type": [
"object",
"null"
]
},
"EniConfig": {
"additionalProperties": false,
Configuration values Info
Specify any additional JSON or YAML configurations that should be applied to the add-on.
1 {
2 "enableNetworkPolicy": "true",
3 "nodeAgent": {
4 "enablePolicyEventLogs": "true"
5 }
6 }

AWS CLI

- Ejecute el siguiente comando de la AWS CLI. Reemplace `my-cluster` por el nombre del clúster y el ARN del rol de IAM por el rol que va a usar.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.14.0-eksbuild.3 \
    --service-account-role-arn arn:aws:iam::123456789012:role/
AmazonEKSVPCCNIRole \
    --resolve-conflicts PRESERVE --configuration-values '{"nodeAgent":
 {"enablePolicyEventLogs": "true"}}'
```

Complemento autoadministrado

Helm

Si ha instalado Amazon VPC CNI plugin for Kubernetes mediante `helm`, puede actualizar la configuración para escribir los registros de políticas de red.

- Ejecute el siguiente comando para habilitar la política de red.

```
helm upgrade --set nodeAgent.enablePolicyEventLogs=true aws-vpc-cni --namespace
 kube-system eks/aws-vpc-cni
```

kubectl

Si ha instalado Amazon VPC CNI plugin for Kubernetes mediante `kubectl`, puede actualizar la configuración para escribir los registros de políticas de red.

1. Abra el `DaemonSet` de `aws-node` en el editor.

```
kubectl edit daemonset -n kube-system aws-node
```

2. Sustituya el `false` por `true` en el argumento del comando `--enable-policy-event-logs=false` en el `args:` del contenedor `aws-network-policy-agent` en el manifiesto del daemonset de `aws-node` del CNI de VPC.

```
     - args:
```

```
        - --enable-policy-event-logs=true
```

Enviar los registros de políticas de red a los Registros de Amazon CloudWatch

Puede supervisar los registros de políticas de red mediante servicios como los Registros de Amazon CloudWatch. Puede usar los siguientes métodos para enviar los registros de políticas de red a los Registros de CloudWatch.

En el caso de los clústeres de EKS, los registros de políticas se ubicarán en `/aws/eks/`*`cluster-name`*`/cluster/` y, en el caso de los clústeres K8S autoadministrados, los registros se colocarán en`/aws/k8s-cluster/cluster/`.

Enviar los registros de políticas de red con el Amazon VPC CNI plugin for Kubernetes

Si habilita la política de red, se añade un segundo contenedor a los pods de `aws-node` para un agente de nodos. Este agente de nodos puede enviar los registros de políticas de red a los Registros de CloudWatch.

> **Note**
>
> El agente de nodos solo envía los registros de políticas de red. No se incluyen otros registros creados por el CNI de VPC.

Requisitos previos

- Añada los siguientes permisos como una política individual o independiente al rol de IAM que está utilizando para el CNI de VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Complemento de Amazon EKS

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clústeres y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento CNI de Amazon VPC.
3. Elija la pestaña Complementos.
4. Seleccione la casilla situada en la parte superior derecha del cuadro y, a continuación, elija Edit (Editar).
5. En la página Configurar el ***nombre del complemento***:

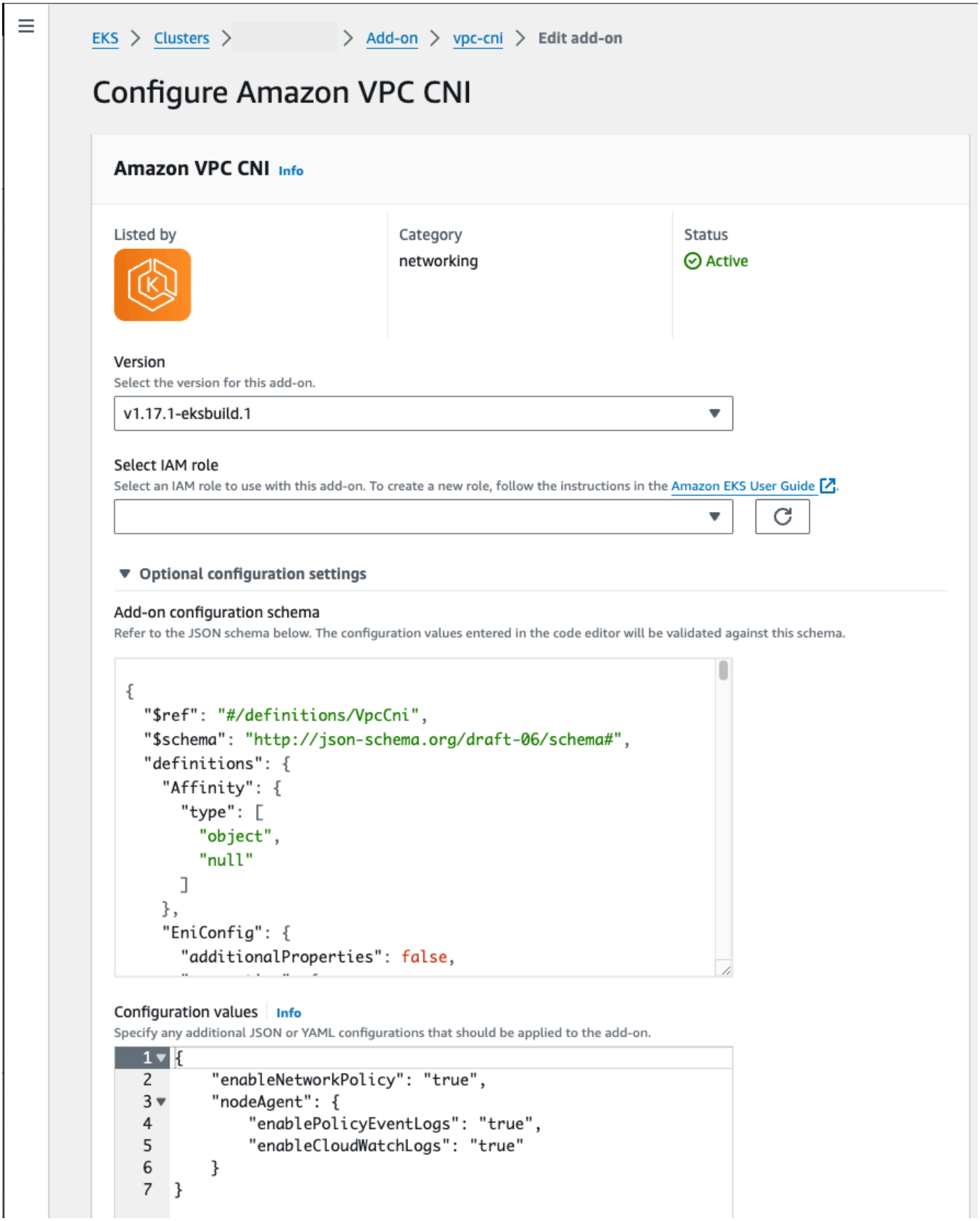
    a. Seleccione la versión `v1.14.0-eksbuild.3` o posterior en la lista desplegable Versión.

    b. Seleccione Ajustes de configuración opcionales.

    c. Introduzca la clave de JSON de nivel superior `"nodeAgent":` y el valor en un objeto con una clave `"enableCloudWatchLogs":` y un valor de `"true"` en Valores de configuración. El texto resultante debe ser un objeto JSON válido. En el siguiente ejemplo se muestra que la política de red y los registros de políticas de red están habilitados, y que estos últimos se envían a los Registros de CloudWatch:

    ```
    {
        "enableNetworkPolicy": "true",
        "nodeAgent": {
            "enablePolicyEventLogs": "true",
            "enableCloudWatchLogs": "true",
        }
    }
    ```

En la siguiente captura de pantalla se muestra un ejemplo de este escenario.


EKS > Clusters > > Add-on > vpc-cni > Edit add-on
Configure Amazon VPC CNI
Amazon VPC CNI Info
Listed by
Category
networking
Status
Active
Version
Select the version for this add-on.
v1.17.1-eksbuild.1
Select IAM role
Select an IAM role to use with this add-on. To create a new role, follow the instructions in the Amazon EKS User Guide.
Optional configuration settings
Add-on configuration schema
Refer to the JSON schema below. The configuration values entered in the code editor will be validated against this schema.
{
"$ref": "#/definitions/VpcCni",
"$schema": "http://json-schema.org/draft-06/schema#",
"definitions": {
"Affinity": {
"type": [
"object",
"null"
]
},
"EniConfig": {
"additionalProperties": false,
Configuration values Info
Specify any additional JSON or YAML configurations that should be applied to the add-on.
1 {
2 "enableNetworkPolicy": "true",
3 "nodeAgent": {
4 "enablePolicyEventLogs": "true",
5 "enableCloudWatchLogs": "true"
6 }
7 }

AWS CLI

- Ejecute el siguiente comando de la AWS CLI. Reemplace `my-cluster` por el nombre del clúster y el ARN del rol de IAM por el rol que va a usar.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version v1.14.0-eksbuild.3 \
    --service-account-role-arn arn:aws:iam::123456789012:role/
AmazonEKSVPCCNIRole \
    --resolve-conflicts PRESERVE --configuration-values '{"nodeAgent":
 {"enablePolicyEventLogs": "true", "enableCloudWatchLogs": "true"}}'
```

Complemento autoadministrado

Helm

Si ha instalado el Amazon VPC CNI plugin for Kubernetes mediante `helm`, puede actualizar la configuración para enviar los registros de políticas de red a los Registros de CloudWatch.

- Ejecute el siguiente comando para habilitar los registros de políticas de red y enviarlos a los Registros de CloudWatch.

```
helm upgrade --set nodeAgent.enablePolicyEventLogs=true --set
 nodeAgent.enableCloudWatchLogs=true aws-vpc-cni --namespace kube-system eks/
aws-vpc-cni
```

kubectl

1. Abra el `DaemonSet` de `aws-node` en el editor.

```
kubectl edit daemonset -n kube-system aws-node
```

2. Sustituya `false` por `true` en los dos argumentos de comando `--enable-policy-event-logs=false` y `--enable-cloudwatch-logs=false` de `args:` del contenedor `aws-network-policy-agent` en el manifiesto del daemonset de `aws-node` del CNI de la VPC.

```
    - args:
       - --enable-policy-event-logs=true
```

```
        - --enable-cloudwatch-logs=true
```

Enviar los registros de políticas de red con el daemonset de Fluent Bit

Si utiliza Fluent Bit en un daemonset para enviar registros desde sus nodos, puede agregar una configuración para incluir los registros de políticas de red de las políticas de red. Puede utilizar la siguiente configuración de ejemplo:

```
    [INPUT]
        Name              tail
        Tag               eksnp.*
        Path              /var/log/aws-routed-eni/network-policy-agent*.log
        Parser            json
        DB                /var/log/aws-routed-eni/flb_npagent.db
        Mem_Buf_Limit     5MB
        Skip_Long_Lines   On
        Refresh_Interval  10
```

SDK de eBPF incluido

El Amazon VPC CNI plugin for Kubernetes instala la colección de herramientas del SDK de eBPF en los nodos. Puede utilizar las herramientas del SDK de eBPF para identificar problemas con las políticas de red. Por ejemplo, el siguiente comando muestra los programas que se ejecutan en el nodo.

```
sudo /opt/cni/bin/aws-eks-na-cli ebpf progs
```

Para ejecutar este comando, puede utilizar cualquier método para conectarse al nodo.

Políticas de red de Kubernetes

Para implementar políticas de red de Kubernetes, cree objetos `NetworkPolicy` de Kubernetes e impleméntelos en su clúster. Los objetos `NetworkPolicy` están limitados a un espacio de nombres. Implemente políticas para permitir o denegar el tráfico entre Pods en función de los selectores de etiquetas, los espacios de nombres y los rangos de direcciones IP. Para obtener más información sobre cómo crear objetos `NetworkPolicy`, consulte Políticas de red en la documentación de Kubernetes.

La aplicación de los objetos `NetworkPolicy` de Kubernetes se implementa mediante el Extended Berkeley Packet Filter (eBPF). En relación con las implementaciones basadas en `iptables`, ofrece

características de rendimiento y latencia más bajas, que incluyen una menor utilización de la CPU y evitan las búsquedas secuenciales. Además, las sondas de eBPF proporcionan acceso a datos en contexto que ayudan a depurar problemas complejos a nivel del núcleo y a mejorar la observabilidad. Amazon EKS es compatible con un exportador basado en eBPF que aprovecha las sondas para registrar los resultados de las políticas en cada nodo y exportar los datos a recopiladores de registros externos para facilitar la depuración. Para obtener más información, consulte la Documentación de eBPF.

Redes personalizadas para los pods

De forma predeterminada, cuando el Amazon VPC CNI plugin for Kubernetes crea interfaces de red elásticas secundarias (interfaces de red) para su nodo Amazon EC2, las crea en la misma subred que la interfaz de red principal del nodo. También asocia los mismos grupos de seguridad a la interfaz de red secundaria asociada a la interfaz de red principal. Por uno o varios de los siguientes motivos, es posible que desee que el complemento cree interfaces de red secundarias en una subred diferente o desee asociar distintos grupos de seguridad a las interfaces de red secundarias, o ambas:

- Hay un número limitado de direcciones `IPv4` que están disponibles en la subred en la que se encuentra la interfaz de red principal. Esto podría limitar el número de Pods que puede crear en la subred. Si utiliza una subred diferente para las interfaces de red secundarias, puede aumentar el número de direcciones `IPv4` disponibles para Pods.
- Por motivos de seguridad, sus Pods podrían tener que utilizar distintos grupos de seguridad o subredes que la interfaz de red principal del nodo.
- Los nodos se encuentran configurados en subredes públicas y debe ubicar los Pods en subredes privadas. La tabla de enrutamiento asociada a una subred pública incluye una puerta de enlace de Internet. La tabla de enrutamiento asociada a una subred privada no incluye ninguna puerta de enlace de Internet.

Consideraciones

- Con las redes personalizadas habilitadas, no se asignan direcciones IP asignadas a la interfaz de red principal a los Pods. Solo se asignan direcciones IP de interfaces de red secundarias a `Pods`.
- Si el clúster utiliza la familia `IPv6`, no puede utilizar redes personalizadas.
- Si planea utilizar redes personalizadas solo para ayudar a aliviar el agotamiento de direcciones `IPv4`, puede crear un clúster mediante la familia `IPv6` en su lugar. Para obtener más información, consulte Direcciones `IPv6` de clústeres, Pods y services.

- Si bien los Pods implementados en las subredes especificadas para interfaces de red secundarias pueden utilizar subredes y grupos de seguridad diferentes a los de la interfaz de red principal del nodo, las subredes y los grupos de seguridad deben estar en la misma VPC que el nodo.

Requisitos previos

- Estar familiarizado con cómo Amazon VPC CNI plugin for Kubernetes crea interfaces de red secundarias y asigna direcciones IP a los Pods. Para obtener más información, consulte Asignación de ENI en GitHub.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- Le recomendamos que siga los pasos de este tema en un intérprete de comandos Bash. Si no está utilizando un intérprete de comandos Bash, algunos comandos de script, como los caracteres de continuación de línea y la forma en que se establecen y utilizan las variables, requieren ajustes para su intérprete de comandos. Además, las reglas de entrecomillado y escape de su intérprete de comandos pueden ser diferentes. Para obtener más información, consulte Uso de entrecomillado de cadenas en la AWS CLI de la Guía del usuario de la AWS Command Line Interface.

Para este tutorial, le recomendamos utilizar el *`example values`*, excepto en los casos en que se indique que los reemplace. Puede reemplazar cualquier *`example value`* al completar los pasos del clúster de producción. Recomendamos completar todos los pasos en la misma terminal. Esto se

debe a que las variables se establecen y utilizan a lo largo de los pasos y no existirán en terminales diferentes.

Los comandos de este tema se formatean según las convenciones que se indican en Uso de los ejemplos de AWS CLI. Si ejecuta comandos desde la línea de comandos con recursos que se encuentran en una Región de AWS diferente a la Región de AWS predeterminada definida en el perfil de la AWS CLI que está utilizando, tendrá que agregar **`--region`** ***`region-code`*** a los comandos.

Cuando desee implementar redes personalizadas en su clúster de producción, vaya a Paso 2: Configurar la VPC.

Paso 1: crear una VPC de prueba y un clúster

Pasos para crear un clúster

Los siguientes procedimientos le ayudan a crear una VPC de prueba y un clúster, y a configurar redes personalizadas para ese clúster. No recomendamos utilizar el clúster de pruebas para cargas de trabajo de producción porque en este tema no se tratan varias características no relacionadas que podría utilizar en el clúster de producción. Para obtener más información, consulte Creación de un clúster de Amazon EKS.

1. Defina algunas variables para utilizarlas en los pasos restantes.

```
export cluster_name=my-custom-networking-cluster
account_id=$(aws sts get-caller-identity --query Account --output text)
```

2. Cree una VPC.

   1. Cree una VPC mediante una plantilla de Amazon EKS AWS CloudFormation.

```
aws cloudformation create-stack --stack-name my-eks-custom-networking-vpc \
  --template-url https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml \
  --parameters ParameterKey=VpcBlock,ParameterValue=192.168.0.0/24 \
  ParameterKey=PrivateSubnet01Block,ParameterValue=192.168.0.64/27 \
  ParameterKey=PrivateSubnet02Block,ParameterValue=192.168.0.96/27 \
  ParameterKey=PublicSubnet01Block,ParameterValue=192.168.0.0/27 \
  ParameterKey=PublicSubnet02Block,ParameterValue=192.168.0.32/27
```

La pila AWS CloudFormation tarda unos minutos en crearse. Para verificar el estado de implementación de la pila, ejecute el siguiente comando.

```
aws cloudformation describe-stacks --stack-name my-eks-custom-networking-vpc --
query Stacks\[\].StackStatus  --output text
```

No continúe con el siguiente paso hasta que la salida del comando sea `CREATE_COMPLETE`.

2. Defina variables con los valores de los ID de subred privada creados por la plantilla.

```
subnet_id_1=$(aws cloudformation describe-stack-resources --stack-name my-eks-
custom-networking-vpc \
    --query "StackResources[?
LogicalResourceId=='PrivateSubnet01'].PhysicalResourceId" --output text)
subnet_id_2=$(aws cloudformation describe-stack-resources --stack-name my-eks-
custom-networking-vpc \
    --query "StackResources[?
LogicalResourceId=='PrivateSubnet02'].PhysicalResourceId" --output text)
```

3. Defina variables con las zonas de disponibilidad de las subredes recuperadas en el paso anterior.

```
az_1=$(aws ec2 describe-subnets --subnet-ids $subnet_id_1 --query
 'Subnets[*].AvailabilityZone' --output text)
az_2=$(aws ec2 describe-subnets --subnet-ids $subnet_id_2 --query
 'Subnets[*].AvailabilityZone' --output text)
```

3. Cree un rol de IAM de clúster.

a. Ejecute el siguiente comando para crear un archivo de política de confianza JSON de IAM.

```
cat >eks-cluster-role-trust-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
EOF
```

b. Cree el rol de IAM del clúster de Amazon EKS. Si es necesario, prefacio `eks-cluster-role-trust-policy.json` con la ruta del equipo en la que escribió el archivo en el paso anterior. El comando asocia la política de confianza creada en el paso anterior al rol. Para crear un rol de IAM, a la entidad principal de IAM que está creando el rol se le debe asignar la acción `iam:CreateRole` (permiso).

```
aws iam create-role --role-name myCustomNetworkingAmazonEKSClusterRole --
assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
```

c. Adjunte la política administrada de IAM por Amazon EKS denominada `AmazonEKSClusterPolicy` al rol. Para adjuntar una política de IAM a una entidad principal de IAM, se debe asignar una de las siguientes acciones de IAM (permisos) a la entidad principal que adjunta la política: `iam:AttachUserPolicy` o `iam:AttachRolePolicy`.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSClusterPolicy --role-name myCustomNetworkingAmazonEKSClusterRole
```

4. Cree un clúster de Amazon EKS y configure su dispositivo para que se comunique con él.

a. Cree un clúster.

```
aws eks create-cluster --name my-custom-networking-cluster \
   --role-arn arn:aws:iam::$account_id:role/
myCustomNetworkingAmazonEKSClusterRole \
   --resources-vpc-config subnetIds=$subnet_id_1","$subnet_id_2
```

> **Note**
>
> Es posible que reciba un error que indique que una de las zonas de disponibilidad de la solicitud no tiene capacidad suficiente para crear un clúster de Amazon EKS. Si esto ocurre, el mensaje de error indicará las zonas de disponibilidad que admiten un clúster nuevo. Intente crear el clúster de nuevo con al menos dos subredes

ubicadas en las zonas de disponibilidad admitidas para su cuenta. Para obtener más información, consulte Capacidad insuficiente.

b. El clúster tarda varios minutos en crearse. Ejecute el siguiente comando para verificar el estado de implementación del clúster.

```
aws eks describe-cluster --name my-custom-networking-cluster --query
 cluster.status
```

No continúe con el siguiente paso hasta que la salida del comando sea `"ACTIVE"`.

c. Configure `kubectl` para comunicarse con el clúster.

```
aws eks update-kubeconfig --name my-custom-networking-cluster
```

Paso 2: Configurar la VPC

Este tutorial requiere una VPC creada en Paso 1: crear una VPC de prueba y un clúster. En el caso de un clúster de producción, ajuste los pasos correspondientes a su VPC sustituyendo todos los *`example values`* con los suyos propios.

1. Confirme que la versión de Amazon VPC CNI plugin for Kubernetes instalada actualmente sea la más reciente. Para determinar la versión más reciente del tipo de complemento de Amazon EKS y actualizar su versión a ella, consulte Actualización de un complemento. Para determinar la versión más reciente del tipo de complemento autoadministrado y actualizar su versión a ella, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.
2. Recupere el ID de la VPC de su clúster y guárdelo en una variable para utilizarlo en un paso posterior. Para un clúster de producción, sustituya *`my-custom-networking-cluster`* con el nombre de su clúster.

```
vpc_id=$(aws eks describe-cluster --name my-custom-networking-cluster --query
 "cluster.resourcesVpcConfig.vpcId" --output text)
```

3. Asocie un bloque de enrutamiento entre dominios sin clases (CIDR) con la VPC de su clúster. El bloque CIDR no se puede solapar con ningún bloque CIDR asociado existente.

   1. Vea los bloques CIDR actuales asociados a la VPC.

```
aws ec2 describe-vpcs --vpc-ids $vpc_id \
    --query 'Vpcs[*].CidrBlockAssociationSet[*].{CIDRBlock: CidrBlock, State:
 CidrBlockState.State}' --out table
```

Un ejemplo de salida sería el siguiente.

```
---------------------------------
|          DescribeVpcs          |
+-----------------+--------------+
|    CIDRBlock    |    State     |
+-----------------+--------------+
|  192.168.0.0/24 |  associated  |
+-----------------+--------------+
```

2. Asocie un bloque CIDR adicional a su VPC. Para obtener más información, consulte Asociar un bloque de CIDR IPv4 adicional a su VPC en la Guía del usuario de Amazon VPC.

```
aws ec2 associate-vpc-cidr-block --vpc-id $vpc_id --cidr-block 192.168.1.0/24
```

3. Confirme que el nuevo bloque está asociado.

```
aws ec2 describe-vpcs --vpc-ids $vpc_id --query
 'Vpcs[*].CidrBlockAssociationSet[*].{CIDRBlock: CidrBlock, State:
 CidrBlockState.State}' --out table
```

Un ejemplo de salida sería el siguiente.

```
---------------------------------
|          DescribeVpcs          |
+-----------------+--------------+
|    CIDRBlock    |    State     |
+-----------------+--------------+
|  192.168.0.0/24 |  associated  |
|  192.168.1.0/24 |  associated  |
+-----------------+--------------+
```

No continúe con el siguiente paso hasta que el `State` del nuevo bloque CIDR sea `associated`.

4. Cree tantas subredes como desee utilizar en cada zona de disponibilidad en la que se encuentran las subredes existentes. Especifique un bloque CIDR que se encuentra dentro del bloque CIDR que asoció a la VPC en un paso anterior.

   1. Crea nuevas subredes. Las subredes deben crearse en un bloque CIDR de VPC diferente al que están las subredes existentes, pero en las mismas zonas de disponibilidad que las subredes existentes. En este ejemplo, se crea una subred en el nuevo bloque CIDR de cada zona de disponibilidad en la que existen las subredes privadas actuales. Los ID de las subredes creadas se almacenan en variables para usarlas en los pasos posteriores. Los valores `Name` coinciden con los valores asignados a las subredes creadas mediante la plantilla de Amazon EKS VPC en un paso anterior. No se requieren nombres. Puede utilizar diferentes nombres.

```
new_subnet_id_1=$(aws ec2 create-subnet --vpc-id $vpc_id --availability-zone
 $az_1 --cidr-block 192.168.1.0/27 \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=my-eks-
custom-networking-vpc-PrivateSubnet01},{Key=kubernetes.io/role/internal-
elb,Value=1}]' \
    --query Subnet.SubnetId --output text)
new_subnet_id_2=$(aws ec2 create-subnet --vpc-id $vpc_id --availability-zone
 $az_2 --cidr-block 192.168.1.32/27 \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=my-eks-
custom-networking-vpc-PrivateSubnet02},{Key=kubernetes.io/role/internal-
elb,Value=1}]' \
    --query Subnet.SubnetId --output text)
```

      Important

      De forma predeterminada, las nuevas subredes están asociadas implícitamente con su tabla de enrutamiento principal de VPC. Esta tabla de enrutamiento permite la comunicación entre todos los recursos que se implementan en la VPC. Sin embargo, no permite la comunicación con recursos que tienen direcciones IP que están fuera de los bloques CIDR asociados a la VPC. Puede asociar su propia tabla de enrutamiento a las subredes para cambiar este comportamiento. Para obtener más información, consulte Tablas de enrutamiento de subredes en la Guía del usuario de Amazon VPC.

   2. Para ver las subredes actuales en su VPC.

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=$vpc_id" \
```

```
    --query 'Subnets[*].{SubnetId: SubnetId,AvailabilityZone:
 AvailabilityZone,CidrBlock: CidrBlock}' \
    --output table
```

Un ejemplo de salida sería el siguiente.

```
-------------------------------------------------------------------
|                          DescribeSubnets                        |
+------------------+--------------------+---------------------------+
| AvailabilityZone |     CidrBlock      |         SubnetId          |
+------------------+--------------------+---------------------------+
|  us-west-2d      |  192.168.0.0/27    |     subnet-example1       |
|  us-west-2a      |  192.168.0.32/27   |     subnet-example2       |
|  us-west-2a      |  192.168.0.64/27   |     subnet-example3       |
|  us-west-2d      |  192.168.0.96/27   |     subnet-example4       |
|  us-west-2a      |  192.168.1.0/27    |     subnet-example5       |
|  us-west-2d      |  192.168.1.32/27   |     subnet-example6       |
+------------------+--------------------+---------------------------+
```

Puede ver que las subredes en el bloque CIDR `192.168.1.0` que ha creado se encuentran en las mismas zonas de disponibilidad que las subredes del bloque CIDR `192.168.0.0`.

Paso 3: Configure los recursos de Kubernetes

Para configurar los recursos de Kubernetes

1. Establezca la variable de entorno de `AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG` en `true` en el `aws-node` `DaemonSet`.

   ```
   kubectl set env daemonset aws-node -n kube-system
    AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG=true
   ```

2. Recupere el ID del grupo de seguridad del clúster y guárdelo en una variable para utilizarlo en un paso posterior. Amazon EKS crea automáticamente este grupo de seguridad cuando crea el clúster.

   ```
   cluster_security_group_id=$(aws eks describe-cluster --name $cluster_name --query
    cluster.resourcesVpcConfig.clusterSecurityGroupId --output text)
   ```

3. Cree un recurso personalizado `ENIConfig` para cada subred en la que desee programar Pods.

a. Cree un archivo único para cada configuración de interfaz de red.

Los siguientes comandos crean archivos `ENIConfig` por separado de las dos subredes que se han creado en un paso anterior. El valor de `name` debe ser único. El nombre es el mismo que la zona de disponibilidad en la que se encuentra la subred. El grupo de seguridad del clúster está asignado al `ENIConfig`.

```
cat >$az_1.yaml <<EOF
apiVersion: crd.k8s.amazonaws.com/v1alpha1
kind: ENIConfig
metadata:
  name: $az_1
spec:
  securityGroups:
    - $cluster_security_group_id
  subnet: $new_subnet_id_1
EOF
```

```
cat >$az_2.yaml <<EOF
apiVersion: crd.k8s.amazonaws.com/v1alpha1
kind: ENIConfig
metadata:
  name: $az_2
spec:
  securityGroups:
    - $cluster_security_group_id
  subnet: $new_subnet_id_2
EOF
```

Para un clúster de producción, puede realizar los siguientes cambios en los comandos anteriores:

- Reemplace *`$cluster_security_group_id`* con el ID de un grupo de seguridad existente que desea utilizar para cada `ENIConfig`.
- Recomendamos nombrar su `ENIConfigs` igual que la zona de disponibilidad para la que utilizará el `ENIConfig`, siempre que sea posible. Es posible que tenga que utilizar nombres diferentes para su `ENIConfigs` que los nombres de las zonas de disponibilidad por diversas razones. Por ejemplo, si tiene más de dos subredes en la misma zona de

disponibilidad y desea utilizarlas con redes personalizadas, necesita varias `ENIConfigs` para la misma zona de disponibilidad. Dado que cada `ENIConfig` requiere un nombre único, no puede nombrar más de una de sus `ENIConfigs` utilizando el nombre de zona de disponibilidad.

Si los nombres de sus `ENIConfig` no son todos iguales que los nombres de las zonas de disponibilidad, entonces reemplace *`$az_1`* y *`$az_2`* con sus propios nombres en los comandos anteriores y anote los nodos con la `ENIConfig` más adelante en este tutorial.

> **Note**
>
> Si no especifica un grupo de seguridad válido para utilizarlo con un clúster de producción y utiliza:
>
> - la versión `1.8.0` o posterior del Amazon VPC CNI plugin for de Kubernetes, entonces se utilizan los grupos de seguridad asociados a la principal interfaz de redes elástica del nodo.
> - una versión del Amazon VPC CNI plugin for Kubernetes anterior a `1.8.0`, entonces el grupo de seguridad predeterminado para la VPC se asignará a interfaces de red elásticas secundarias.

> **Important**
>
> - `AWS_VPC_K8S_CNI_EXTERNALSNAT=false` es un ajuste predeterminado de la configuración del complemento CNI de Amazon VPC para Kubernetes. Si utiliza la configuración predeterminada, el tráfico destinado a direcciones IP que no se encuentran dentro de uno de los bloques CIDR asociados a la VPC utiliza los grupos de seguridad y las subredes de la interfaz de red principal del nodo. Las subredes y los grupos de seguridad definidos en la `ENIConfigs` que se utilizan para crear interfaces de red secundarias no se utilizan para este tráfico. Para obtener más información sobre esta configuración, consulte SNAT para Pods.
> - Si también utiliza grupos de seguridad para Pods, el grupo de seguridad especificado en una `SecurityGroupPolicy` se utiliza en lugar del grupo de

seguridad especificado en la `ENIConfigs`. Para obtener más información, consulte Grupos de seguridad de Pods.

b. Aplique al clúster cada archivo de recursos personalizados que creó anteriormente con los siguientes comandos:

```
kubectl apply -f $az_1.yaml
kubectl apply -f $az_2.yaml
```

4. Confirmación de que se crearon `ENIConfigs`.

```
kubectl get ENIConfigs
```

Un ejemplo de salida sería el siguiente.

```
NAME         AGE
us-west-2a   117s
us-west-2d   105s
```

5. Si está habilitando redes personalizadas en un clúster de producción y da un nombre a su `ENIConfigs` que no sea la zona de disponibilidad para la que las utiliza, entonces pase al siguiente paso para implementar nodos de Amazon EC2.

Habilite Kubernetes para aplicar automáticamente la `ENIConfig` para una zona de disponibilidad en cualquier nuevo nodo de Amazon EC2 creado en su clúster.

1. Para ver el clúster de pruebas de este tutorial, vaya al siguiente paso.

   En caso de un clúster de producción, compruebe si existe una annotation con la clave `k8s.amazonaws.com/eniConfig` para la variable de entorno `ENI_CONFIG_ANNOTATION_DEF` en la especificación del contenedor para el `aws-node` `DaemonSet`.

   ```
   kubectl describe daemonset aws-node -n kube-system | grep
    ENI_CONFIG_ANNOTATION_DEF
   ```

   Si se devuelve la salida, la anotación existe. Si no se devuelve ninguna salida, entonces la variable no se establece. Para un clúster de producción, puede utilizar esta configuración o la

configuración del siguiente paso. Si utiliza este ajuste, anule el ajuste en el siguiente paso. En este tutorial, se utiliza la configuración del siguiente paso.

2. Actualice su `aws-node` `DaemonSet` para aplicar automáticamente la `ENIConfig` para una zona de disponibilidad para cualquier nuevo nodo de Amazon EC2 creado en su clúster.

```
kubectl set env daemonset aws-node -n kube-system
 ENI_CONFIG_LABEL_DEF=topology.kubernetes.io/zone
```

Paso 4: Implementar nodos de Amazon EC2

Para implementar nodos de Amazon EC2

1. Cree un rol de IAM de nodo.

   a. Ejecute el siguiente comando para crear un archivo de política de confianza JSON de IAM.

```
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

   b. Ejecute el siguiente comando para establecer una variable para el nombre de su rol. Puede reemplazar *myCustomNetworkingAmazonEKSNodeRole* con cualquier nombre que elija.

```
export node_role_name=myCustomNetworkingAmazonEKSNodeRole
```

   c. Cree el rol de IAM y almacene el nombre de recurso de Amazon (ARN) devuelto en una variable para usarla en un paso posterior.

```
node_role_arn=$(aws iam create-role --role-name $node_role_name --assume-role-
policy-document file://"node-role-trust-relationship.json" \
    --query Role.Arn --output text)
```

d. Adjunte al rol de IAM las tres políticas administradas por IAM necesarias.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name $node_role_name
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name $node_role_name
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
    --role-name $node_role_name
```

> **Important**
>
> Para simplificar este tutorial, la política `AmazonEKS_CNI_Policy` se adjunta al rol de IAM de nodo. Sin embargo, en un clúster de producción, recomendamos adjuntar la política a un rol de IAM independiente que se usa solo con el Amazon VPC CNI plugin for Kubernetes. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

2. Cree uno de los siguientes tipos de grupos de nodos. Para determinar el tipo de instancia que desea implementar, consulte Elección de un tipo de instancia de Amazon EC2. Para este tutorial, complete la opción Administrado, Sin plantilla de lanzamiento o con plantilla de lanzamiento sin un ID de AMI especificado. Si va a utilizar el grupo de nodos para cargas de trabajo de producción, le recomendamos que se familiarice con todas las opciones de grupo de nodos administrados y autoadministrados antes de implementar el grupo de nodos.

   - Administrado: implemente el grupo de nodos mediante una de las siguientes opciones:
     - Sin plantilla de lanzamiento o con plantilla de lanzamiento sin un ID de AMI especificado: complete el procedimiento indicado. Para este tutorial, usaremos los *`example values`*. Para un grupo de nodos de producción, sustituya todos los *`example values`* con los suyos propios. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe

empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales.

```
aws eks create-nodegroup --cluster-name $cluster_name --nodegroup-name my-
nodegroup \
    --subnets $subnet_id_1 $subnet_id_2 --instance-types t3.medium --node-role
 $node_role_arn
```

- Con plantilla de lanzamiento con un ID de AMI especificado

  1. Determine el número máximo de Pods que recomienda Amazon EKS para sus nodos. Siga las instrucciones de Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2, agregando **`--cni-custom-networking-enabled`** al paso tres de ese tema. Observe la salida de su uso en el siguiente paso.
  2. En la plantilla de lanzamiento, especifique un ID de AMI optimizado para Amazon EKS o una AMI personalizada creada a partir de la AMI optimizada para Amazon EKS y, luego, implemente el grupo de nodos mediante una plantilla de lanzamiento y proporcione los siguientes datos de usuario en la plantilla de lanzamiento. Estos datos de usuario pasan los argumentos en el archivo `bootstrap.sh`. Para obtener más información acerca del archivo del proceso de arranque, consulte bootstrap.sh en GitHub. Puede sustituir *20* por el valor del paso anterior (recomendado) o por un valor propio.

     ```
     /etc/eks/bootstrap.sh my-cluster --use-max-pods false --kubelet-extra-args
      '--max-pods=20'
     ```

     Si ha creado una AMI personalizada, pero no a partir de la AMI optimizada de Amazon EKS, debe crear personalmente la configuración.

- Autoadministrado

  1. Determine el número máximo de Pods que recomienda Amazon EKS para sus nodos. Siga las instrucciones de Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2, agregando **`--cni-custom-networking-enabled`** al paso tres de ese tema. Observe la salida de su uso en el siguiente paso.
  2. Implemente el grupo de nodos con las instrucciones en Lanzar nodos autoadministrados de Amazon Linux. Especifique el siguiente texto para el parámetro BootstrapArguments. Puede sustituir *20* por el valor del paso anterior (recomendado) o por un valor propio.

     ```
     --use-max-pods false --kubelet-extra-args '--max-pods=20'
     ```

> **Note**
>
> Si desea que los nodos de un clúster de producción admitan un número significativamente mayor de Pods, ejecute el script en Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2 de nuevo. Además, agregue la opción **`--cni-prefix-delegation-enabled`** al siguiente comando. Por ejemplo, se devuelve *`110`* para un tipo de instancia `m5.large`. Para obtener instrucciones sobre cómo habilitar esta capacidad, consulte Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2. Puede utilizar esta capacidad con redes personalizadas.

La creación de grupos de nodos tarda varios minutos. Puede comprobar el estado de la creación de un grupo de nodos administrados con el siguiente comando.

```
aws eks describe-nodegroup --cluster-name $cluster_name --nodegroup-name my-nodegroup --query nodegroup.status --output text
```

No continúe con el paso siguiente hasta que la salida devuelta sea `ACTIVE`.

3. Para ver el tutorial, puede omitir este paso.

   En caso de un clúster de producción, si no ha nombrado su `ENIConfigs` con el mismo nombre que la zona de disponibilidad para la que la utiliza, entonces debe anotar sus nodos con el nombre de `ENIConfig` que debe utilizarse con el nodo. Este paso no es necesario si solo tiene una subred en cada zona de disponibilidad y ha dado nombre a sus `ENIConfigs` con los mismos nombres que las zonas de disponibilidad. Esto se debe a que el Amazon VPC CNI plugin for de Kubernetes asocia automáticamente la `ENIConfig` correcta con el nodo para usted cuando lo habilitó para esto en un paso anterior.

   a. Obtenga la lista de nodos del clúster.

      ```
      kubectl get nodes
      ```

      Un ejemplo de salida sería el siguiente.

      ```
      NAME                                          STATUS   ROLES    AGE     VERSION
      ```

```
ip-192-168-0-126.us-west-2.compute.internal   Ready    <none>   8m49s
 v1.22.9-eks-810597c
ip-192-168-0-92.us-west-2.compute.internal    Ready    <none>   8m34s
 v1.22.9-eks-810597c
```

b. Determine en qué zona de disponibilidad se encuentra cada nodo. Ejecute el siguiente comando para cada nodo que se devolvió en el paso anterior.

```
aws ec2 describe-instances --filters Name=network-interface.private-dns-
name,Values=ip-192-168-0-126.us-west-2.compute.internal \
--query 'Reservations[].Instances[].{AvailabilityZone:
 Placement.AvailabilityZone, SubnetId: SubnetId}'
```

Un ejemplo de salida sería el siguiente.

```
[
    {
        "AvailabilityZone": "us-west-2d",
        "SubnetId": "subnet-Example5"
    }
]
```

c. Anote cada nodo con la `ENIConfig` que creó para el ID de subred y la zona de disponibilidad. Solo puede anotar un nodo con una `ENIConfig`, aunque se pueden anotar varios nodos con la misma `ENIConfig`. Reemplace los *`example values`* por los de su propiedad.

```
kubectl annotate node ip-192-168-0-126.us-west-2.compute.internal
 k8s.amazonaws.com/eniConfig=EniConfigName1
kubectl annotate node ip-192-168-0-92.us-west-2.compute.internal
 k8s.amazonaws.com/eniConfig=EniConfigName2
```

4. Si tenía nodos en un clúster de producción con ejecución Pods antes de cambiar para utilizar la característica de redes personalizadas, realice las siguientes tareas:

a. Asegúrese de tener nodos disponibles que utilizan la característica de red personalizada.

b. Acordone y drene los nodos para apagar con gracia el Pods. Para obtener más información, consulte Drene un nodo de manera segura en la documentación de Kubernetes.

c. Termine los nodos. Si los nodos se encuentran en un grupo de nodos administrado existente, puede eliminar el grupo de nodos. Copie el comando que sigue en su dispositivo.

Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado:

- Reemplace *my-cluster* por el nombre del clúster.
- Reemplace *my-nodegroup* por el nombre de su grupo de nodos.

```
aws eks delete-nodegroup --cluster-name my-cluster --nodegroup-name my-nodegroup
```

Solo los nodos nuevos que se registran con la etiqueta `k8s.amazonaws.com/eniConfig` utilizan la característica de redes personalizadas.

5. Confirme que los Pods estén asignados en una dirección IP de un bloque CIDR asociado a una de las subredes que creó en un paso anterior.

```
kubectl get pods -A -o wide
```

Un ejemplo de salida sería el siguiente.

```
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE     IP
           NODE                                            NOMINATED NODE   READINESS
 GATES
kube-system   aws-node-2rkn4             1/1     Running   0          7m19s
 192.168.0.92    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   aws-node-k96wp             1/1     Running   0          7m15s
 192.168.0.126   ip-192-168-0-126.us-west-2.compute.internal   <none>
 <none>
kube-system   coredns-657694c6f4-smcgr   1/1     Running   0          56m
 192.168.1.23    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   coredns-657694c6f4-stwv9   1/1     Running   0          56m
 192.168.1.28    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
kube-system   kube-proxy-jgshq           1/1     Running   0          7m19s
 192.168.0.92    ip-192-168-0-92.us-west-2.compute.internal    <none>
 <none>
```

```
kube-system   kube-proxy-wx9vk           1/1     Running   0          7m15s
 192.168.0.126   ip-192-168-0-126.us-west-2.compute.internal   <none>
 <none>
```

Puede ver que a los `coredns` Pods se les asignan direcciones IP desde el bloque CIDR `192.168.1.0` que ha agregado a la VPC. Sin redes personalizadas, se les habrían asignado direcciones desde el bloque CIDR `192.168.0.0`, porque era el único bloque CIDR asociado originalmente a la VPC.

Si la `spec` de un Pod's contiene `hostNetwork=true`, se le asigna la dirección IP principal del nodo. No se le asigna una dirección de las subredes que ha agregado. De forma predeterminada, este valor se establece en `false`. Este valor se establece en `true` para los Pods `kube-proxy` y Amazon VPC CNI plugin for Kubernetes (`aws-node`) que se ejecutan en el clúster. Es por eso que al `kube-proxy` y a los pods Pods de `aws-node` del complemento no se les asignan direcciones `192.168.1.`*x* de la salida anterior. Para obtener más información acerca de la configuración de `hostNetwork` de Pod's, consulte Núcleo de PodSpec v1 en la referencia de API de Kubernetes.

Paso 5: eliminar recursos del tutorial

Una vez completado el tutorial, le recomendamos que elimine los recursos que creó. Puede ajustar los pasos para habilitar las redes personalizadas para un clúster de producción.

Para eliminar los recursos del tutorial

1. Si el grupo de nodos que creó fue solo para pruebas, bórrelo.

   ```
   aws eks delete-nodegroup --cluster-name $cluster_name --nodegroup-name my-nodegroup
   ```

   Incluso después de que la salida de la AWS CLI indica que el clúster se ha eliminado, es posible que el proceso de eliminación no esté completo. El proceso de eliminación tarda unos minutos. Confirme que se ha completado al ejecutar el siguiente comando.

   ```
   aws eks describe-nodegroup --cluster-name $cluster_name --nodegroup-name my-
   nodegroup --query nodegroup.status --output text
   ```

   No continúe hasta que la salida devuelta sea similar a la siguiente.

```
An error occurred (ResourceNotFoundException) when calling the DescribeNodegroup
 operation: No node group found for name: my-nodegroup.
```

2. Si el grupo de nodos que creó fue solo para pruebas, elimine el rol de IAM de nodo.

   a. Separe la política del rol.

   ```
   aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSNodeRole --
   policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
   aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSNodeRole --
   policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly
   aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSNodeRole --
   policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy
   ```

   b. Elimine el rol.

   ```
   aws iam delete-role --role-name myCustomNetworkingAmazonEKSNodeRole
   ```

3. Eliminar el clúster.

   ```
   aws eks delete-cluster --name $cluster_name
   ```

   Confirme que el clúster se elimine con el siguiente comando.

   ```
   aws eks describe-cluster --name $cluster_name --query cluster.status --output text
   ```

   Cuando se devuelve un resultado similar al siguiente, el clúster se eliminará correctamente.

   ```
   An error occurred (ResourceNotFoundException) when calling the DescribeCluster
    operation: No cluster found for name: my-cluster.
   ```

4. Elimine el rol de IAM del clúster.

   a. Separe la política del rol.

   ```
   aws iam detach-role-policy --role-name myCustomNetworkingAmazonEKSClusterRole
    --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
   ```

   b. Elimine el rol.

```
aws iam delete-role --role-name myCustomNetworkingAmazonEKSClusterRole
```

5. Elimine las subredes que creó en un paso anterior.

```
aws ec2 delete-subnet --subnet-id $new_subnet_id_1
aws ec2 delete-subnet --subnet-id $new_subnet_id_2
```

6. Elimine la VPC que ha creado.

```
aws cloudformation delete-stack --stack-name my-eks-custom-networking-vpc
```

Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2

Cada instancia de Amazon EC2 admite una cantidad máxima de interfaces de red elásticas y una cantidad máxima de direcciones IP que se pueden asignar a cada interfaz de red. Cada nodo requiere una dirección IP para cada interfaz de red. Se pueden asignar todas las demás direcciones IP disponibles a `Pods`. Cada uno `Pod` requiere su propia dirección IP. Como resultado, es posible que tenga nodos que tengan recursos informáticos y de memoria disponibles, pero que no puedan acomodar `Pods` adicionales porque el nodo se quedó sin direcciones IP para asignar a `Pods`.

En este tema, aprenderá a aumentar significativamente la cantidad de direcciones IP que los nodos pueden asignar a `Pods` mediante la asignación de prefijos de IP, en lugar de asignar direcciones IP secundarias individuales a sus nodos. Cada prefijo incluye varias direcciones IP. Si no configura su clúster para la asignación de prefijos IP, debe realizar más llamadas a la interfaz de programación de aplicaciones (API) de Amazon EC2 para configurar las interfaces de red y las direcciones IP necesarias para la conectividad Pod. A medida que los clústeres crecen a tamaños más grandes, la frecuencia de estas llamadas a la API puede generar tiempos de lanzamiento de instancias y Pod más prolongados. Esto causa retrasos en el escalado para satisfacer la demanda de cargas de trabajo grandes y exigentes, y agrega costos y gastos generales de administración, ya que necesita aprovisionar clústeres y VPC adicionales para cumplir con los requisitos de escalado. Para obtener más información, consulte Umbrales de escalabilidad de Kubernetes en GitHub.

Consideraciones

- Cada tipo de instancia de Amazon EC2 admite una cantidad máxima de Pods. Si el grupo de nodos administrado consta de varios tipos de instancias, el número de Pods máximo menor de una instancia del clúster se aplica a todos los nodos del clúster.

- De forma predeterminada, el número máximo de `Pods` que puede ejecutar en un nodo es 110, pero puede cambiar ese número. Si cambia el número y tiene un grupo de nodos administrado existente, la siguiente AMI o la actualización de la plantilla de lanzamiento de su grupo de nodos generará nuevos nodos con el valor modificado.
- Al pasar de la asignación de direcciones IP a la asignación de prefijos de IP, le recomendamos que cree nuevos grupos de nodos para aumentar la cantidad de direcciones IP disponibles, en lugar de realizar un reemplazo gradual de los nodos existentes. La ejecución de Pods en un nodo que tiene direcciones IP y prefijos asignados puede generar inconsistencias en la capacidad de direcciones IP anunciada, lo que afecta las futuras cargas de trabajo en el nodo. Para conocer la forma recomendada de realizar la transición, consulte Reemplazar todos los nodos durante la migración del modo de IP secundaria al modo de delegación de prefijo o viceversa en la guía de prácticas recomendadas de Amazon EKS.
- Solo para clústeres con nodos Linux.
  - Una vez que configura el complemento para asignar prefijos a las interfaces de red, no puede degradar su complemento Amazon VPC CNI plugin for Kubernetes a una versión anterior a `1.9.0` (o `1.10.1`) sin eliminar todos los nodos en todos los grupos de nodos en su clúster.
  - Si también utiliza grupos de seguridad para Pods, con `POD_SECURITY_GROUP_ENFORCING_MODE` = `standard` y `AWS_VPC_K8S_CNI_EXTERNALSNAT` =`false`, cuando se Pods comunica con puntos de conexión fuera de su VPC, se utilizan los grupos de seguridad del nodo, en lugar de cualquier grupo de seguridad que haya asignado a su Pods.

    Si también utiliza grupos de seguridad para Pods, con `POD_SECURITY_GROUP_ENFORCING_MODE` =`strict`, cuando su `Pods` se comunica con puntos de conexión fuera de su VPC, se utilizan los grupos de seguridad `Pod's`.

Requisitos previos

- Un clúster existente. Para implementar uno, consulte Creación de un clúster de Amazon EKS.
- Las subredes en las que se encuentran sus nodos de Amazon EKS deben tener suficientes bloques contiguos `/28` (para clústeres `IPv4`) o `/80` (para clústeres `IPv6`) enrutamiento entre dominios sin clases (CIDR). Solo puede tener nodos Linux en un clúster `IPv6`. El uso de prefijos IP puede fallar si las direcciones IP están dispersas por toda la subred CIDR. Le recomendamos lo siguiente:

  - Utilizar una reserva CIDR de subred para que, aunque se siga utilizando alguna dirección IP dentro del rango reservado, una vez publicada, no se reasignen las direcciones IP. Esto garantiza que los prefijos estén disponibles para su asignación sin segmentación.
  - Utilice nuevas subredes que se usen específicamente para ejecutar las cargas de trabajo a las que se asignan los prefijos IP. Tanto las cargas de trabajo Windows como las Linux pueden ejecutarse en la misma subred cuando se asignan prefijos de IP.
- Para asignar prefijos de IP a sus nodos, sus nodos deben estar basados en AWS Nitro. Las instancias que no están basadas en Nitro continúan asignando direcciones IP secundarias individuales, pero tienen una cantidad significativamente menor de direcciones IP para asignar a las instancias Pods que las instancias Nitro-based.
- Solo para clústeres con nodos Linux: si su clúster está configurado para la familia `IPv4`, debe tener instalada la versión `1.9.0` o posterior del complemento Amazon VPC CNI plugin for Kubernetes. Puede comprobar su versión actual con el siguiente comando.

  ```
  kubectl describe daemonset aws-node --namespace kube-system | grep Image | cut -d "/"
   -f 2
  ```

  Si su clúster está configurado para la familia `IPv6`, debe tener instalada la versión `1.10.1` o del complemento. Si la versión de su complemento es anterior a las versiones requeridas, debe actualizarlo. Para obtener más información, consulte las secciones de actualización de Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.
- Solo para clústeres con nodos Windows.
  - El clúster y su versión de la plataforma deben ser iguales o posteriores a las versiones de la siguiente tabla. Para actualizar la versión de su clúster, utilice Actualización de una versión de Kubernetes de clúster de Amazon EKS. Si su clúster no tiene la versión mínima de la plataforma, no puede asignar prefijos IP a sus nodos hasta que Amazon EKS haya actualizado la versión de la plataforma.

Versión de Kubernetes	Versión de la plataforma
`1.27`	`eks.3`
`1.26`	`eks.4`
`1.25`	`eks.5`

Puede verificar su versión actual de Kubernetes y de la plataforma reemplazando *my-cluster* en el siguiente comando con el nombre de su clúster y luego ejecutando el comando modificado: **aws eks describe-cluster --name *my-cluster* --query 'cluster.{"Kubernetes Version": version, "Platform Version": platformVersion}'**.

- Windows habilita la compatibilidad con su clúster. Para obtener más información, consulte Activación de la compatibilidad con Windows para su clúster de Amazon EKS.

A fin de aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2

1. Configure el clúster para asignar prefijos de direcciones IP a los nodos. Complete el procedimiento en la pestaña que coincida con el sistema operativo de su nodo.

   Linux

   1. Habilite el parámetro a fin de asignar prefijos a las interfaces de red para el `DaemonSet` de CNI de Amazon VPC. Cuando implementa un `1.21` clúster posterior, la versión `1.10.1` o posterior del complemento Amazon VPC CNI plugin for Kubernetes se implementa con él. Si ha creado el clúster con la familia `IPv6`, este ajuste se configuró en `true` de forma predeterminada. Si ha creado el clúster con la familia `IPv4`, este ajuste se configuró en `false` de forma predeterminada.

      ```
      kubectl set env daemonset aws-node -n kube-system
       ENABLE_PREFIX_DELEGATION=true
      ```

      Important

      Incluso si la subred tiene direcciones IP disponibles, si la subred no tiene disponible ningún bloque `/28` contiguo, verá el siguiente error en los registros Amazon VPC CNI plugin for Kubernetes.

      ```
      InsufficientCidrBlocks: The specified subnet does not have enough free
       cidr blocks to satisfy the request
      ```

      Esto puede ocurrir debido a la fragmentación de las direcciones IP secundarias existentes distribuidas por una subred. Para resolver este error, cree una nueva subred y lance Pods allí, o utilice una reserva CIDR de subred de Amazon EC2

para reservar espacio dentro de una subred para utilizarla con la asignación de prefijos. Para obtener más información, consulte Reservas de la subred de CIDR en la Guía del usuario de Amazon VPC.

2. Si planea implementar un grupo de nodos administrado sin una plantilla de lanzamiento o con una plantilla de lanzamiento en la que no ha especificado un ID de AMI, y está utilizando una versión del Amazon VPC CNI plugin for Kubernetes igual o posterior a las versiones enumeradas en los requisitos previos, continúe con el siguiente paso. Los grupos de nodos administrados calculan automáticamente el número máximo de Pods.

   Si va a implementar un grupo de nodos autoadministrado o un grupo de nodos administrado con una plantilla de lanzamiento en la que ha especificado un ID de AMI, debe determinar el número máximo de Pods recomendados por Amazon EKS para los nodos. Siga las instrucciones de Número máximo de Pods recomendado por Amazon EKS para cada tipo de instancia de Amazon EC2, agregando **`--cni-prefix-delegation-enabled`** al paso tres. Observe la salida de su uso en un paso posterior.

   > **Important**
   >
   > Los grupos de nodos administrados aplican un número máximo en el valor `maxPods`. Para las instancias con menos de 30 vCPU, el número máximo es 110 y para todas las demás instancias el número máximo es 250. Este número máximo se aplica independientemente de si la delegación de prefijos está habilitada o no.

3. Si utiliza un clúster `1.21` o posterior configurado para `IPv6`, continúe con el siguiente paso.

   Especifique los parámetros en una de las siguientes opciones. Para determinar qué opción es adecuada para usted y qué valor debe proporcionarle, consulte `WARM_PREFIX_TARGET`, `WARM_IP_TARGET` y `MINIMUM_IP_TARGET` en GitHub.

   Puede reemplazar el *`example values`* por un valor mayor a cero.

   - `WARM_PREFIX_TARGET`

     ```
     kubectl set env ds aws-node -n kube-system WARM_PREFIX_TARGET=1
     ```

- `WARM_IP_TARGET` o `MINIMUM_IP_TARGET`: si se establece este valor, sustituye a cualquier valor establecido para `WARM_PREFIX_TARGET`.

```
kubectl set env ds aws-node -n kube-system WARM_IP_TARGET=5
```

```
kubectl set env ds aws-node -n kube-system MINIMUM_IP_TARGET=2
```

4. Cree uno de los siguientes tipos de grupos de nodos con al menos un tipo de instancia Nitro Amazon Linux 2 de Amazon EC2. A fin de obtener una lista de los tipos de instancias de Nitro, consulte Instancias integradas en el sistema Nitro en la Guía del usuario de Amazon EC2. Esta capacidad no es compatible con Windows. En el caso de las opciones que incluyen *110*, reemplácelo por el valor del paso tres (recomendado) o un valor propio.

   - Autoadministrado: implementa el grupo de nodos según las instrucciones en Lanzar nodos autoadministrados de Amazon Linux. Especifique el siguiente texto para el parámetro BootstrapArguments.

     ```
     --use-max-pods false --kubelet-extra-args '--max-pods=110'
     ```

     Si utiliza `eksctl` para crear el grupo de nodos, puede usar el siguiente comando.

     ```
     eksctl create nodegroup --cluster my-cluster --managed=false --max-pods-per-node 110
     ```

   - Administrado: implemente el grupo de nodos mediante una de las siguientes opciones:
     - Sin una plantilla de lanzamiento o con una plantilla de lanzamiento sin un ID de AMI especificado: complete el procedimiento indicado en Creación de un grupo de nodos administrados. Los grupos de nodos administrados calculan automáticamente el valor `max-pods` recomendados por Amazon EKS.
     - Con una plantilla de lanzamiento con un ID de AMI especificado: en la plantilla de lanzamiento, especifique un ID de AMI optimizada para Amazon EKS o una AMI personalizada creada a partir de la AMI optimizada para Amazon EKS y, a continuación, implemente el grupo de nodos mediante una plantilla de lanzamiento y proporcione los siguientes datos de usuario en la plantilla de lanzamiento. Estos datos de usuario pasan los argumentos en el archivo `bootstrap.sh`. Para obtener más información acerca del archivo del proceso de arranque, consulte bootstrap.sh en GitHub.

```
/etc/eks/bootstrap.sh my-cluster \
  --use-max-pods false \
  --kubelet-extra-args '--max-pods=110'
```

Si utiliza `eksctl` para crear el grupo de nodos, puede usar el siguiente comando.

```
eksctl create nodegroup --cluster my-cluster --max-pods-per-node 110
```

Si ha creado una AMI personalizada, pero no a partir de la AMI optimizada de Amazon EKS, debe crear personalmente la configuración.

> **Note**
>
> Si también desea asignar direcciones IP a Pods de una subred diferente a la de la instancia, debe habilitar la capacidad en este paso. Para obtener más información, consulte Redes personalizadas para los pods.

Windows

1. Habilite la asignación de prefijos IP.
   a. Abra el `ConfigMap` de `amazon-vpc-cni` para editar.

   ```
   kubectl edit configmap -n kube-system amazon-vpc-cni -o yaml
   ```

   b. Añada la siguiente línea a la sección `data`:

   ```
     enable-windows-prefix-delegation: "true"
   ```

   c. Guarde el archivo y cierre el editor.
   d. Confirme que la línea se agregó a `ConfigMap`.

   ```
   kubectl get configmap -n kube-system amazon-vpc-cni -o
    "jsonpath={.data.enable-windows-prefix-delegation}"
   ```

Si la salida devuelta no es `true`, es posible que haya habido un error. Intente completar el paso de nuevo.

> **Important**
>
> Incluso si su subred tiene direcciones IP disponibles, si la subred no tiene bloques `/28` contiguos disponibles, verá el siguiente error en los eventos del nodo.
>
> ```
> "failed to allocate a private IP/Prefix address:
>  InsufficientCidrBlocks: The specified subnet does not have enough
>  free cidr blocks to satisfy the request"
> ```
>
> Esto puede ocurrir debido a la fragmentación de las direcciones IP secundarias existentes distribuidas por una subred. Para resolver este error, cree una nueva subred y lance Pods allí, o utilice una reserva CIDR de subred de Amazon EC2 para reservar espacio dentro de una subred para utilizarla con la asignación de prefijos. Para obtener más información, consulte Reservas de la subred de CIDR en la Guía del usuario de Amazon VPC.

2. (Opcional) Especifique una configuración adicional para controlar el comportamiento de escalado previo y dinámico del clúster. Para obtener más información, consulte Opciones de configuración con modo de delegación de prefijo en Windows en GitHub.

   a. Abra el `ConfigMap` de `amazon-vpc-cni` para editar.

   ```
   kubectl edit configmap -n kube-system amazon-vpc-cni -o yaml
   ```

   b. Reemplace el *`example values`* con un valor mayor que cero y agregue las entradas que necesite a la sección `data` de `ConfigMap`. Si establece un valor para `warm-ip-target` o `minimum-ip-target`, el valor anula cualquier valor establecido para `warm-prefix-target`.

   ```
   warm-prefix-target: "1"
   warm-ip-target: "5"
   minimum-ip-target: "2"
   ```

   c. Guarde el archivo y cierre el editor.

3. Cree grupos de nodos Windows con al menos un tipo de instancia Nitro de Amazon EC2. Para obtener una lista de los tipos de instancias Nitro, consulte Instancias creadas en el sistema Nitro en la Guía del usuario de Amazon EC2. De forma predeterminada, la cantidad máxima de Pods que puede implementar en un nodo es 110. Si desea aumentar o disminuir ese número, especifique lo siguiente en los datos de usuario para la configuración de arranque. Reemplace *max-pods-quantity* con su valor máximo de pods.

   ```
   -KubeletExtraArgs '--max-pods=max-pods-quantity'
   ```

   Si está implementando grupos de nodos administrados, esta configuración debe agregarse en la plantilla de lanzamiento. Para obtener más información, consulte Personalización de nodos administrados con plantillas de lanzamiento. Para obtener más información sobre los parámetros de configuración para el script de arranque Windows, consulte Parámetros de configuración del script de arranque.

2. Una vez que se implementan los nodos, consulte los nodos del clúster.

   ```
   kubectl get nodes
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME                                              STATUS     ROLES    AGE   VERSION
   ip-192-168-22-103.region-code.compute.internal    Ready      <none>   19m   v1.XX.X-eks-6b7464
   ip-192-168-97-94.region-code.compute.internal     Ready      <none>   19m   v1.XX.X-eks-6b7464
   ```

3. Describa uno de los nodos para determinar el valor de `max-pods` para el nodo y el número de direcciones IP disponibles. Reemplace *192.168.30.193* con la dirección `IPv4` en el nombre de uno de sus nodos devueltos en la salida anterior.

   ```
   kubectl describe node ip-192-168-30-193.region-code.compute.internal | grep 'pods\|PrivateIPv4Address'
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   pods:                              110
   ```

```
vpc.amazonaws.com/PrivateIPv4Address:  144
```

En el resultado anterior, `110` es el número máximo de Pods que Kubernetes implementará en el nodo, aunque haya *144* direcciones IP disponibles.

Grupos de seguridad de Pods

Los grupos de seguridad de Pods integran los grupos de seguridad de Amazon EC2 con los Pods de Kubernetes. Puede utilizar grupos de seguridad de Amazon EC2 para definir reglas que permitan el tráfico de red entrante y saliente hacia y desde los Pods que implemente en nodos que se ejecutan en muchos tipos de instancias de Amazon EC2 y Fargate. Para obtener una explicación más detallada de esta capacidad, consulte la publicación de blog Presentación de los grupos de seguridad para Pods.

Consideraciones

- Antes de implementar grupos de seguridad para Pods, tenga en cuenta las siguientes limitaciones y condiciones:
- Los grupos de seguridad de Pods no se pueden utilizar con nodos de Windows.
- Los grupos de seguridad para Pods pueden utilizarse en clústeres configurados para la familia `IPv6` que contengan nodos de Amazon EC2 mediante la versión 1.16.0 o posterior del complemento CNI de Amazon VPC. Puede usar grupos de seguridad para Pods con clústeres configurados para la familia `IPv6` que contengan solo nodos de Fargate mediante la versión 1.7.7 o posterior del complemento CNI de Amazon VPC. Para obtener más información, consulte Direcciones `IPv6` de clústeres, Pods y services
- Los grupos de seguridad para Pods son compatibles con la mayoría de las familias de instancias de Amazon EC2 basadas en Nitro, aunque no en todas las generaciones de una familia. Por ejemplo, las generaciones y familia de instancias `m5`, `c5`, `r5`, `m6g`, `c6g` y `r6g` son compatibles. No se admite ningún tipo de instancia de la familia `t`. Para obtener una lista completa de los tipos de instancias compatibles, consulte el archivo limits.go en GitHub. Sus nodos deben ser uno de los tipos de instancias enumerados que tienen `IsTrunkingCompatible: true` en ese archivo.
- Si también utiliza políticas de seguridad del Pod a fin de restringir el acceso a la mutación de Pod, el usuario del `eks:vpc-resource-controller` de Kubernetes debe especificarse en `ClusterRoleBinding` de Kubernetes para el `role` al que se le asigna `psp`. Si utiliza el `psp`, `role` y `ClusterRoleBinding` predeterminados de Amazon EKS, este es el `ClusterRoleBinding` de `eks:podsecuritypolicy:authenticated`. Por ejemplo, agrega el usuario a la sección `subjects:`, tal como se muestra en el siguiente ejemplo:

```
[...]
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: system:authenticated
  - apiGroup: rbac.authorization.k8s.io
    kind: User
    name: eks:vpc-resource-controller
  - kind: ServiceAccount
    name: eks-vpc-resource-controller
```

- Si utiliza redes y grupos de seguridad personalizados para los Pods de forma conjunta, se utiliza el grupo de seguridad especificado por los grupos de seguridad de Pods en lugar del grupo de seguridad especificado en `ENIConfig`.
- Si utiliza la versión `1.10.2` o anterior del complemento CNI de Amazon VPC e incluye la configuración de `terminationGracePeriodSeconds` en la especificación del Pod, el valor de la configuración no puede ser cero.
- Si utiliza la versión `1.10` o anterior del complemento CNI de Amazon VPC o la versión `1.11` con `POD_SECURITY_GROUP_ENFORCING_MODE=strict`, que es la configuración predeterminada, entonces los servicios de Kubernetes de tipo `NodePort` y `LoadBalancer` con destinos de instancia con un conjunto `externalTrafficPolicy` establecido en `Local` no son compatibles con los Pods a los que asigna grupos de seguridad. Para obtener más información sobre el uso de un equilibrador de carga con destinos de instancia, consulte Equilibrio de carga de red en Amazon EKS
- Si utiliza la versión `1.10` o anterior del complemento o versión CNI de Amazon VPC `1.11` con `POD_SECURITY_GROUP_ENFORCING_MODE=strict`, que es la configuración predeterminada, el NAT de origen está deshabilitado para el tráfico saliente de Pods con grupos de seguridad asignados a fin de que se apliquen las reglas de grupo de seguridad salientes. Para acceder a Internet, los Pods con grupos de seguridad asignados deben lanzarse en nodos que se implementen en una subred privada configurada con una instancia o puerta de enlace NAT. Los Pods con los grupos de seguridad asignados implementados en subredes públicas no pueden acceder a la internet.

  Si utiliza la versión `1.11` o posterior del plugin con `POD_SECURITY_GROUP_ENFORCING_MODE=standard`, entonces el tráfico de Pod destinado para salir de la VPC se traduce a la dirección IP de la interfaz de red principal de la instancia. Para

este tráfico, se utilizan las reglas de los grupos de seguridad de la interfaz de red principal, en lugar de las reglas de los grupos de seguridad de los Pod's.

- Para utilizar la política de red de Calico con Pods que tienen grupos de seguridad asociados, debe utilizar la versión `1.11.0` o posterior del complemento CNI de Amazon VPC y establecerla en `POD_SECURITY_GROUP_ENFORCING_MODE=standard`. De otro modo, el flujo de tráfico hacia y desde los Pods con grupos de seguridad asociados no están sujetos al cumplimiento de la política de red de Calico y solo se limitan a la aplicación de grupos de seguridad de Amazon EC2. Para actualizar la versión del CNI de Amazon VPC, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS
- Los Pods que funciona en nodos de Amazon EC2 que utilizan grupos de seguridad en clústeres que utilizan Nodelocal DNSCache solo se admiten con la versión `1.11.0` o posterior del complemento CNI de Amazon VPC y con `POD_SECURITY_GROUP_ENFORCING_MODE=standard`. Para actualizar la versión del complemento CNI de Amazon VPC, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS
- Los grupos de seguridad de Pods pueden generar un aumento en la latencia de inicio de un Pod en el caso de los Pods con alta pérdida. Esto se debe a la limitación de velocidad en el controlador de recursos.

Configuración del Amazon VPC CNI plugin for Kubernetes para grupos de seguridad de los Pods

Para implementar grupos de seguridad de Pods

Si solo utiliza grupos de seguridad para Pods de Fargate y no cuenta con nodos de Amazon EC2 en el clúster, vaya al paso Implemente un aplicación de ejemplo .

1. Verifique su versión del Amazon VPC CNI plugin for Kubernetes actual con el siguiente comando:

   ```
   kubectl describe daemonset aws-node --namespace kube-system | grep amazon-k8s-cni:
    | cut -d : -f 3
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v1.7.6
   ```

Si su versión del Amazon VPC CNI plugin for Kubernetes es anterior a la `1.7.7`, actualice el complemento a la versión `1.7.7` o posterior. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS

2. Agregue la política de IAM administrada `AmazonEKSVPCResourceController` al rol de clúster asociado a su clúster de Amazon EKS. La política permite que el rol administre las interfaces de red, sus direcciones IP privadas y su vinculación y desvinculación desde y hacia las instancias de red.

   a. Recupere el nombre del rol de IAM de su clúster y guárdelo en una variable. Reemplace *`my-cluster`* por el nombre del clúster.

   ```
   cluster_role=$(aws eks describe-cluster --name my-cluster --query
    cluster.roleArn --output text | cut -d / -f 2)
   ```

   b. Asocie la política de al rol.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   AmazonEKSVPCResourceController --role-name $cluster_role
   ```

3. Habilite el complemento CNI de Amazon VPC para administrar las interfaces de red de los Pods al establecer la variable `ENABLE_POD_ENI` en `true` en el `aws-node` DaemonSet. Una vez que esta configuración se establece en `true`, para cada nodo del clúster, el complemento crea un recurso personalizado de cninode. El controlador de recursos de VPC crea y adjunta una interfaz de red especial denominada interfaz de red troncal con la descripción `aws-k8s-trunk-eni`.

   ```
   kubectl set env daemonset aws-node -n kube-system ENABLE_POD_ENI=true
   ```

   > **Note**
   >
   > La interfaz de red troncal se incluye en el número máximo de interfaces de red que admite el tipo de instancia. A fin de obtener una lista del número máximo de interfaces de red que admite cada tipo de instancia, consulte Direcciones IP por interfaz de red por tipo de instancia en la Guía del usuario de Amazon EC2. Si su nodo ya cuenta con el número máximo de interfaces de red estándar adjuntas, el controlador de recursos de VPC reservará un espacio. Tendrá que reducir verticalmente los Pods en ejecución lo suficiente para que el controlador desconecte y elimine una interfaz de red estándar, cree la interfaz de red troncal y la adjunte a la instancia.

4. Si desea ver cuál de sus nodos tiene un recurso personalizado `CNINode` establecido, utilice el siguiente comando. Si se devuelve `No resources found`, espere varios segundos e inténtelo de nuevo. El paso anterior requiere reiniciar los Pods del Amazon VPC CNI plugin for Kubernetes, lo cual tarda varios segundos.

```
$ kubectl get cninode -A
     NAME FEATURES
     ip-192-168-64-141.us-west-2.compute.internal
 [{"name":"SecurityGroupsForPods"}]
     ip-192-168-7-203.us-west-2.compute.internal [{"name":"SecurityGroupsForPods"}]
```

Si utiliza versiones de CNI de VPC anteriores a `1.15`, se utilizaron etiquetas de nodo en lugar del recurso personalizado `CNINode`. Si desea ver cuál de sus nodos tienen la etiqueta de nodo `aws-k8s-trunk-eni` establecida en `true`, utilice el siguiente comando. Si se devuelve `No resources found`, espere varios segundos e inténtelo de nuevo. El paso anterior requiere reiniciar los Pods del Amazon VPC CNI plugin for Kubernetes, lo cual tarda varios segundos.

```
kubectl get nodes -o wide -l vpc.amazonaws.com/has-trunk-attached=true
-
```

Una vez que se crea la interfaz de red troncal, se pueden asignar direcciones IP secundarias a los Pods desde las interfaces de red troncales o estándar. La interfaz troncal se elimina de forma automática si se elimina el nodo.

Cuando implementa un grupo de seguridad para un Pod en un paso posterior, el controlador de recursos de VPC crea una interfaz de red especial denominada interfaz de red de ramificación con una descripción de `aws-k8s-branch-eni` y les asocia los grupos de seguridad. Se crean las interfaces de red de ramificación además de las interfaces de red estándar y troncal adjuntas al nodo.

Si utiliza sondeos de estado o preparación, también necesita desactivar el demux temprano de TCP, de modo que `kubelet` pueda conectarse a los Pods en las interfaces de red de ramificación a través de TCP. Para desactivar el demux temprano de TCP, ejecute el siguiente comando:

```
kubectl patch daemonset aws-node -n kube-system \
```

```
  -p '{"spec": {"template": {"spec": {"initContainers": [{"env":
[{"name":"DISABLE_TCP_EARLY_DEMUX","value":"true"}],"name":"aws-vpc-cni-
init"}]}}}}'
```

> **Note**
>
> Si utiliza `1.11.0` o posterior del complemento Amazon VPC CNI plugin for Kubernetes y lo establece en `POD_SECURITY_GROUP_ENFORCING_MODE=standard`, como se describe en el siguiente paso, no es necesario ejecutar el comando anterior.

5. Si su clúster usa `NodeLocal DNSCache` o desea usar la política de red de Calico con los Pods que tienen sus propios grupos de seguridad, o si tiene servicios de Kubernetes de tipo `NodePort` y `LoadBalancer` mediante los destinos de instancia con una `externalTrafficPolicy` establecida en `Local` para Pods a los que desea asignar grupos de seguridad, debe usar la versión `1.11.0` o posterior del complemento Amazon VPC CNI plugin for Kubernetes y habilitar la siguiente configuración:

```
kubectl set env daemonset aws-node -n kube-system
 POD_SECURITY_GROUP_ENFORCING_MODE=standard
```

> **Important**
>
> - Las reglas del grupo de seguridad del Pod no se aplican al tráfico entre Pods o entre Pods y services, como `kubelet` o `nodeLocalDNS`, que se encuentran en el mismo nodo. Los pods que utilizan diferentes grupos de seguridad en el mismo nodo no pueden comunicarse porque están configurados en diferentes subredes y el enrutamiento está deshabilitado entre estas subredes.
> - Tráfico saliente de Pods a las direcciones fuera de la VPC es la dirección de red traducida a la dirección IP de la interfaz de red principal de la instancia (a menos que también haya configurado `AWS_VPC_K8S_CNI_EXTERNALSNAT=true`). Para este tráfico, se utilizan las reglas de los grupos de seguridad de la interfaz de red principal, en lugar de las reglas de los grupos de seguridad de los Pod's.
> - Para que esta configuración se aplique a los Pods existentes, debe reiniciar los Pods o los nodos en que se están ejecutando los Pods.

Implemente un aplicación de ejemplo

Para utilizar grupos de seguridad para Pods, debe tener un grupo de seguridad existente e Implementar un Amazon EKS `SecurityGroupPolicy` al clúster, tal y como se describe en el siguiente procedimiento. En los siguientes pasos se muestra cómo utilizar la política de grupo de seguridad para un Pod. A menos que se indique lo contrario, complete todos los pasos del mismo terminal ya que en los siguientes pasos se utilizan variables que no persisten en los terminales.

Para implementar un ejemplo de Pod con un grupo de seguridad

1. Cree un espacio de nombres de Kubernetes en el que implementar los recursos. Puede reemplazar *my-namespace* por el nombre del espacio de nombres que desee usar.

```
kubectl create namespace my-namespace
```

2. Implemente una `SecurityGroupPolicy` de Amazon EKS en su clúster.

   a. Copie los siguientes contenidos en su dispositivo. Puede reemplazar *podSelector* por **serviceAccountSelector** si prefiere seleccionar Pods en función de las etiquetas de cuenta de servicio. Debe especificar un selector o el otro. Un `podSelector` vacío (ejemplo: `podSelector: {}`) selecciona todos los Pods del espacio de nombres. Puede cambiar *my-role* por el nombre de su rol. Un `serviceAccountSelector` vacío selecciona todas las cuentas de servicio del espacio de nombres. Puede reemplazar *my-security-group-policy* por un nombre para su `SecurityGroupPolicy` y *my-namespace* por el espacio de nombres en el que desea crear la `SecurityGroupPolicy`.

   Debe reemplazar *my_pod_security_group_id* por el ID de un grupo de seguridad existente. Si no dispone de un grupo de seguridad existente, debe crear uno. Para obtener más información, consulte Grupos de seguridad de Amazon EC2 para instancias de Linux en la Guía del usuario de Amazon EC2. Puede especificar de uno a cinco ID de grupo de seguridad. Si especifica más de un ID, la combinación de todas las reglas de todos los grupos de seguridad será efectiva para los Pods seleccionados.

```
cat >my-security-group-policy.yaml <<EOF
apiVersion: vpcresources.k8s.aws/v1beta1
kind: SecurityGroupPolicy
metadata:
  name: my-security-group-policy
  namespace: my-namespace
spec:
```

```
  podSelector:
    matchLabels:
      role: my-role
  securityGroups:
    groupIds:
      - my_pod_security_group_id
EOF
```

Important

El grupo o grupos de seguridad que especifique para sus Pod debe cumplir los siguientes criterios:

- Deben existir. Si no existen, entonces, cuando implementa un Pod que coincida con el selector, el Pod permanece atascado en el proceso de creación. Si describe el Pod, verá un mensaje de error similar al siguiente: `An error occurred (InvalidSecurityGroupID.NotFound) when calling the CreateNetworkInterface operation: The securityGroup ID 'sg-05b1d815d1EXAMPLE' does not exist.`
- Deben permitir la comunicación entrante desde el grupo de seguridad de clúster aplicado a sus nodos (para `kubelet`) a través de los puertos para los que haya configurado sondeos.
- Deben permitir la comunicación saliente a través de los puertos `TCP` y `UDP` 53 a un grupo de seguridad asignado a los Pods (o los nodos en los que los Pods se ejecutan) que ejecutan CoreDNS. El grupo de seguridad de sus Pods de CoreDNS debe permitir el tráfico entrante del puerto `TCP` y `UDP` 53 del grupo de seguridad que especifique.
- Deben tener las reglas entrantes y salientes necesarias para comunicarse con otros Pods con los que deben comunicarse.
- Deben tener reglas que permitan a los Pods comunicarse con el plano de control de Kubernetes si utilizan el grupo de seguridad con Fargate. La forma más sencilla de hacerlo es especificar el grupo de seguridad de clúster como uno de los grupos de seguridad.

Las políticas de grupos de seguridad solo se aplican a los nuevos Pods programados. No afectan a los Pods en ejecución.

b. Implemente la política.

```
kubectl apply -f my-security-group-policy.yaml
```

3. Implemente una aplicación de muestra con una etiqueta que coincida con el valor *my-role* para *podSelector* que especificó en un paso anterior.

   a. Copie los siguientes contenidos en su dispositivo. Reemplace los *valores de ejemplo* por los suyos y, a continuación, ejecute el comando modificado. Si reemplaza *my-role*, asegúrese de que sea igual al valor que especificó para el selector en un paso anterior.

```
cat >sample-application.yaml <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  namespace: my-namespace
  labels:
    app: my-app
spec:
  replicas: 4
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
        role: my-role
    spec:
      terminationGracePeriodSeconds: 120
      containers:
      - name: nginx
        image: public.ecr.aws/nginx/nginx:1.23
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: my-app
  namespace: my-namespace
```

```
  labels:
    app: my-app
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
EOF
```

b. Implemente la aplicación con el siguiente comando. Cuando implementa la aplicación, el Amazon VPC CNI plugin for Kubernetes coincide con la etiqueta de `role` y los grupos de seguridad que especificó en el paso anterior se aplican al Pod.

```
kubectl apply -f sample-application.yaml
```

4. Vea los Pods implementados con la aplicación de muestra. En el resto de este tema, se hace referencia a este terminal como `TerminalA`.

```
kubectl get pods -n my-namespace -o wide
```

Un ejemplo de salida sería el siguiente.

```
NAME                             READY   STATUS    RESTARTS   AGE     IP
       NODE                                                NOMINATED NODE   READINESS
 GATES
my-deployment-5df6f7687b-4fbjm   1/1     Running   0          7m51s   192.168.53.48
    ip-192-168-33-28.region-code.compute.internal   <none>           <none>
my-deployment-5df6f7687b-j9fl4   1/1     Running   0          7m51s
  192.168.70.145   ip-192-168-92-33.region-code.compute.internal   <none>
 <none>
my-deployment-5df6f7687b-rjxcz   1/1     Running   0          7m51s
  192.168.73.207   ip-192-168-92-33.region-code.compute.internal   <none>
 <none>
my-deployment-5df6f7687b-zmb42   1/1     Running   0          7m51s   192.168.63.27
    ip-192-168-33-28.region-code.compute.internal   <none>           <none>
```

> **Note**
>
> - Si hay Pods atascados en estado `Waiting`, ejecute `kubectl describe pod` ***my-deployment-xxxxxxxxxx-xxxxx*** `-n` *my-namespace*. Si ve `Insufficient permissions: Unable to create Elastic Network Interface.`, confirme que agregó la política de IAM al rol de clúster de IAM en un paso anterior.
> - Si algún Pods se encuentra atascado en estado `Pending`, confirme que el tipo de instancia del nodo aparece en `limits.go` y que el producto del número máximo de interfaces de red de ramificación admitidas por el tipo de instancia multiplicado por el número de nodos del grupo de nodos aún no se ha alcanzado. Por ejemplo, una instancia `m5.large` admite nueve interfaces de red de ramificación. Si el grupo de nodos tiene cinco nodos, se puede crear un máximo de 45 interfaces de red de ramificación para el grupo de nodos. El Pod 46 que intente implementar se establecerá en el estado `Pending` hasta que se elimine otro Pod que tenga grupos de seguridad asociados.

Si ejecuta `kubectl describe pod` ***my-deployment-xxxxxxxxxx-xxxxx*** **`-n`** ***my-namespace*** y ve un mensaje similar al siguiente mensaje, puede ignorarlo de forma segura. Este mensaje puede aparecer cuando el Amazon VPC CNI plugin for Kubernetes intenta configurar las redes de host y falla mientras se crea la interfaz de red. El complemento registra este evento hasta que se crea la interfaz de red.

```
Failed to create Pod sandbox: rpc error: code = Unknown desc = failed to set up
 sandbox container
 "e24268322e55c8185721f52df6493684f6c2c3bf4fd59c9c121fd4cdc894579f" network for Pod
 "my-deployment-5df6f7687b-4fbjm": networkPlugin
cni failed to set up Pod "my-deployment-5df6f7687b-4fbjm-c89wx_my-namespace"
 network: add cmd: failed to assign an IP address to container
```

No puede exceder el número máximo de Pods que se pueden ejecutar en el tipo de instancia. Para obtener una lista del número máximo de Pods que puede ejecutar en cada tipo de instancia, consulte eni-max-pods.txt en GitHub. Cuando elimina un Pod que tiene grupos de seguridad asociados o elimina el nodo en el que se ejecuta el Pod, el controlador de recursos de VPC elimina la interfaz de red de ramificación. Si elimina un clúster con Pods mediante Pods para grupos de seguridad, el controlador no elimina las interfaces de red de ramificación, por lo

que deberá eliminarlas por su cuenta. A fin de obtener más información sobre las interfaces de red, consulte Eliminar una interfaz de red en la Guía del usuario de Amazon EC2.

5. En un terminal separado, se inserta en uno de los Pods. En el resto de este tema, se hace referencia a este terminal como `TerminalB`. Reemplace *`5df6f7687b-4fbjm`* con el ID de uno de los Pods devueltos en la salida del paso anterior.

```
kubectl exec -it -n my-namespace my-deployment-5df6f7687b-4fbjm -- /bin/bash
```

6. Desde el intérprete de comandos de `TerminalB`, confirme que la aplicación de ejemplo funciona.

```
curl my-app
```

Un ejemplo de salida sería el siguiente.

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
[...]
```

Recibió el resultado porque todos los Pods que se ejecutan en la aplicación están asociados al grupo de seguridad que creó. Ese grupo contiene una regla que permite todo el tráfico entre todos los Pods al que está asociado el grupo de seguridad. Se permite el tráfico DNS saliente de ese grupo de seguridad al grupo de seguridad del clúster, lo que está asociado a los nodos. Los nodos ejecutan los Pods CoreDNS, en los que sus Pods realizó una búsqueda de nombre.

7. Desde `TerminalA`, elimine las reglas del grupo de seguridad que permita la comunicación DNS al grupo de seguridad del clúster del grupo de seguridad. Si no agregó las reglas DNS al grupo de seguridad del clúster en un paso anterior, sustituya *`$my_cluster_security_group_id`* con el ID del grupo de seguridad en el que creó las reglas.

```
aws ec2 revoke-security-group-ingress --group-id $my_cluster_security_group_id --
security-group-rule-ids $my_tcp_rule_id
aws ec2 revoke-security-group-ingress --group-id $my_cluster_security_group_id --
security-group-rule-ids $my_udp_rule_id
```

8. Desde `TerminalB`, intente acceder de nuevo a la aplicación.

```
curl my-app
```

Un ejemplo de salida sería el siguiente.

```
curl: (6) Could not resolve host: my-app
```

El intento falla porque el Pod ya no puede acceder a los Pods CoreDNS, que tienen asociado el grupo de seguridad del clúster. El grupo de seguridad del clúster ya no tiene las reglas del grupo de seguridad que permiten la comunicación DNS desde el grupo de seguridad asociado a su Pod.

Si intenta acceder a la aplicación utilizando las direcciones IP devueltas para uno de los Pods en un paso anterior, sigue recibiendo una respuesta dado que todos los puertos están permitidos entre Pods que tienen el grupo de seguridad asociado a ellos y no es necesaria una búsqueda de nombres.

9. Una vez que haya terminado de experimentar, puede eliminar la política de grupo de seguridad, la aplicación y el grupo de seguridad de ejemplo que creó. Ejecute los siguientes comandos desde la `TerminalA`.

```
kubectl delete namespace my-namespace
aws ec2 revoke-security-group-ingress --group-id $my_pod_security_group_id --
security-group-rule-ids $my_inbound_self_rule_id
wait
sleep 45s
aws ec2 delete-security-group --group-id $my_pod_security_group_id
```

Varias interfaces de red para Pods

Multus CNI es un complemento de interfaz de red de contenedor (CNI) para Amazon EKS que permite adjuntar varias interfaces de red a un Pod. Para obtener más información, consulte la documentación de Multus-CNI en GitHub.

En Amazon EKS, cada Pod tiene una interfaz de red asignada por el complemento CNI de Amazon VPC. Con Multus, puede crear un Pod de varios alojamientos que tenga varias interfaces. Esto se logra mediante Multus al actuar como un “meta-complemento”; un complemento CNI que puede llamar a varios otros complementos CNI. El soporte de AWS para Multus viene configurado con el complemento CNI de Amazon VPC como complemento delegado predeterminado.

Consideraciones

- Amazon EKS no creará ni publicará complementos CNI de virtualización de E/S de raíz única (SR-IOV) y del kit de desarrollo del plano de datos (DPDK). Sin embargo, puede lograr la aceleración de paquetes al conectarse directamente a Amazon EC2 Elastic Network Adapters (ENA) a través del dispositivo host administrado de Multus y los complementos `ipvlan`.
- Amazon EKS es compatible con Multus, que proporciona un proceso genérico que permite encadenar fácilmente complementos CNI adicionales. Se admite Multus y el proceso de encadenamiento, pero AWS no proporcionará soporte para todos los complementos CNI compatibles que se pueden encadenar, o problemas que puedan surgir en esos complementos CNI que no están relacionados con la configuración de encadenamiento.
- Amazon EKS ofrece soporte y gestión del ciclo de vida para el complemento de Multus, pero no se hace responsable de ninguna dirección IP ni de la gestión adicional asociada con las interfaces de red adicionales. La dirección IP y la gestión de la interfaz de red predeterminada que utiliza el complemento CNI de Amazon VPC permanece sin cambios.
- Solo se admite oficialmente el complemento CNI de Amazon VPC como complemento delegado predeterminado. Debe modificar el manifiesto de instalación de Multus publicado para volver a configurar el complemento delegado predeterminado en un CNI alternativo si decide no utilizar el complemento CNI de Amazon VPC para redes principales.
- Multus solo se admite cuando se utiliza el CNI de Amazon VPC como CNI principal. No admitimos el CNI de Amazon VPC cuando se utiliza para interfaces de orden superior, secundarias o de otro tipo.
- Para evitar que el complemento CNI de Amazon VPC intente administrar interfaces de red adicionales asignadas a los Pods, agregue la etiqueta siguiente a la interfaz de red:

  clave: `node.k8s.amazonaws.com/no_manage`

  valor: `true`
- Multus es compatible con las políticas de red, pero la política tiene que ser enriquecida para incluir puertos y direcciones IP que pueden formar parte de interfaces de red adicionales adjuntas a los Pods.

Para ver la explicación de la implementación, consulte la Guía de configuración de Multus en GitHub.

Complementos CNI compatibles alternativos

Amazon VPC CNI plugin for Kubernetes es el único complemento de CNI compatible con Amazon EKS. Amazon EKS ejecuta Kubernetes de forma ascendente, por lo que puede instalar complementos de CNI alternativos compatibles en los nodos de Amazon EC2 del clúster. Si tiene nodos de Fargate en el clúster, Amazon VPC CNI plugin for Kubernetes ya está en sus nodos de Fargate. Es el único complemento de CNI que puede usar con los nodos de Fargate. Se produce un error cuando intenta instalar un complemento de CNI alternativo en los nodos de Fargate.

Si planea usar un complemento de CNI alternativo en nodos de Amazon EC2, le recomendamos obtener compatibilidad comercial para el complemento o pedir al experto en plantilla que solucione los problemas y aporte correcciones al proyecto de complemento de CNI.

Amazon EKS mantiene relaciones con una red de socios que ofrecen soporte para complementos CNI compatibles alternativos. Consulte la siguiente documentación de socios para obtener información detallada sobre las versiones, las calificaciones y pruebas realizadas.

Socio	Producto	Documentación
Tigera	Calico	Instrucciones de instalación
Isovalent	Cilium	Instrucciones de instalación
Juniper	Redes en contrail nativas en la nube (CN2)	Instrucciones de instalación
VMware	Antrea	Instrucciones de instalación

Amazon EKS tiene como objetivo darle una amplia selección de opciones para cubrir todos los casos de uso.

Plugins de políticas de red compatibles alternativos

Calico es una solución ampliamente adoptada para seguridad y redes de contenedores. El uso de Calico en EKS proporciona una implementación de la política de red plenamente compatible para sus clústeres de EKS. Además, puede optar por utilizar la red de Calico, que conserva las direcciones IP de la VPC subyacente. Calico Cloud mejora las características deCalico Open Source con capacidades avanzadas de seguridad y observabilidad.

¿Qué es el AWS Load Balancer Controller?

El AWS Load Balancer Controller administra los equilibradores de carga elástica de AWS para un clúster de Kubernetes. Puede usar el controlador para exponer las aplicaciones del clúster a Internet. El controlador aprovisiona los equilibradores de carga de AWS que apuntan a los recursos de Service o Ingress del clúster. En otras palabras, el controlador crea una única dirección IP o nombre de DNS que apunta a varios pods del clúster.

El controlador vigila los recursos para Kubernetes Ingress o Service. En respuesta, crea los recursos adecuados de Elastic Load Balancing de AWS. Es posible configurar el comportamiento específico de los equilibradores de carga mediante la aplicación de anotaciones a los recursos de Kubernetes. Por ejemplo, puede adjuntar grupos de seguridad de AWS a los equilibradores de carga mediante anotaciones.

El controlador aprovisiona los siguientes recursos:

Kubernetes `Ingress`

> El LBC crea un equilibrador de carga de aplicación (application load balancer, ALB) de AWS cuando se crea un Kubernetes `Ingress`. Revise las anotaciones que puede aplicar a un recurso de Ingress.

Servicio de Kubernetes del tipo `LoadBalancer`

> El LBC crea un equilibrador de carga de red (network load balancer, NLB) de AWScuando se crea un servicio de Kubernetes del tipo `LoadBalancer`. Revise las anotaciones que puede aplicar a un recurso de Service.
>
> En el pasado, se utilizaba el equilibrador de carga de red de Kubernetes para destinos de la instancia, pero se usaba el LBC para destinos de IP. Con la versión AWS Load Balancer Controller `2.3.0` o posterior, puede crear NLB con cualquiera de los tipos de destino. Para obtener más información acerca de los tipos de destinos del NLB, consulte Tipo de destino en la Guía del usuario para Network Load Balancers.

El controlador es un proyecto de código abierto administrado en GitHub.

Antes de implementar el controlador, recomendamos que consulte los requisitos previos y las consideraciones en Equilibrio de carga de aplicaciones en Amazon EKS y Equilibrio de carga de red en Amazon EKS. En esos temas, implementará una aplicación de muestra que incluye un equilibrador de carga de AWS.

Instalación del controlador

- Aprenda cómo the section called “Instalación con Helm”. Utilice este procedimiento si es la primera vez que utiliza Amazon EKS. Este procedimiento utiliza Helm, un administrador de paquetes para Kubernetes y `eksctl` para simplificar la instalación del LBC.
- Otra opción: the section called “Instalación con manifiestos”. Este procedimiento es adecuado para configuraciones avanzadas de clústeres. Esto incluye los clústeres con acceso de red restringido a los registros de contenedores públicos.

Migración desde versiones de controlador obsoletas

- Si tiene versiones obsoletas del AWS Load Balancer Controller instaladas, obtenga información sobre cómo the section called “Migración desde un controlador obsoleto”.
- Las versiones obsoletas no se pueden actualizar. Deben eliminarse y se debe instalar una versión actual del AWS Load Balancer Controller.
- Las versiones obsoleta incluyen lo siguiente:
  - AWSControlador de Ingress del ALB para Kubernetes («controlador de Ingress»), un predecesor del AWS Load Balancer Controller.
  - Cualquier versión `0.1.`*x* del AWS Load Balancer Controller

Proveedor de nube heredado

Kubernetes incluye un proveedor de nube heredado para AWS. El proveedor de nube heredado es capaz de aprovisionar equilibradores de carga de AWS, similares al AWS Load Balancer Controller. El proveedor de nube heredado crea equilibradores de carga clásicos. Si no instala el AWS Load Balancer Controller, Kubernetes utilizará de forma predeterminada el proveedor de nube heredado. Debe instalar el AWS Load Balancer Controller y evitar utilizar el proveedor de nube heredado.

> **Important**
>
> En las versiones 2.5 y posteriores, el AWS Load Balancer Controller se convierte en el controlador predeterminado para los recursos del servicio de Kubernetes con el `type: LoadBalancer` y hace un Equilibrador de carga de red (NLB) de AWS para cada servicio. Para ello, crea un webhook mutante para los servicios, que establece el campo `spec.loadBalancerClass` para `service.k8s.aws/nlb` para nuevos servicios de `type: LoadBalancer`. Puede desactivar esta característica y volver a utilizar

el proveedor de nube tradicional como controlador predeterminado, estableciendo el valor del gráfico de Helm de `enableServiceMutatorWebhook` a `false`. El clúster no aprovisionará nuevos equilibradores de carga clásicos para sus servicios a menos que desactive esta característica. Los equilibradores de carga clásicos existentes seguirán funcionando.

Instalación de AWS Load Balancer Controller usando Helm

En este tema se describe cómo instalar AWS Load Balancer Controller con Helm, un administrador de paquetes para Kubernetes y `eksctl`. El controlador se instala con las opciones predeterminadas. Para obtener más información sobre el controlador, incluidos los detalles sobre su configuración con anotaciones, consulte la documentación del AWS Load Balancer Controller en GitHub.

En los siguientes pasos, reemplace *`example values`* por sus propios valores.

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesitará para crear y administrar un clúster de Amazon EKS.

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.
- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- Asegúrese de que sus complementos Amazon VPC CNI plugin for Kubernetes, `kube-proxy` y CoreDNS tengan las versiones mínimas enumeradas en los tokens de cuenta de servicio.
- Familiaridad con AWS Elastic Load Balancing. Para obtener más información, consulte la Guía del usuario de Elastic Load Balancing.
- Familiaridad con el servicio de Kubernetes y los recursos de entrada.

- Instalación de Helm de forma local.

Paso 1: crear el rol de IAM usando **eksctl**

> **Note**
>
> Solo necesita crear un rol de IAM para el AWS Load Balancer Controller por cada cuenta de AWS. Compruebe si `AmazonEKSLoadBalancerControllerRole` existe en la Consola de IAM. Si este rol existe, vaya a the section called “Paso 2: instalar el AWS Load Balancer Controller”.

Cree una política de IAM.

1. Descargue una política de IAM para el AWS Load Balancer Controller que le permita realizar llamadas a las API de AWS en su nombre.

   AWS

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy.json
   ```

   AWS GovCloud (US)

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy_us-gov.json
   ```

   ```
   $ mv iam_policy_us-gov.json iam_policy.json
   ```

2. Cree una política de IAM con la política descargada en el paso anterior.

   ```
   $ aws iam create-policy \
       --policy-name AWSLoadBalancerControllerIAMPolicy \
       --policy-document file://iam_policy.json
   ```

   > **Note**
   >
   > Si ve la política en la AWS Management Console, la consola muestra advertencias para el servicio ELB, pero no para el servicio ELB v2. Esto ocurre porque algunas de

las acciones de la política existen para ELB v2, pero no para ELB. Puede obviar estas advertencias para ELB.

Creación de un rol de IAM mediante **eksctl**

- Reemplace *my-cluster* por el nombre de su clúster y *111122223333* con el ID de su cuenta y luego ejecute el comando. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

```
$ eksctl create iamserviceaccount \
  --cluster=my-cluster \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --attach-policy-
arn=arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

Paso 2: instalar el AWS Load Balancer Controller

Instale AWS Load Balancer Controller con Helm V3

1. Añada el repositorio de gráficos de Helm `eks-charts`. AWS mantiene este repositorio en GitHub.

```
$ helm repo add eks https://aws.github.io/eks-charts
```

2. Actualice el repositorio local para asegurarse de que cuenta con los gráficos más recientes.

```
$ helm repo update eks
```

3. Instale la AWS Load Balancer Controller.

   Reemplace *my-cluster* por el nombre del clúster. En el siguientes comando, `aws-load-balancer-controller` es la cuenta de servicio de Kubernetes que creó en un paso anterior.

   Para obtener más información acerca de la configuración de gráficos de Helm, consulte `values.yaml` en GitHub.

```
$ helm install aws-load-balancer-controller eks/aws-load-balancer-controller \
  -n kube-system \
  --set clusterName=my-cluster \
  --set serviceAccount.create=false \
  --set serviceAccount.name=aws-load-balancer-controller
```

a. Si está implementando el controlador en los nodos de Amazon EC2 que tienen acceso restringido al servicio de metadatos de la instancia de Amazon EC2 (IMDS), o si está implementando en Fargate, debe agregar los siguientes indicadores al siguiente comando de `helm`:

   - **`--set region=region-code`**
   - **`--set vpcId=vpc-xxxxxxxx`**

b. Para ver las versiones disponibles del gráfico de Helm y del controlador del equilibrador de carga, use el siguiente comando:

```
helm search repo eks/aws-load-balancer-controller --versions
```

> **Important**
>
> El gráfico implementado no recibe actualizaciones de seguridad de forma automática. Debe actualizar de forma manual a un gráfico más reciente cuando se encuentre disponible. Al actualizar, cambie *`install`* por **`upgrade`** en el comando anterior.
> El comando `helm install` instala automáticamente las definiciones de recursos personalizadas (CRDs) para el controlador. El comando `helm upgrade` no lo hace. Si usa `helm upgrade`, , debe instalar manualmente CRDs. Ejecute el siguiente comando para instalar CRDs:
>
> ```
> wget https://raw.githubusercontent.com/aws/eks-charts/master/stable/aws-load-balancer-controller/crds/crds.yaml
> kubectl apply -f crds.yaml
> ```

Paso 3: verificar que el controlador se encuentre instalado

1. Verifique que el controlador se encuentre instalado.

```
$ kubectl get deployment -n kube-system aws-load-balancer-controller
```

Un ejemplo de salida sería el siguiente.

```
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2            2           84s
```

Recibe la salida anterior si ha implementado mediante Helm. Si ha implementado utilizando el manifiesto de Kubernetes, solo tiene una réplica.

2. Antes de utilizar el controlador para aprovisionar el recurso de AWS, el clúster debe cumplir requisitos específicos. Para obtener más información, consulte Equilibrio de carga de aplicaciones en Amazon EKS y Equilibrio de carga de red en Amazon EKS.

Instalación del complemento de AWS Load Balancer Controller mediante manifiestos de Kubernetes

En este tema, se describe cómo instalar el controlador mediante la descarga y la aplicación de los manifiestos de Kubernetes. Puede ver la documentación completa para el controlador en GitHub.

En los siguientes pasos, reemplace *example values* por sus propios valores.

Requisitos previos

Antes de comenzar este tutorial, debe instalar y configurar las siguientes herramientas y recursos que necesitará para crear y administrar un clúster de Amazon EKS.

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.
- Un proveedor OpenID Connect (OIDC) de AWS Identity and Access Management (IAM) existente para el clúster. Para determinar si ya tiene un proveedor o para crear uno, consulte Creación de un proveedor de OIDC de IAM para su clúster.
- Asegúrese de que sus complementos Amazon VPC CNI plugin for Kubernetes, `kube-proxy` y CoreDNS tengan las versiones mínimas enumeradas en los tokens de cuenta de servicio.
- Familiaridad con AWS Elastic Load Balancing. Para obtener más información, consulte la Guía del usuario de Elastic Load Balancing.
- Familiaridad con el servicio de Kubernetes y los recursos de entrada.

Paso 1: configurar IAM

> **Note**
>
> Solo necesita crear un rol de IAM para el AWS Load Balancer Controller por cada cuenta de AWS. Compruebe si `AmazonEKSLoadBalancerControllerRole` existe en la Consola de IAM. Si este rol existe, vaya a the section called “Paso 2: instalar el `cert-manager`”.

Cree una política de IAM.

1. Descargue una política de IAM para el AWS Load Balancer Controller que le permita realizar llamadas a las API de AWS en su nombre.

   AWS

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy.json
   ```

   AWS GovCloud (US)

   ```
   $ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/install/iam_policy_us-gov.json
   ```

   ```
   $ mv iam_policy_us-gov.json iam_policy.json
   ```

2. Cree una política de IAM con la política descargada en el paso anterior.

   ```
   $ aws iam create-policy \
       --policy-name AWSLoadBalancerControllerIAMPolicy \
       --policy-document file://iam_policy.json
   ```

   > **Note**
   >
   > Si ve la política en la AWS Management Console, la consola muestra advertencias para el servicio ELB, pero no para el servicio ELB v2. Esto ocurre porque algunas de las acciones de la política existen para ELB v2, pero no para ELB. Puede obviar estas advertencias para ELB.

eksctl

Creación de un rol de IAM mediante **eksctl**

- Reemplace *my-cluster* por el nombre de su clúster y *111122223333* con el ID de su cuenta y luego ejecute el comando. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

```
$ eksctl create iamserviceaccount \
  --cluster=my-cluster \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --attach-policy-
arn=arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerIAMPolicy \
  --approve
```

AWS CLI and kubectl

Creación de un rol de IAM mediante AWS CLI y **kubectl**

1. Recupere el ID del proveedor de OIDC de su clúster y almacénelo en una variable.

```
oidc_id=$(aws eks describe-cluster --name my-cluster --query
 "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5)
```

2. Determine si un proveedor de OIDC de IAM con el ID de su clúster ya está en su cuenta. Debe tener configurado OIDC tanto para el clúster como para IAM.

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

   Si no se devuelve ninguna salida, debe contar con un proveedor de OIDC de IAM para su clúster. Si no se devuelve ninguna salida, debe crear un proveedor de OIDC de IAM para su clúster. Para obtener más información, consulte Creación de un proveedor de OIDC de IAM para su clúster.

3. Copie los siguientes contenidos en su dispositivo. Reemplace *111122223333* por su ID de cuenta. Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Reemplace *EXAMPLED539D4633E53DE1B71EXAMPLE* por los resultados que

se devolvieron en el paso anterior. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`. Después de reemplazar el texto, ejecute el comando modificado para crear el archivo `load-balancer-role-trust-policy.json`.

```
cat >load-balancer-role-trust-policy.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.region-code.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
                    "oidc.eks.region-code.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:kube-
system:aws-load-balancer-controller"
                }
            }
        }
    ]
}
EOF
```

4. Cree el rol de IAM.

```
aws iam create-role \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --assume-role-policy-document file://"load-balancer-role-trust-policy.json"
```

5. Adjunte la política de IAM administrada por Amazon EKS requerida al rol de IAM. Reemplace *111122223333* por su ID de cuenta.

```
aws iam attach-role-policy \
  --policy-arn
 arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerIAMPolicy \
```

```
  --role-name AmazonEKSLoadBalancerControllerRole
```

6. Copie los siguientes contenidos en su dispositivo. Reemplace *111122223333* por su ID de cuenta. Si su clúster está en las Regiones de AWS AWS GovCloud (Este de EE. UU.) o AWS GovCloud (Oeste de EE. UU.), reemplace `arn:aws:` con `arn:aws-us-gov:`. Después de reemplazar el texto, ejecute el comando modificado para crear el archivo `aws-load-balancer-controller-service-account.yaml`.

```
cat >aws-load-balancer-controller-service-account.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/name: aws-load-balancer-controller
  name: aws-load-balancer-controller
  namespace: kube-system
  annotations:
    eks.amazonaws.com/role-arn:
 arn:aws:iam::111122223333:role/AmazonEKSLoadBalancerControllerRole
EOF
```

7. Cree la cuenta de servicio Kubernetes en el clúster. La cuenta de servicio de Kubernetes denominada `aws-load-balancer-controller` está anotado con el rol de IAM que creó con el nombre *AmazonEKSLoadBalancerControllerRole*.

```
$ kubectl apply -f aws-load-balancer-controller-service-account.yaml
```

Paso 2: instalar el **`cert-manager`**

Instale el `cert-manager` con uno de los siguientes métodos para ingresar la configuración del certificado en los webhooks. Para obtener más información, consulte Introducción en la documentación de *`cert-manager`*.

Se recomienda utilizar el registro del contenedor `quay.io` para realizar la instalación de `cert-manager`. Si los nodos no tienen acceso al registro contenedor de `quay.io`, instale `cert-manager` mediante Amazon ECR (consulte a continuación).

Quay.io

Instalación de **`cert-manager`** usando Quay.io

- Si los nodos tienen acceso al registro de contenedores de `quay.io`, instale el `cert-manager` para ingresar la configuración del certificado en los webhooks.

```
$ kubectl apply \
    --validate=false \
    -f https://github.com/jetstack/cert-manager/releases/download/v1.13.5/cert-manager.yaml
```

Amazon ECR

Instalación de **`cert-manager`** mediante Amazon ECR

1. Instale el `cert-manager` con uno de los siguientes métodos para ingresar la configuración del certificado en los webhooks. Para obtener más información, consulte Introducción en la documentación de *`cert-manager`*.
2. Descargue el manifiesto.

```
curl -Lo cert-manager.yaml https://github.com/jetstack/cert-manager/releases/download/v1.13.5/cert-manager.yaml
```

3. Extraiga las siguientes imágenes y envíelas un repositorio al que tengan acceso sus nodos. Para obtener más información sobre cómo extraer, etiquetar y enviar las imágenes en su propio repositorio, consulte Copiar una imagen de contenedor de un repositorio en otro repositorio.

```
quay.io/jetstack/cert-manager-cainjector:v1.13.5
quay.io/jetstack/cert-manager-controller:v1.13.5
quay.io/jetstack/cert-manager-webhook:v1.13.5
```

4. Reemplace `quay.io` en el manifiesto de las tres imágenes por su propio nombre de registro. El siguiente comando supone que el nombre del repositorio privado es el mismo que el repositorio de origen. Reemplace *`111122223333.dkr.ecr.region-code.amazonaws.com`* por su registro privado.

```
$ sed -i.bak -e 's|quay.io|111122223333.dkr.ecr.region-code.amazonaws.com|' ./
cert-manager.yaml
```

5. Aplique el manifiesto.

```
$ kubectl apply \
    --validate=false \
    -f ./cert-manager.yaml
```

Paso 3: instalar el AWS Load Balancer Controller

Instalación de AWS Load Balancer Controller mediante un manifiesto de Kubernetes

1. Descargue la especificación del controlador. Para obtener más información sobre el controlador, consulte la documentación en GitHub.

```
curl -Lo v2_7_2_full.yaml https://github.com/kubernetes-sigs/aws-load-balancer-
controller/releases/download/v2.7.2/v2_7_2_full.yaml
```

2. Lleve a cabo las siguientes modificaciones en el archivo.

   a. Si ha descargado el archivo `v2_7_2_full.yaml`, ejecute el siguiente comando para eliminar la sección `ServiceAccount` del manifiesto. Si no elimina esta sección, se sobrescribirá la anotación obligatoria que hizo en la cuenta de servicio en un paso anterior. Al eliminar esta sección también conserva la cuenta de servicio que creó en un paso anterior si elimina el controlador.

```
$ sed -i.bak -e '596,604d' ./v2_7_2_full.yaml
```

   Si ha descargado una versión de archivo diferente, abra el archivo en un editor y elimine las siguientes líneas.

```
apiVersion: v1
kind: ServiceAccount
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/name: aws-load-balancer-controller
  name: aws-load-balancer-controller
```

```
  namespace: kube-system
---
```

b. Reemplace `your-cluster-name` en la sección `Deployment spec` del archivo por el nombre del clúster. Para ello, reemplace *`my-cluster`* por el nombre del clúster.

```
$ sed -i.bak -e 's|your-cluster-name|my-cluster|' ./v2_7_2_full.yaml
```

c. Si los nodos no tienen acceso a los repositorios de imágenes de Amazon ECR de Amazon EKS, tiene que extraer la siguiente imagen y enviarla a un repositorio al que tengan acceso los nodos. Para obtener más información sobre cómo extraer, etiquetar y enviar una imagen a su propio repositorio, consulte Copiar una imagen de contenedor de un repositorio en otro repositorio.

```
public.ecr.aws/eks/aws-load-balancer-controller:v2.7.2
```

Agregue el nombre del registro al manifiesto. El siguiente comando supone que el nombre del repositorio privado es el mismo que el repositorio de origen y agrega el nombre del registro privado al archivo. Reemplace *`111122223333.dkr.ecr.region-code.amazonaws.com`* por su registro. En esta línea se supone que ha asignado el mismo nombre al repositorio privado que al repositorio de origen. Si no es así, cambie el texto `eks/aws-load-balancer-controller` después del nombre del registro privado al nombre del repositorio.

```
$ sed -i.bak -e 's|public.ecr.aws/eks/aws-load-balancer-
controller|111122223333.dkr.ecr.region-code.amazonaws.com/eks/aws-load-
balancer-controller|' ./v2_7_2_full.yaml
```

d. (Necesario solo para IMDS restringido o Fargate)

Si implementa el controlador en los nodos de Amazon EC2 que tienen acceso restringido al servicio de metadatos de la instancia de Amazon EC2 (IMDS), o si realiza la implementación en Fargate, agregue lo **`following parameters`** en `- args:`.

```
[...]
spec:
      containers:
        - args:
            - --cluster-name=your-cluster-name
            - --ingress-class=alb
```

```
            - --aws-vpc-id=vpc-xxxxxxxx
            - --aws-region=region-code


[...]
```

3. Aplique el archivo.

```
$ kubectl apply -f v2_7_2_full.yaml
```

4. Descargue el manifiesto `IngressClass` y `IngressClassParams` a su clúster.

```
$ curl -Lo v2_7_2_ingclass.yaml https://github.com/kubernetes-sigs/aws-load-balancer-controller/releases/download/v2.7.2/v2_7_2_ingclass.yaml
```

5. Aplique el manifiesto al clúster.

```
$ kubectl apply -f v2_7_2_ingclass.yaml
```

Paso 4: verificar que el controlador se encuentre instalado

1. Verifique que el controlador se encuentre instalado.

```
$ kubectl get deployment -n kube-system aws-load-balancer-controller
```

   Un ejemplo de salida sería el siguiente.

```
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2     2            2           84s
```

   Recibe la salida anterior si ha implementado mediante Helm. Si ha implementado utilizando el manifiesto de Kubernetes, solo tiene una réplica.

2. Antes de utilizar el controlador para aprovisionar el recurso de AWS, el clúster debe cumplir requisitos específicos. Para obtener más información, consulte Equilibrio de carga de aplicaciones en Amazon EKS y Equilibrio de carga de red en Amazon EKS.

Migración desde un controlador obsoleto

En este tema se explica cómo migrar desde versiones obsoletas de controladores. Más específicamente, se describe cómo eliminar las versiones obsoletas de AWS Load Balancer Controller.

- Las versiones obsoletas no se pueden actualizar. Deben eliminarse y se debe instalar una versión actualizada del LBC instalado.
- Las versiones obsoleta incluyen lo siguiente:
  - AWSControlador de Ingress del ALB para Kubernetes («controlador de Ingress»), un predecesor del AWS Load Balancer Controller.
  - Cualquier versión `0.1.x` del AWS Load Balancer Controller

Eliminar la versión obsoleta del controlador

> **Note**
>
> Es posible que haya instalado la versión obsoleta con Helm o manualmente con manifiestos de Kubernetes. Realice el procedimiento utilizando la herramienta con la que la instaló originalmente.

Eliminación del controlador de entrada con Helm

1. Si ha instalado el gráfico de Helm `incubator/aws-alb-ingress-controller`, desinstálelo.

   ```
   $ helm delete aws-alb-ingress-controller -n kube-system
   ```

2. Si tiene la versión `0.1.x` del gráfico `eks-charts/aws-load-balancer-controller` instalado, desinstálelo. La actualización de `0.1.x` a la versión `1.0.0` no funciona debido a la incompatibilidad con la versión de la API webhook.

   ```
   $ helm delete aws-load-balancer-controller -n kube-system
   ```

Eliminación del controlador de entrada con manifiesto de Kubernetes

1. Verifique si el controlador se encuentra instalado actualmente.

```
$ kubectl get deployment -n kube-system alb-ingress-controller
```

Esta es la salida si el controlador no está instalado.

Error del servidor (no se encontró): deployments.apps “alb-ingress-controller” no se encontró

Esta es la salida si el controlador está instalado.

```
NAME                   READY UP-TO-DATE AVAILABLE AGE
alb-ingress-controller 1/1   1          1         122d
```

2. Ingrese el siguiente comando para eliminar el controlador.

```
$ kubectl delete -f https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-
ingress-controller/v1.1.8/docs/examples/alb-ingress-controller.yaml
kubectl delete -f https://raw.githubusercontent.com/kubernetes-sigs/aws-alb-
ingress-controller/v1.1.8/docs/examples/rbac-role.yaml
```

Migración a AWS Load Balancer Controller

Para migrar del controlador de entrada de ALB para Kubernetes al AWS Load Balancer Controller, realice lo siguiente:

1. Retire el controlador de entrada de ALB (consulte más arriba).
2. Instale el AWS Load Balancer Controller.
3. Añada una política adicional al rol de IAM utilizado por el LBC. Esta política permite al LBC administrar los recursos creados por el controlador de entrada de ALB para Kubernetes.

Añada la política de migración al rol de IAM de AWS Load Balancer Controller.

1. Descargue la política de IAM. Esta política permite al LBC administrar los recursos creados por el controlador de entrada de ALB para Kubernetes. También puede ver la política.

```
$ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-
controller/v2.7.2/docs/install/iam_policy_v1_to_v2_additional.json
```

2. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE.UU. Oeste), reemplace `arn:aws:` con `arn:aws-us-gov:`.

```
$ sed -i.bak -e 's|arn:aws:|arn:aws-us-gov:|' iam_policy_v1_to_v2_additional.json
```

3. Cree la política de IAM y anote el ARN devuelto.

```
$ aws iam create-policy \
  --policy-name AWSLoadBalancerControllerAdditionalIAMPolicy \
  --policy-document file://iam_policy_v1_to_v2_additional.json
```

4. Adjunte la política de IAM al rol de IAM que usa el LBC. Reemplace *your-role-name* por el nombre del rol, como `AmazonEKSLoadBalancerControllerRole`.

   Si creó el rol con `eksctl`, para encontrar el nombre del rol que se creó, abra la consola de AWS CloudFormation y seleccione la pila eksctl-***my-cluster***-addon-iamserviceaccount-kube-system-aws-load-balancer-controller. Seleccione la pestaña Recursos. El nombre del rol se encuentra en la columna de ID físicos. Si su clúster está en las Regiones de AWS GovCloud de AWS (EE. UU. Este) o GovCloud de AWS (EE. UU. Oeste), sustituya `arn:aws:` con `arn:aws-us-gov:`.

```
$ aws iam attach-role-policy \
  --role-name your-role-name \
  --policy-arn
 arn:aws:iam::111122223333:policy/AWSLoadBalancerControllerAdditionalIAMPolicy
```

Trabajar con el complemento CoreDNS de Amazon EKS

CoreDNS es un servidor de DNS flexible y extensible que puede servir como el DNS del clúster de Kubernetes. Al lanzar un clúster de Amazon EKS con al menos un nodo, se implementan dos réplicas de la imagen de CoreDNS de forma predeterminada, independientemente del número de nodos implementados en el clúster. Los Pods CoreDNS proporcionan resolución de nombres para todos los Pods del clúster. Los Pods CoreDNS se pueden implementar en los nodos de Fargate si su clúster incluye un Perfil de AWS Fargate con un espacio de nombres que coincida con el espacio de nombres para la `deployment` de CoreDNS. A fin de obtener más información sobre CoreDNS, consulte Uso de para la detección de serviciosCoreDNS en la documentación de Kubernetes.

En la siguiente tabla se muestra la versión más reciente del tipo de complemento de Amazon EKS para cada versión de Kubernetes.

Versión de Kubernetes	1.30	1.29	1.28	1.27	1.26	1.25	1.24	1.23
	v1.11.1-eksbuild	v1.11.1-eksbuild	v1.10.1-eksbuild.1	v1.10.1-eksbuild.1	v1.9.3-eksbuild.	v1.9.3-eksbuild.	v1.9.3-eksbuild.	v1.8.7-eksbuild.10

> **Important**
>
> Si administra este complemento, es posible que las versiones de la tabla no sean las mismas que las versiones autoadministradas disponibles. Para obtener más información acerca de la actualización de complementos autoadministrados, consulte Actualizar el complemento autoadministrado.

Consideraciones importantes sobre la actualización de CoreDNS

- Para mejorar la estabilidad y la disponibilidad de CoreDNS Deployment, las versiones `v1.9.3-eksbuild.6` y posteriores y `v1.10.1-eksbuild.3` se implementan con un `PodDisruptionBudget`. Si ha implementado un `PodDisruptionBudget` existente, la actualización a estas versiones podría fallar. Si se produce un error en la actualización, se solucionará el problema al completar una de las siguientes tareas:
  - Al actualizar el complemento Amazon EKS, elija anular la configuración existente como opción de resolución de conflictos. Si ha realizado otros ajustes personalizados en el Deployment, asegúrese de hacer una copia de seguridad de los ajustes antes de realizar la actualización para poder volver a aplicar los demás ajustes personalizados después de la actualización.
  - Elimine el `PodDisruptionBudget` que ya tiene y vuelva a intentar la actualización.
- En las versiones del complemento de EKS `v1.9.3-eksbuild.3` y posteriores, y `v1.10.1-eksbuild.6` y posteriores, Deployment de CoreDNS establece `readinessProbe` para utilizar el punto de conexión `/ready`. Este punto de conexión se habilita en el archivo de configuración `Corefile` de CoreDNS.

Si utiliza un `Corefile` personalizado, debe agregar el complemento `ready` a la configuración, para que el punto de conexión `/ready` esté activo en CoreDNS de modo que lo pueda utilizar la sonda.

- En las versiones del complemento de EKS `v1.9.3-eksbuild.7` y posteriores, y `v1.10.1-eksbuild.4` y posteriores, puede cambiar el objeto `PodDisruptionBudget`. Puede editar el complemento y cambiar esta configuración en Valores de configuración opcionales mediante los campos del siguiente ejemplo. En este ejemplo se muestra el objeto `PodDisruptionBudget` predeterminado.

```
{
    "podDisruptionBudget": {
        "enabled": true,
        "maxUnavailable": 1
        }
}
```

Puede establecer `maxUnavailable` o `minAvailable`, pero no puede establecer ambos en un solo `PodDisruptionBudget`. Para obtener más información sobre `PodDisruptionBudgets`, consulte Specifying a `PodDisruptionBudget` en la documentación de Kubernetes.

Tenga en cuenta que si establece `enabled` como `false`, no se elimina `PodDisruptionBudget`. Después de establecer este campo como `false`, debe eliminar el objeto `PodDisruptionBudget`. Del mismo modo, si edita el complemento para que utilice una versión anterior (es decir, desactualiza el complemento) después de actualizarlo a una versión con un objeto `PodDisruptionBudget`, no se elimina `PodDisruptionBudget`. Para eliminar el objeto `PodDisruptionBudget`, puede ejecutar el siguiente comando:

```
kubectl delete poddisruptionbudget coredns -n kube-system
```

- En las versiones complementarias de EKS `v1.10.1-eksbuild.5` y posteriores, cambie la tolerancia predeterminada de `node-role.kubernetes.io/master:NoSchedule` a `node-role.kubernetes.io/control-plane:NoSchedule` para cumplir con el KEP 2067. Para obtener más información sobre el KEP 2067, consulte KEP-2067: Rename the kubeadm "master" label and taint en las Propuestas de mejora de Kubernetes (KEP, por sus siglas en inglés) en GitHub.

En las versiones complementarias de EKS `v1.8.7-eksbuild.8` y posteriores y `v1.9.3-eksbuild.9` y posteriores, ambas tolerancias están configuradas para que sean compatibles con todas las versiones de Kubernetes.

- En las versiones complementarias de EKS `v1.9.3-eksbuild.11` y `v1.10.1-eksbuild.7` y posteriores, el CoreDNS Deployment establece un valor predeterminado para `topologySpreadConstraints`. El valor predeterminado garantiza que CoreDNS Pods se distribuyan entre las zonas de disponibilidad si hay nodos en varias zonas de disponibilidad disponibles. Puede establecer un valor personalizado que se utilizará en lugar del valor predeterminado. El valor predeterminado es el siguiente:

```
topologySpreadConstraints:
  - maxSkew: 1
    topologyKey: topology.kubernetes.io/zone
    whenUnsatisfiable: ScheduleAnyway
    labelSelector:
      matchLabels:
        k8s-app: kube-dns
```

Consideraciones de la actualización de CoreDNS v**1.11**

- En las versiones complementarias de EKS `v1.11.1-eksbuild.4` y posteriores, la imagen del contenedor está basada en una imagen básica mínima mantenida por Amazon EKS Distro, el cual contiene paquetes mínimos y no contiene intérpretes de comandos. Para obtener más información, consulte ¿Qué es Amazon EKS Distro? El uso y la solución de problemas de la imagen de CoreDNS siguen siendo los mismos.

Creación del complemento de Amazon EKS

Cree el tipo de Amazon EKS del complemento. Check

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.

1. Consulte qué versión del complemento está instalada en el clúster.

```
kubectl describe deployment coredns --namespace kube-system | grep coredns: | cut -
d : -f 3
```

Un ejemplo de salida sería el siguiente.

```
v1.10.1-eksbuild.11
```

2. Consulte qué tipo del complemento está instalado en el clúster. Según la herramienta con la que haya creado el clúster, es posible que actualmente no tenga instalado el tipo de complemento Amazon EKS en el clúster. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
 addon.addonVersion --output text
```

Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster y no es necesario que complete los pasos restantes del procedimiento. Si se devuelve un error, no tiene el tipo de complemento de Amazon EKS instalado en el clúster. Complete los pasos restantes de este procedimiento para instalarlo.

3. Guarde la configuración del complemento instalado actualmente.

```
kubectl get deployment coredns -n kube-system -o yaml > aws-k8s-coredns-old.yaml
```

4. Cree el complemento mediante el AWS CLI. Si desea utilizar AWS Management Console o `eksctl` para crear el complemento, consulteCreación de un complemento y especifique`coredns` para el nombre del complemento. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado.

   - Reemplace *my-cluster* por el nombre del clúster.
   - Reemplace *v1.11.1-eksbuild.9* por la versión más reciente que aparece en la tabla de versiones más recientes de la versión de su clúster.

```
aws eks create-addon --cluster-name my-cluster --addon-name coredns --addon-
version v1.11.1-eksbuild.9
```

Si ha aplicado una configuración personalizada al complemento actual que entra en conflicto con la configuración predeterminada del complemento de Amazon EKS, es posible que se produzca un error en la creación. Si se produce un error en la creación, recibe un error que puede serle de utilidad para resolver el problema. Como alternativa, puede añadir **`--resolve-conflicts OVERWRITE`** al comando anterior. Esto permite que el complemento sobrescriba cualquier configuración personalizada existente. Una vez que haya creado el complemento, puede actualizarlo con la configuración personalizada.

5. Confirme que la versión más reciente del complemento de la Kubernetes versión de su clúster se haya agregado al clúster. Reemplace *`my-cluster`* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
 addon.addonVersion --output text
```

Es posible que la creación del complemento tarde varios segundos en completarse.

Un ejemplo de salida sería el siguiente.

```
v1.11.1-eksbuild.9
```

6. Si ha realizado ajustes personalizados en el complemento original, antes de crear el complemento de Amazon EKS, utilice la configuración que guardó en el paso anterior para actualizar el complemento de Amazon EKS con su configuración personalizada.

Actualizar el complemento de Amazon EKS

Actualice el tipo de Amazon EKS del complemento. Si no ha agregado el tipo Amazon EKS del complemento al clúster, agréguelo o consulte Actualizar el complemento autoadministrado, en lugar de completar este procedimiento.

1. Consulte qué versión del complemento está instalada en el clúster. Reemplace *`my-cluster`* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
 "addon.addonVersion" --output text
```

Un ejemplo de salida sería el siguiente.

```
v1.10.1-eksbuild.11
```

Si la versión devuelta es la misma que la versión Kubernetes del clúster en la tabla de versiones más recientes, significa que ya tiene la última versión instalada en el clúster y no necesita completar el resto de este procedimiento. Si recibe un error, en lugar de un número de versión en la salida, significa que no tiene el tipo de versión de Amazon EKS en el clúster. Debe crear el complemento antes de poder actualizarlo mediante este procedimiento.

2. Guarde la configuración del complemento instalado actualmente.

```
kubectl get deployment coredns -n kube-system -o yaml > aws-k8s-coredns-old.yaml
```

3. Actualice el complemento con la AWS CLI. Si desea utilizar AWS Management Console o `eksctl` para actualizar el complemento, consulte Actualización de un complemento. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado.

   - Reemplace *my-cluster* por el nombre del clúster.
   - Reemplace *v1.11.1-eksbuild.9* por la versión más reciente que aparece en la tabla de versiones más recientes de la versión de su clúster.
   - La opción *CONSERVAR* de **--resolve-conflicts** conserva los valores de configuración existentes del complemento. Si ha establecido valores personalizados para la configuración del complemento y no utiliza esta opción, Amazon EKS sobrescribe los valores con los valores predeterminados. Si utiliza esta opción, le recomendamos que pruebe cualquier cambio de campo y valor en un clúster que no sea de producción antes de actualizar el complemento del clúster de producción. Si cambia este valor a `OVERWRITE`, todas las configuraciones cambiarán a los valores predeterminados de Amazon EKS. Si ha establecido valores personalizados para cualquier configuración, es posible que se sobrescriban con los valores predeterminados de Amazon EKS. Si cambia este valor a `none`, Amazon EKS no cambia el valor de ninguna configuración, pero la actualización podría fallar. Si se produce un error en la actualización, recibe un mensaje de error que lo ayuda a resolver el conflicto.
   - Si no va a actualizar un ajuste de configuración, elimine **--configuration-values '{*"replicaCount":3*}'** del comando. Si está actualizando una configuración, reemplace *"replicaCount":3* por la configuración que desee establecer. En este ejemplo, el número de réplicas de CoreDNS se establece en `3`. El valor que especifique debe ser válido para el esquema de configuración. Si no conoce el esquema de configuración, ejecute **aws eks**

describe-addon-configuration --addon-name coredns --addon-version *v1.11.1-eksbuild.9* y reemplace *v1.11.1-eksbuild.9* por el número de versión del complemento cuya configuración desea ver. El esquema se devuelve en la salida. Si ya tiene alguna configuración personalizada, quiere eliminarla por completo y volver a establecer los valores de todos los ajustes a los valores predeterminados de Amazon EKS, elimine *"replicaCount":3* del comando para que quede vacío **{}**. Para obtener más información sobre la configuración de CoreDNS, consulte Personalización del servicio DNS en la documentación de Kubernetes.

```
aws eks update-addon --cluster-name my-cluster --addon-name coredns --addon-
version v1.11.1-eksbuild.9 \
    --resolve-conflicts PRESERVE --configuration-values '{"replicaCount":3}'
```

La actualización puede tardar varios segundos en completarse.

4. Confirme que la versión del complemento se ha actualizado. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns
```

La actualización puede tardar varios segundos en completarse.

Un ejemplo de salida sería el siguiente.

```
{
    "addon": {
        "addonName": "coredns",
        "clusterName": "my-cluster",
        "status": "ACTIVE",
        "addonVersion": "v1.11.1-eksbuild.9",
        "health": {
            "issues": []
        },
        "addonArn": "arn:aws:eks:region:111122223333:addon/my-cluster/coredns/
d2c34f06-1111-2222-1eb0-24f64ce37fa4",
        "createdAt": "2023-03-01T16:41:32.442000+00:00",
        "modifiedAt": "2023-03-01T18:16:54.332000+00:00",
        "tags": {},
        "configurationValues": "{\"replicaCount\":3}"
    }
```

```
}
```

Actualizar el complemento autoadministrado

> **Important**
>
> Recomendamos agregar el tipo de complemento de Amazon EKS al clúster en lugar de utilizar el tipo de complemento autoadministrado. Si no está familiarizado con la diferencia entre los tipos, consulte the section called “Complementos de Amazon EKS”. Para obtener más información acerca de cómo agregar un complemento de Amazon EKS al clúster, consulte the section called “Creación de un complemento”. Si no puede usar el complemento de Amazon EKS, le recomendamos que envíe una pregunta sobre los motivos por los que no puede hacerlo al repositorio de GitHub de la hoja de ruta de contenedores.

1. Confirme que tiene instalado en el clúster el tipo de complemento autoadministrado. Reemplace *my-cluster* por el nombre de su clúster.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
    addon.addonVersion --output text
   ```

   Si se devuelve un mensaje de error, tiene el tipo de complemento autoadministrado instalado en el clúster. Complete los pasos restantes de este procedimiento. Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster. Para actualizar el tipo de Amazon EKS del complemento, siga el procedimiento que aparece en Actualizar el complemento de Amazon EKS, en lugar de este procedimiento. Si no está familiarizado con las diferencias entre los tipos de complementos, consulte Complementos de Amazon EKS.
2. Consulte qué versión de la imagen del contenedor está instalada actualmente en el clúster.

   ```
   kubectl describe deployment coredns -n kube-system | grep Image | cut -d ":" -f 3
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v1.8.7-eksbuild.2
   ```
3. Si la versión actual de CoreDNS es `v1.5.0` o posterior, pero anterior a la versión que aparece en la tabla de versiones de CoreDNS, omita este paso. Si su versión actual es anterior a `1.5.0`,

debe modificar el `ConfigMap` para que CoreDNS utilice el complemento de reenvío, en lugar del complemento proxy.

1. Abra el mapa de configuración con el siguiente comando.

```
kubectl edit configmap coredns -n kube-system
```

2. Sustituya el `proxy` en la línea siguiente por `forward`. Guarde el archivo y salga del editor.

```
proxy . /etc/resolv.conf
```

4. Si implementó su clúster en Kubernetes `1.17` o una versión anterior inicialmente, es posible que deba eliminar un término interrumpido de su manifiesto CoreDNS.

> **Important**
>
> Debe completar esto antes de actualizar a la versión `1.7.0` de CoreDNS, pero se recomienda que complete este paso incluso si está actualizando a una versión anterior.

1. Verifique si su manifiesto CoreDNS cuenta con la línea.

```
kubectl get configmap coredns -n kube-system -o jsonpath='{$.data.Corefile}' |
 grep upstream
```

Si no se devuelve un resultado, el manifiesto no cuenta con la línea y puede pasar al siguiente paso para actualizar CoreDNS. Si se devuelve el resultado, debe eliminar la línea.

2. Edite el `ConfigMap` con el siguiente comando, al eliminar la línea en el archivo que tiene la palabra `upstream` en ella. No realice más cambios en el archivo. Una vez que elimine la línea, guarde los cambios.

```
kubectl edit configmap coredns -n kube-system -o yaml
```

5. Recupere su imagen actual de CoreDNS:

```
kubectl describe deployment coredns -n kube-system | grep Image
```

Un ejemplo de salida sería el siguiente.

```
602401143452.dkr.ecr.region-code.amazonaws.com/eks/coredns:v1.8.7-eksbuild.2
```

6. Si está actualizando a CoreDNS `1.8.3` o posterior, debe agregar el permiso `endpointslices` para el del `system:coredns` Kubernetes `clusterrole`.

```
kubectl edit clusterrole system:coredns -n kube-system
```

Agregue las siguientes líneas en las líneas de permisos existentes en la sección `rules` del archivo.

```
[...]
- apiGroups:
  - discovery.k8s.io
  resources:
  - endpointslices
  verbs:
  - list
  - watch
[...]
```

7. Actualice el complemento CoreDNS reemplazando *602401143452* y *region-code* con los valores de la salida devuelta en un paso anterior. Reemplace *v1.11.1-eksbuild.9* por la versión de CoreDNS que aparece en la tabla de versiones más recientes de la versión de Kubernetes.

```
kubectl set image deployment.apps/coredns -n kube-system
 coredns=602401143452.dkr.ecr.region-code.amazonaws.com/eks/coredns:v1.11.1-
eksbuild.9
```

Un ejemplo de salida sería el siguiente.

```
deployment.apps/coredns image updated
```

8. Vuelva a comprobar la versión de la imagen del contenedor para confirmar que se actualizó a la versión que especificó en el paso anterior.

```
kubectl describe deployment coredns -n kube-system | grep Image | cut -d ":" -f 3
```

Un ejemplo de salida sería el siguiente.

```
v1.11.1-eksbuild.9
```

Escalado automático CoreDNS

Al lanzar un clúster de Amazon EKS con al menos un nodo, se implementa un Deployment de dos réplicas de la imagen de CoreDNS de forma predeterminada, independientemente del número de nodos implementados en el clúster. Los pods de CoreDNS proporcionan resolución de nombres para todos los pods del clúster. Las aplicaciones utilizan la resolución de nombres para conectarse a los pods y los servicios del clúster, así como para conectarse a los servicios fuera del clúster. A medida que aumenta el número de solicitudes de resolución de nombres (consultas) procedentes de los pods, los pods de CoreDNS se sobrecargan y se ralentizan, y se rechazan las solicitudes que los pods no pueden gestionar.

Para gestionar la mayor carga de los pods de CoreDNS, considere la posibilidad de utilizar un sistema de escalado automático para CoreDNS. Amazon EKS puede administrar el escalado automático de la implementación de CoreDNS en la versión CoreDNS del complemento de EKS. Este escalador automático CoreDNS monitorea continuamente el estado del clúster, incluida la cantidad de nodos y núcleos de CPU. En función de esa información, el controlador adaptará dinámicamente el número de réplicas de la implementación de CoreDNSen un clúster de EKS. Esta característica funciona en CoreDNS `v1.9` y en la versión de lanzamiento de EKS `1.25` y versiones superiores. Para obtener más información acerca de qué versiones son compatibles con el escalado automático CoreDNS, consulte la siguiente sección.

Recomendamos utilizar esta característica junto con otras prácticas recomendadas de escalado automático de clústeres de EKS para mejorar la disponibilidad general de las aplicaciones y la escalabilidad de los clústeres.

Requisitos previos

Para que Amazon EKS escale la implementación de CoreDNS, se deben cumplir tres requisitos previos:

- Debe utilizar la versión de CoreDNS del complemento de EKS.
- El clúster debe ejecutar al menos las versiones de clúster y plataforma mínimas.
- El clúster debe ejecutar al menos la versión mínima del complemento de EKS de CoreDNS.

Versión mínima del clúster

El escalado automático de CoreDNS se realiza mediante un nuevo componente en el plano de control del clúster, administrado por Amazon EKS. Por este motivo, debe actualizar el clúster a una versión de EKS que sea compatible con la versión de la plataforma mínima que incluya el nuevo componente.

Un nuevo clúster de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS. El clúster debe ser la versión `1.25` de Kubernetes o posterior. El clúster debe estar ejecutando una de las versiones de Kubernetes y versiones de la plataforma que se enumeran en la siguiente tabla o una versión superior. Tenga en cuenta que también se admite cualquier versión de Kubernetes y de la plataforma posterior a las enumeradas. Puede comprobar la versión actual de Kubernetes reemplazando *`my-cluster`* en el siguiente comando por el nombre del clúster y luego ejecutando el comando modificado:

```
aws eks describe-cluster
              --name my-cluster --query cluster.version --output
              text
```

Versión de Kubernetes	Versión de la plataforma
`1.29.3`	`eks.7`
`1.28.8`	`eks.13`
`1.27.12`	`eks.17`
`1.26.15`	`eks.18`
`1.25.16`	`eks.19`

> **Note**
>
> También se admiten todas las versiones de plataforma de las versiones de Kubernetes posteriores, por ejemplo, versión Kubernetes `1.30` de `eks.1` y posterior.

Versión mínima del complemento de EKS

Versión de Kubernetes	1.29	1.28	1.27	1.26	1.25
	v1.11.1-eksbuild.9	v1.10.1-eksbuild.11	v1.10.1-eksbuild.11	v1.9.3-eksbuild.15	v1.9.3-eksbuild.15

Configuración del CoreDNS escalado automático de en AWS Management Console

1. Asegúrese de que el clúster sea igual o superior a la versión mínima del clúster.

   Amazon EKS actualiza automáticamente los clústeres entre versiones de la plataforma de la misma versión de Kubernetes y no puede iniciar este proceso usted mismo. En su lugar, puede actualizar el clúster a la siguiente versión de Kubernetes, y el clúster se actualizará a esa versión de K8s y a la versión de la plataforma más reciente. Por ejemplo, si actualiza de `1.25` a `1.26`, el clúster se actualizará a `1.26.15 eks.18`.

   Las nuevas versiones de Kubernetes suelen presentar cambios significativos. Por ende, recomendamos que pruebe el comportamiento de las aplicaciones con un clúster separado de la nueva versión de Kubernetes antes de realizar la actualización en los clústeres de producción.

   Para actualizar un clúster a una nueva versión de Kubernetes, siga el procedimiento descrito en Actualización de una versión de Kubernetes de clúster de Amazon EKS.
2. Asegúrese de tener el complemento de EKS para CoreDNS, no la implementación autoadministrada de CoreDNS.

   Según la herramienta con la que haya creado el clúster, es posible que actualmente no tenga instalado el tipo de complemento Amazon EKS en el clúster. Para ver qué tipo de complemento está instalado en el clúster, puede ejecutar el siguiente comando. Reemplace `my-cluster` por el nombre del clúster.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
    addon.addonVersion --output text
   ```

Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster y puede continuar con el siguiente paso. Si se devuelve un error, no tiene el tipo de complemento de Amazon EKS instalado en el clúster. Complete los pasos restantes del procedimiento Creación del complemento de Amazon EKS para reemplazar la versión autoadministrada por el complemento de Amazon EKS.

3. Asegúrese de que la versión del complemento de EKS para CoreDNS sea igual o superior a la versión mínima del complemento de EKS.

   Consulte qué versión del complemento está instalada en el clúster. Puede verificarlo en la AWS Management Console o ejecutar el siguiente comando:

   ```
   kubectl describe deployment coredns --namespace kube-system | grep coredns: | cut -d : -f 3
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v1.10.1-eksbuild.11
   ```

   Compare esta versión con la versión mínima del complemento de EKS de la sección anterior. Si es necesario, actualice el complemento de EKS a una versión superior; para ello, siga el procedimiento Actualizar el complemento de Amazon EKS.

4. Agregue la configuración de escalado automático a los ajustes de configuración opcionales del complemento de EKS.

   a. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

   b. En el panel de navegación izquierdo, seleccione Clusters (Clústeres) y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento.

   c. Elija la pestaña Complementos.

   d. Seleccione la casilla situada en la parte superior derecha del cuadro del complemento CoreDNS y, a continuación, elija Editar.

   e. En la página Configuración de CoreDNS, haga lo siguiente:

      i. Seleccione la Version (Versión) que desea utilizar. Le recomendamos que mantenga la misma versión que en el paso anterior y que actualice la versión y la configuración en acciones separadas.

ii. Seleccione Ajustes de configuración opcionales.

iii. Introduzca la clave JSON `"autoscaling":` y el valor de un objeto JSON anidado con una clave `"enabled":` y un valor `true` en Valores de configuración. El texto resultante debe ser un objeto JSON válido. Si esta clave y este valor son los únicos datos del cuadro de texto, rodee la clave y el valor entre corchetes `{}`. En el siguiente ejemplo, se muestra que el escalado automático está activado:

```
{
  "autoScaling": {
    "enabled": true
  }
}
```

iv. (Opcional) Puede proporcionar valores mínimos y máximos a los que el escalado automático pueda escalar la cantidad de pods de CoreDNS.

En el siguiente ejemplo, se muestra que el escalado automático está activado y que todas las claves opcionales tienen valores. Recomendamos que la cantidad mínima de pods de CoreDNS sea siempre superior a 2 para proporcionar resiliencia al servicio DNS del clúster.

```
{
  "autoScaling": {
    "enabled": true,
    "minReplicas": 2,
    "maxReplicas": 10
  }
}
```

f. Para aplicar la nueva configuración mediante la sustitución de los pods de CoreDNS, seleccione Guardar cambios.

Amazon EKS aplica los cambios a los complementos de EKS mediante la implementación de Kubernetes para CoreDNS. Puede hacer un seguimiento del estado del lanzamiento en el historial de actualizaciones en la AWS Management Console y con `kubectl rollout status deployment/coredns --namespace kube-system`.

`kubectl rollout` tiene los siguientes comandos:

```
$ kubectl rollout
```

```
history  -- View rollout history
pause    -- Mark the provided resource as paused
restart  -- Restart a resource
resume   -- Resume a paused resource
status   -- Show the status of the rollout
undo     -- Undo a previous rollout
```

Si la implementación lleva demasiado tiempo, Amazon EKS la anulará y se agregará al historial de actualizaciones del complemento un mensaje con el tipo de actualización del complemento y el estado Fallido. Para investigar cualquier problema, comience por el historial de la implementación y ejecute `kubectl logs` en un pod de CoreDNS para ver los registros de CoreDNS.

5. Si la nueva entrada en el historial de actualizaciones tiene el estado Correcto, esto significa que la implementación se ha completado y que el complemento está utilizando la nueva configuración en todos los pods de CoreDNS. A medida que cambia la cantidad de nodos y los núcleos de CPU de los nodos del clúster, Amazon EKS escala la cantidad de réplicas de la implementación de CoreDNS.

Configuración del CoreDNS escalado automático de en AWS Command Line Interface

1. Asegúrese de que el clúster sea igual o superior a la versión mínima del clúster.

   Amazon EKS actualiza automáticamente los clústeres entre versiones de la plataforma de la misma versión de Kubernetes y no puede iniciar este proceso usted mismo. En su lugar, puede actualizar el clúster a la siguiente versión de Kubernetes, y el clúster se actualizará a esa versión de K8s y a la versión de la plataforma más reciente. Por ejemplo, si actualiza de `1.25` a `1.26`, el clúster se actualizará a `1.26.15 eks.18`.

   Las nuevas versiones de Kubernetes suelen presentar cambios significativos. Por ende, recomendamos que pruebe el comportamiento de las aplicaciones con un clúster separado de la nueva versión de Kubernetes antes de realizar la actualización en los clústeres de producción.

   Para actualizar un clúster a una nueva versión de Kubernetes, siga el procedimiento descrito en Actualización de una versión de Kubernetes de clúster de Amazon EKS.

2. Asegúrese de tener el complemento de EKS para CoreDNS, no la implementación autoadministrada de CoreDNS.

Según la herramienta con la que haya creado el clúster, es posible que actualmente no tenga instalado el tipo de complemento Amazon EKS en el clúster. Para ver qué tipo de complemento está instalado en el clúster, puede ejecutar el siguiente comando. Reemplace `my-cluster` por el nombre del clúster.

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns --query
 addon.addonVersion --output text
```

Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster. Si se devuelve un error, no tiene el tipo de complemento de Amazon EKS instalado en el clúster. Complete los pasos restantes del procedimiento Creación del complemento de Amazon EKS para reemplazar la versión autoadministrada por el complemento de Amazon EKS.

3. Asegúrese de que la versión del complemento de EKS para CoreDNS sea igual o superior a la versión mínima del complemento de EKS.

   Consulte qué versión del complemento está instalada en el clúster. Puede verificarlo en la AWS Management Console o ejecutar el siguiente comando:

   ```
   kubectl describe deployment coredns --namespace kube-system | grep coredns: | cut -
   d : -f 3
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v1.10.1-eksbuild.11
   ```

   Compare esta versión con la versión mínima del complemento de EKS de la sección anterior. Si es necesario, actualice el complemento de EKS a una versión superior; para ello, siga el procedimiento Actualizar el complemento de Amazon EKS.

4. Agregue la configuración de escalado automático a los ajustes de configuración opcionales del complemento de EKS.

   Ejecute el siguiente comando de la AWS CLI. Reemplace `my-cluster` por el nombre del clúster y el ARN del rol de IAM por el rol que va a usar.

   ```
   aws eks update-addon --cluster-name my-cluster --addon-name coredns \
   ```

```
    --resolve-conflicts PRESERVE --configuration-values '{"autoScaling":
{"enabled":true}}'
```

Amazon EKS aplica los cambios a los complementos de EKS mediante la implementación de Kubernetes para CoreDNS. Puede hacer un seguimiento del estado del lanzamiento en el historial de actualizaciones en la AWS Management Console y con `kubectl rollout status deployment/coredns --namespace kube-system`.

`kubectl rollout` tiene los siguientes comandos:

```
kubectl rollout

history  -- View rollout history
pause    -- Mark the provided resource as paused
restart  -- Restart a resource
resume   -- Resume a paused resource
status   -- Show the status of the rollout
undo     -- Undo a previous rollout
```

Si la implementación lleva demasiado tiempo, Amazon EKS la anulará y se agregará al historial de actualizaciones del complemento un mensaje con el tipo de actualización del complemento y el estado Fallido. Para investigar cualquier problema, comience por el historial de la implementación y ejecute `kubectl logs` en un pod de CoreDNS para ver los registros de CoreDNS.

5. (Opcional) Puede proporcionar valores mínimos y máximos a los que el escalado automático pueda escalar la cantidad de pods de CoreDNS.

   En el siguiente ejemplo, se muestra que el escalado automático está activado y que todas las claves opcionales tienen valores. Recomendamos que la cantidad mínima de pods de CoreDNS sea siempre superior a 2 para proporcionar resiliencia al servicio DNS del clúster.

   ```
   aws eks update-addon --cluster-name my-cluster --addon-name coredns \
       --resolve-conflicts PRESERVE --configuration-values '{"autoScaling":
   {"enabled":true}, "minReplicas": 2, "maxReplicas": 10}'
   ```

6. Consulte el estado de la actualización del complemento mediante la ejecución del siguiente comando:

```
aws eks describe-addon --cluster-name my-cluster --addon-name coredns \
```

Si ve esta línea: `"status": "ACTIVE"`, significa que la implementación se ha completado y que el complemento está usando la nueva configuración en todos los pods de CoreDNS. A medida que cambia la cantidad de nodos y los núcleos de CPU de los nodos del clúster, Amazon EKS escala la cantidad de réplicas de la implementación de CoreDNS.

Métricas de CoreDNS

CoreDNS como un complemento de EKS expone las métricas de CoreDNS en el puerto 9153 en el formato Prometheus en el servicio de `kube-dns`. Puede utilizar Prometheus, el agente de Amazon CloudWatch o cualquier otro sistema compatible para raspar (recopilar) estas métricas.

Para ver un ejemplo de configuración de raspado que sea compatible tanto con Prometheus como con el agente de CloudWatch, consulte CloudWatch agent configuration for Prometheus en la Guía del usuario de Amazon CloudWatch.

Trabajar con el complemento **`kube-proxy`** Kubernetes

> **Important**
>
> Recomendamos agregar el tipo de complemento de Amazon EKS al clúster en lugar de utilizar el tipo de complemento autoadministrado. Si no está familiarizado con la diferencia entre los tipos, consulte the section called “Complementos de Amazon EKS”. Para obtener más información acerca de cómo agregar un complemento de Amazon EKS al clúster, consulte the section called “Creación de un complemento”. Si no puede usar el complemento de Amazon EKS, le recomendamos que envíe una pregunta sobre los motivos por los que no puede hacerlo al repositorio de GitHub de la hoja de ruta de contenedores.

El complemento `kube-proxy` se implementa en cada nodo de Amazon EC2 del clúster de Amazon EKS. Mantiene las reglas de red en los nodos y permite la comunicación de red con los Pods. El complemento no se implementa en los nodos de Fargate del clúster. Para obtener más información, consulte `kube-proxy` en la documentación del Kubernetes.

En la siguiente tabla se muestra la versión más reciente del tipo de complemento de Amazon EKS para cada versión de Kubernetes.

Versión de Kubernetes	1.30	1.29	1.28	1.27	1.26	1.25	1.24	1.23
	v1.30.0-eksbuild.5	v1.29.3-eksbuild.5	v1.28.8-eksbuild.8	v1.27.16-eksbuild.8	v1.26.15-eksbuild.8	v1.25.16-eksbuild.8	v1.24.17-eksbuild.8	v1.23.17-eksbuild.9

> **Important**
>
> Una versión anterior de la documentación era incorrecta. Las versiones `v1.28.5`, `v1.27.9`, y `v1.26.12` de `kube-proxy` no están disponibles.
> Si administra este complemento, es posible que las versiones de la tabla no sean las mismas que las versiones autoadministradas disponibles.

Existen dos tipos de imagen de contenedor `kube-proxy` disponibles para cada versión de clúster de Amazon EKS:

- Predeterminado: este tipo de imagen se basa en una imagen de Docker basada en Debian y mantenida por la comunidad ascendente de Kubernetes.
- Mínimo: este tipo se basa en una imagen de base mínima mantenida por Amazon EKS Distro, la cual contiene los paquetes mínimos y no tiene intérprete de comandos. Para obtener más información, consulte ¿Qué es Amazon EKS Distro?

Versión de imagen de contenedor de **`kube-proxy`** más reciente, autoadministrada y disponible para cada versión de clúster de Amazon EKS

Image type (Tipo de imagen)	**1.30**	**1.29**	**1.28**	**1.27**	**1.26**	**1.25**	**1.24**	**1.23**
kube-proxy (tipo predeterminado)	Solo hay un tipo mínimo disponible	Solo hay un tipo mínimo disponible	Solo hay un tipo mínimo disponible	Solo hay un tipo mínimo disponible	Solo hay un tipo mínimo disponible	Solo hay un tipo mínimo disponible	v1.24.1-eksbuild.2	v1.23.16-eksbuild.2
kube-proxy (tipo mínimo)	v1.30.0-minimal-eksbuild.	v1.29.3-minimal-eksbuild.	v1.28.8-minimal-eksbuild.	v1.27.1-minimal-eksbuild	v1.26.1-minimal-eksbuild	v1.25.1-minimal-eksbuild	v1.24.1-minimal-eksbuild	v1.23.17-minimal-eksbuild.5

> **Important**
>
> - El tipo de imagen predeterminado no está disponible para la versión Kubernetes de `1.25` y posteriores. Debe utilizar el tipo de imagen mínimo.
> - Cuando actualiza un tipo de complemento de Amazon EKS, especifica una versión de complemento de Amazon EKS válida, que puede ser una versión que no aparece en esta tabla. Esto se debe a que las versiones del complemento de Amazon EKS no siempre coinciden con las versiones de imágenes del contenedor especificadas al actualizar el tipo autoadministrado de este complemento. Al actualizar el tipo autoadministrado de este complemento, se especifica una versión válida de la imagen del contenedor que aparece en esta tabla.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.

Consideraciones

- Kube-proxy en un clúster de Amazon EKS tiene la misma política de compatibilidad y sesgo que Kubernetes. Aprenda cómo Recuperación de la compatibilidad de las versiones del complemento.
- Kube-proxy debe ser la misma versión secundaria que kubelet en los nodos de Amazon EC2.
- El Kube-proxy no puede ser posterior a la versión secundaria del plano de control del clúster.
- Si recientemente actualizó el clúster a una nueva versión secundaria de Kubernetes, actualice los nodos de Amazon EC2 a la misma versión secundaria antes de actualizar el kube-proxy a la misma versión secundaria que los nodos.

Para actualizar el complemento autoadministrado **kube-proxy**

1. Confirme que tiene instalado en el clúster el tipo de complemento autoadministrado. Reemplace *my-cluster* por el nombre de su clúster.

   ```
   aws eks describe-addon --cluster-name my-cluster --addon-name kube-proxy --query
    addon.addonVersion --output text
   ```

   Si se devuelve un mensaje de error, tiene el tipo de complemento autoadministrado instalado en el clúster. Los pasos restantes de este tema son para actualizar el tipo de complemento autoadministrado. Si se devuelve el número de versión, tiene el tipo de complemento de Amazon EKS instalado en el clúster. Para actualizarlo, siga el procedimiento que aparece en Actualización de un complemento, en lugar del procedimiento descrito en este tema. Si no está familiarizado con las diferencias entre los tipos de complementos, consulte Complementos de Amazon EKS.
2. Consulte qué versión de la imagen del contenedor está instalada actualmente en el clúster.

   ```
   kubectl describe daemonset kube-proxy -n kube-system | grep Image
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   Image:    602401143452.dkr.ecr.region-code.amazonaws.com/eks/kube-proxy:v1.29.1-
   eksbuild.2
   ```

   En el ejemplo de resultado, *v1.29.1-eksbuild.2* es la versión instalada en el clúster.

3. Actualice el complemento kube-proxy reemplazando *602401143452* y *region-code* por los valores de la salida en el paso anterior. Reemplace *v1.30.0-eksbuild.3* por la versión de kube-proxy que aparece en la última versión autoadministrada disponible de la imagen del contenedor kube-proxy para cada tabla de versiones del clúster de Amazon EKS. Puede especificar el número de versión de tipo de imagen predeterminada o mínima.

```
kubectl set image daemonset.apps/kube-proxy -n kube-system kube-
proxy=602401143452.dkr.ecr.region-code.amazonaws.com/eks/kube-proxy:v1.30.0-
eksbuild.3
```

Un ejemplo de salida sería el siguiente.

```
daemonset.apps/kube-proxy image updated
```

4. Confirme que la nueva versión ya esté instalada en el clúster.

```
kubectl describe daemonset kube-proxy -n kube-system | grep Image | cut -d ":" -f 3
```

Un ejemplo de salida sería el siguiente.

```
v1.30.0-eksbuild.3
```

5. Si utiliza nodos x86 y Arm en el mismo clúster y su clúster se implementó antes del 17 de agosto de 2020. A continuación, edite el manifiesto de kube-proxy a fin de incluir un selector de nodos para varias arquitecturas de hardware con el siguiente comando. Esta es una operación que se realiza una vez. Después de agregar el selector al manifiesto, no es necesario que lo agregue cada vez que realiza una actualización del complemento. Si el clúster se implementó a partir del 17 de agosto de 2020, kube-proxy ya cuenta con capacidad de varias arquitecturas.

```
kubectl edit -n kube-system daemonset/kube-proxy
```

Agregue el siguiente selector de nodos al archivo en el editor y guárdelo. Para ver un ejemplo de dónde incluir este texto en el editor, consulte el archivo de manifiesto de CNI en GitHub. Esto permite a Kubernetes extraer la imagen de hardware correcta según la arquitectura de hardware del nodo.

```
- key: "kubernetes.io/arch"
```

```
  operator: In
  values:
  - amd64
  - arm64
```

6. Si su clúster se creó inicialmente con la versión de `1.14` Kubernetes o posterior, puede omitir este paso porque `kube-proxy` ya incluye esta `Affinity Rule`. Si creó inicialmente un clúster de Amazon EKS con versión `1.13` de Kubernetes o anterior y tiene la intención de utilizar nodos de Fargate, edite su manifiesto de `kube-proxy` para incluir una regla `NodeAffinity` a fin de evitar que los Pods `kube-proxy` programen en los nodos de Fargate. Esta es una edición que se realiza una vez. Después de agregar el `Affinity Rule` al manifiesto, no es necesario que lo agregue cada vez que realiza una actualización del complemento. Edite su `kube-proxy` `DaemonSet`.

   ```
   kubectl edit -n kube-system daemonset/kube-proxy
   ```

   Agregue la siguiente `Affinity Rule` a la sección `DaemonSet spec` del archivo en el editor y guárdelo. Para ver un ejemplo de dónde incluir este texto en el editor, consulte el archivo de manifiesto de CNI en GitHub.

   ```
   - key: eks.amazonaws.com/compute-type
     operator: NotIn
     values:
     - fargate
   ```

Acceder a Amazon Elastic Kubernetes Service mediante un punto de conexión de interfaz (AWS PrivateLink)

Puede usar un AWS PrivateLink para crear una conexión privada entre la VPC y Amazon Elastic Kubernetes Service. Puede acceder a Amazon EKS como si estuviera en su VPC, sin el uso de una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect. Las instancias de la VPC no necesitan direcciones IP públicas para acceder a Amazon EKS.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred

habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a Amazon EKS.

Para obtener más información, consulte Acceso a los Servicios de AWS a través de AWS PrivateLink en la Guía de AWS PrivateLink.

Consideraciones para Amazon EKS

- Antes de configurar un punto de conexión para Amazon EKS, consulte Considerations (Consideraciones) en la Guía de AWS PrivateLink.
- Amazon EKS admite la realización de llamadas a todas las acciones de la API a través del punto de conexión de interfaz, pero no a las API de Kubernetes. El servidor de la API de Kubernetes ahora admite un punto de conexión privado. El punto de conexión del servidor de la API de Kubernetes crea un punto de conexión para el servidor de la API de Kubernetes que se utiliza con el fin de comunicarse con un clúster (mediante herramientas de administración de Kubernetes como, por ejemplo, `kubectl`). Puede habilitar el acceso privado al servidor de la API de Kubernetes para que toda la comunicación entre los nodos y el servidor de la API permanezca dentro de su VPC. El AWS PrivateLink para la API de Amazon EKS lo ayuda a llamar a las API de Amazon EKS desde su VPC sin exponer el tráfico a la Internet pública.
- No puede configurar Amazon EKS para que solo se pueda acceder a través de un punto de conexión de interfaz.
- Los precios estándar de AWS PrivateLink se aplican a los punto de conexión de interfaz para Amazon EKS. Se le facturará por cada hora de aprovisionamiento de un punto de conexión de interfaz en cada zona de disponibilidad y los datos procesados a través del punto de conexión de interfaz. Para más información, consulte Precios de AWS PrivateLink.
- Las políticas de punto de conexión de VPC no son compatibles con Amazon EKS. De forma predeterminada, el acceso completo a Amazon EKS se permite a través del punto de conexión de interfaz. Como alternativa, puede asociar un grupo de seguridad a las interfaces de red de los puntos de conexión para controlar el tráfico a Amazon EKS a través del punto de conexión de interfaz.
- Puede utilizar los registros de flujo de la VPC para capturar información sobre el tráfico IP entrante y saliente de las interfaces, incluidos los puntos de conexión de interfaz Los datos de los registros de flujo se pueden publicar en Amazon CloudWatch o Amazon S3. Para obtener más información, consulte Registro del tráfico IP con registros de flujo de VPC en la Guía del usuario de Amazon VPC.

- Puede acceder a las API de Amazon EKS desde un centro de datos en las instalaciones si lo conecta a una VPC que tenga un punto de conexión de interfaz. Puede usar AWS Direct Connect o AWS Site-to-Site VPN para conectar sus sitios en las instalaciones a una VPC.
- Puede conectar una VPC a otra con puntos de conexión de interfaz mediante AWS Transit Gateway o interconexión de VPC. El emparejamiento de VPC es una conexión de red entre dos VPC. Puede establecer una conexión de emparejamiento de VPC entre dos VPC propias o con una VPC de otra cuenta. Las VPC pueden estar en diferentes Regiones de AWS. El tráfico entre las VPC interconectadas permanece en la red de AWS. El tráfico no atraviesa la Internet pública. Una puerta de enlace de tránsito es un centro de tránsito de la red que puede utilizar para interconectar sus VPC. El tráfico entre una VPC y una puerta de enlace de tránsito permanece en la red privada global de AWS. El tráfico no se expone a la Internet pública.
- Solo se puede acceder a los puntos de conexión de interfaz de la VPC para Amazon EKS a través de `IPv4`. `IPv6` no es compatible.
- El soporte AWS PrivateLink no está disponible en las Regiones de AWS de Asia Pacífico (Hyderabad), Asia Pacífico (Melbourne), Asia Pacífico (Osaka), Canadá occidental (Calgary), Europa (España), Europa (Zúrich) ni Medio Oriente (Emiratos Árabes Unidos).

Crear de un punto de conexión de interfaz para Amazon EKS

Puede crear un punto de conexión de interfaz para Amazon EKS mediante la consola de Amazon VPC o la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte Create a VPC endpoint (Creación de un punto de conexión de VPC) en la Guía de AWS PrivateLink.

Cree un punto de conexión para Amazon EKS con el siguiente nombre de servicio:

```
com.amazonaws.region-code.eks
```

La característica de DNS privada se habilita de forma predeterminada al crear un punto de conexión de interfaz para Amazon EKS y otros Servicios de AWS. Sin embargo, debe asegurarse de que los siguientes atributos de VPC estén configurados como `true`: `enableDnsHostnames` y `enableDnsSupport`. Para obtener más información, consulte Ver y actualizar los atributos de DNS para una VPC en la Guía del usuario de Amazon VPC. Con la característica de DNS privada habilitada para el punto de conexión de interfaz:

- Puede realizar cualquier solicitud de API a Amazon EKS con su nombre de DNS regional predeterminado. Por ejemplo, `eks.region.amazonaws.com`. Para obtener una lista de API, consulte Acciones en la Referencia de la API de Amazon EKS.

- No necesita realizar ningún cambio en las aplicaciones que llamen a las API de EKS.
- Cualquier llamada que se realice al punto de conexión del servicio predeterminado de Amazon EKS se enruta automáticamente a través del punto de conexión de interfaz a través de la red privada de AWS.

Cargas de trabajo

Sus cargas de trabajo se implementan en contenedores, que se implementan en Pods en Kubernetes. Un Pod incluye uno o más contenedores. Normalmente, uno o más Pods que proporcionan el mismo servicio se implementan en un servicio de Kubernetes. Una vez que haya implementado varios Pods que proporcionan el mismo servicio, puede:

- Visualizar información acerca de las cargas de trabajo que se ejecutan en cada uno de los clústeres mediante la AWS Management Console.
- Escalar verticalmente los Pods hacia arriba o hacia abajo con los Kubernetes Escalador automático vertical de pods.
- Escalar horizontalmente el número de Pods necesarios para satisfacer la demanda hacia arriba o hacia abajo con los Kubernetes Escalador automático de pods horizontales.
- Cree un equilibrador de carga de red externo (para Pods accesibles a Internet) o interno (para Pods privados) a fin de equilibrar el tráfico de red entre los Pods. El equilibrador de carga enruta el tráfico en la capa 4 del modelo OSI.
- Cree una Equilibrio de carga de aplicaciones en Amazon EKS para equilibrar el tráfico de aplicaciones entre los Pods. El equilibrador de carga de aplicaciones enruta el tráfico en la capa 7 del modelo OSI.
- Si es la primera vez que utiliza Kubernetes, este tema lo ayudará Implementar una aplicación de muestra.
- Puede: Restringir las direcciones IP que se pueden asignar a un servicio por `externalIPs`.

Implementar una aplicación de muestra

En esta sección, implementará una aplicación de muestra en un clúster.

Requisitos previos

- Un clúster de Kubernetes existente que tenga como mínimo un nodo. Si no dispone de un clúster de Amazon EKS existente, puede implementar uno mediante una de las guías de Introducción a Amazon EKS. Si va a implementar una aplicación Windows, debe tener Compatibilidad con Windows habilitada para el clúster y al menos un nodo Windows de Amazon EC2.
- `Kubectl` instalado en su equipo. Para obtener más información, consulte Instalación o actualización del `kubectl`.

- `Kubectl` configurado para comunicarse con el clúster. Para obtener más información, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.
- Si planea implementar su carga de trabajo de muestra en Fargate, debe tener una perfil de Fargate que incluya el mismo espacio de nombres creado en este tutorial, que es `eks-sample-app`, a menos que cambie el nombre. Si utilizó una de las guías de introducción para crear el clúster, tendrá que crear un nuevo perfil o agregar el espacio de nombres a su perfil existente, porque el perfil creado en las guías de introducción no especifica el espacio de nombres utilizado en este tutorial. La VPC debe tener también al menos una subred privada.

Para implementar una aplicación de muestra

Aunque muchas variables se pueden cambiar en los siguientes pasos, recomendamos cambiar solo los valores de las variables si se especifican. Una vez que conozca mejor los Pods, las implementaciones y los servicios de Kubernetes, puede experimentar cambiando otros valores.

1. Cree un espacio de nombres de . Un espacio de nombres permite agrupar recursos en Kubernetes. Para obtener más información, consulte Espacios de nombres en la documentación de Kubernetes. Si planea implementar su aplicación de muestra en AWS Fargate, asegúrese de que el valor de `namespace` en su Perfil de AWS Fargate sea `eks-sample-app`.

   ```
   kubectl create namespace eks-sample-app
   ```

2. Cree una implementación de Kubernetes. Esta implementación de muestra extrae una imagen de contenedor de un repositorio público e implementa tres réplicas (Pods individuales) de ella en su clúster. Para obtener más información, consulte Implementaciones en la documentación de Kubernetes. Puede implementar la aplicación en nodos Linux o Windows. Si va a implementar en Fargate, solo puede implementar una aplicación Linux.

   a. Guarde los siguientes contenidos en un archivo llamado `eks-sample-deployment.yaml`. Los contenedores de la aplicación de muestra no utilizan almacenamiento en red, pero es posible que tenga aplicaciones que lo necesiten. Para obtener más información, consulte Almacenamiento.

      Linux

      Los `values` `amd64` o `arm64` en `kubernetes.io/arch` significan que la aplicación se puede implementar en cualquiera de las arquitecturas de hardware (si tiene ambos en el clúster). Esto es posible porque esta imagen es una imagen de arquitectura múltiple,

pero no todas lo son. Puede determinar la arquitectura de hardware con la que se admite la imagen a partir de los detalles de imagen en el repositorio del que lo está sacando. Al implementar imágenes que no admiten un tipo de arquitectura de hardware o en las que no desea que se implemente la imagen, elimine ese tipo del manifiesto. Para obtener más información, consulte Etiquetas, anotaciones y taints conocidas en la documentación de Kubernetes.

El `kubernetes.io/os: linux` `nodeSelector` significa que si tuviera nodos Linux y Windows (por ejemplo) en el clúster, la imagen solo se implementaría en nodos Linux. Para obtener más información, consulte Etiquetas, anotaciones y taints conocidas en la documentación de Kubernetes.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: eks-sample-linux-deployment
  namespace: eks-sample-app
  labels:
    app: eks-sample-linux-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: eks-sample-linux-app
  template:
    metadata:
      labels:
        app: eks-sample-linux-app
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: kubernetes.io/arch
                operator: In
                values:
                - amd64
                - arm64
      containers:
      - name: nginx
        image: public.ecr.aws/nginx/nginx:1.23
```

```
        ports:
        - name: http
          containerPort: 80
        imagePullPolicy: IfNotPresent
      nodeSelector:
        kubernetes.io/os: linux
```

Windows

El `kubernetes.io/os: windows` `nodeSelector` significa que si tuviera nodos Windows y Linux (por ejemplo) en el clúster, la imagen solo se implementaría en nodos Windows. Para obtener más información, consulte Etiquetas, anotaciones y taints conocidas en la documentación de Kubernetes.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: eks-sample-windows-deployment
  namespace: eks-sample-app
  labels:
    app: eks-sample-windows-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: eks-sample-windows-app
  template:
    metadata:
      labels:
        app: eks-sample-windows-app
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: beta.kubernetes.io/arch
                operator: In
                values:
                - amd64
      containers:
      - name: windows-server-iis
        image: mcr.microsoft.com/windows/servercore:ltsc2019
```

```
        ports:
        - name: http
          containerPort: 80
        imagePullPolicy: IfNotPresent
        command:
        - powershell.exe
        - -command
        - "Add-WindowsFeature Web-Server; Invoke-WebRequest -UseBasicParsing
 -Uri 'https://dotnetbinaries.blob.core.windows.net/servicemonitor/2.0.1.6/
ServiceMonitor.exe' -OutFile 'C:\\ServiceMonitor.exe'; echo
 '<html><body><br/><br/><marquee><H1>Hello EKS!!!<H1><marquee></body><html>'
 > C:\\inetpub\\wwwroot\\default.html; C:\\ServiceMonitor.exe 'w3svc'; "
      nodeSelector:
        kubernetes.io/os: windows
```

b. Aplique el manifiesto de implementación al clúster.

```
kubectl apply -f eks-sample-deployment.yaml
```

3. Cree un servicio. Un servicio le permite acceder a todas las réplicas a través de una única dirección IP o nombre. Para obtener más información, consulte Servicio en la documentación de Kubernetes. Aunque no se ha implementado en la aplicación de muestra, si tiene aplicaciones que necesitan interactuar con otros servicios de AWS, recomendamos que cree cuentas de servicio de Kubernetes para sus Pods y las asocie a cuentas de IAM de AWS. Si especifica cuentas de servicio, los Pods tendrán solo los permisos mínimos que usted especifique para que interactúen con otros servicios. Para obtener más información, consulte Roles de IAM para cuentas de servicio.

   a. Guarde el siguiente contenido en un archivo denominado `eks-sample-service.yaml`. Kubernetes asigna al servicio su propia dirección IP a la que se puede acceder solo desde el clúster. Para acceder al servicio desde fuera del clúster, implemente AWS Load Balancer Controller para que equilibre la carga de tráfico de aplicaciones o redes del servicio.

      Linux

```
apiVersion: v1
kind: Service
metadata:
  name: eks-sample-linux-service
  namespace: eks-sample-app
  labels:
```

```
    app: eks-sample-linux-app
spec:
  selector:
    app: eks-sample-linux-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Windows

```
apiVersion: v1
kind: Service
metadata:
  name: eks-sample-windows-service
  namespace: eks-sample-app
  labels:
    app: eks-sample-windows-app
spec:
  selector:
    app: eks-sample-windows-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

b. Aplique el manifiesto de servicio al clúster.

```
kubectl apply -f eks-sample-service.yaml
```

4. Consulte todos los recursos que existen en el espacio de nombres `eks-sample-app`.

```
kubectl get all -n eks-sample-app
```

Un ejemplo de salida sería el siguiente.

Si ha implementado recursos de Windows, todas las instancias de *`linux`* en la siguiente salida aparecerán como `windows`. Los otros *`valores de ejemplo`* pueden ser diferentes a los de la salida.

```
NAME                                                READY   STATUS    RESTARTS   AGE
pod/eks-sample-linux-deployment-65b7669776-m6qxz    1/1     Running   0          27m
```

```
pod/eks-sample-linux-deployment-65b7669776-mmxvd   1/1     Running   0          27m
pod/eks-sample-linux-deployment-65b7669776-qzn22   1/1     Running   0          27m

NAME                                 TYPE         CLUSTER-IP      EXTERNAL-IP
 PORT(S)   AGE
service/eks-sample-linux-service   ClusterIP    10.100.74.8     <none>        80/
TCP    32m

NAME                                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/eks-sample-linux-deployment 3/3     3            3           27m

NAME                                                    DESIRED   CURRENT   READY
   AGE
replicaset.apps/eks-sample-linux-deployment-776d8f8fd8    3         3         3
   27m
```

En la salida, verá el servicio y la implementación que se especificaron en los manifiestos de muestra implementados en pasos anteriores. También verá tres Pods. Esto se debe a que especificó `3` `replicas` en el manifiesto de ejemplo. Para obtener más información acerca de Pods, consulte Pod en la documentación de Kubernetes. Kubernetes crea automáticamente el recurso `replicaset`, aunque no se especifica en los manifiestos de ejemplo. Para obtener más información acerca de `ReplicaSets`, consulte ReplicaSet en la documentación de Kubernetes.

> **Note**
>
> Kubernetes mantiene el número de réplicas que se ha especificado en el manifiesto. Si se trata de una implementación de producción y desea que Kubernetes escale horizontalmente el número de réplicas o verticalmente los recursos informáticos de los Pods, utilice el Escalador automático de pods horizontales y Escalador automático vertical de pods para hacerlo.

5. Consulte los detalles del servicio implementado. Si ha implementado un servicio de Windows, sustituya *`linux`* con **`windows`**.

```
kubectl -n eks-sample-app describe service eks-sample-linux-service
```

Un ejemplo de salida sería el siguiente.

Si ha implementado recursos de Windows, todas las instancias de *linux* en la siguiente salida aparecerán como `windows`. Los otros *valores de ejemplo* pueden ser diferentes a los de la salida.

```
Name:              eks-sample-linux-service
Namespace:         eks-sample-app
Labels:            app=eks-sample-linux-app
Annotations:       <none>
Selector:          app=eks-sample-linux-app
Type:              ClusterIP
IP Families:       <none>
IP:                10.100.74.8
IPs:               10.100.74.8
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         192.168.24.212:80,192.168.50.185:80,192.168.63.93:80
Session Affinity:  None
Events:            <none>
```

En la salida anterior, el valor de `IP:` es una dirección IP única a la que se puede acceder desde cualquier nodo o Pod dentro del clúster, pero a la cual no se puede acceder desde fuera del clúster. Los valores de `Endpoints` son direcciones IP asignadas desde la VPC a los Pods que forman parte del servicio.

6. Vea los detalles de uno de los Pods que aparecían en la salida cuando vio el espacio de nombres en un paso anterior. Si implementó una aplicación de Windows, reemplace *linux* por **windows** y *776d8f8fd8-78w66* por el valor devuelto para uno de sus Pods.

```
kubectl -n eks-sample-app describe pod eks-sample-linux-deployment-65b7669776-m6qxz
```

Salida abreviada

Si ha implementado recursos de Windows, todas las instancias de *linux* en la siguiente salida aparecerán como `windows`. Los otros *example values* pueden ser diferentes a los de la salida.

```
Name:         eks-sample-linux-deployment-65b7669776-m6qxz
Namespace:    eks-sample-app
Priority:     0
Node:         ip-192-168-45-132.us-west-2.compute.internal/192.168.45.132
```

```
[...]
IP:           192.168.63.93
IPs:
  IP:           192.168.63.93
Controlled By:  ReplicaSet/eks-sample-linux-deployment-65b7669776
[...]
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
[...]
Events:
  Type    Reason     Age    From
 Message
  ----    ------     ----   ----
 -------
  Normal  Scheduled  3m20s  default-scheduler
 Successfully assigned eks-sample-app/eks-sample-linux-deployment-65b7669776-m6qxz
 to ip-192-168-45-132.us-west-2.compute.internal
[...]
```

En la salida anterior, el valor de `IP:` es una IP única que se asigna al Pod desde el bloque de CIDR asignado a la subred en la que se encuentra el nodo. Si prefiere asignar a los Pods direcciones IP de bloques de CIDR distintos, puede cambiar el comportamiento predeterminado. Para obtener más información, consulte Redes personalizadas para los pods. También puede ver que el programador de Kubernetes programó el Pod en el `Node` con la dirección IP *192.168.45.132*.

> **Tip**
>
> En lugar de utilizar la línea de comandos, puede ver muchos detalles sobre los Pods, los servicios, las implementaciones y otros recursos de Kubernetes en la AWS Management Console. Para obtener más información, consulte Vea los recursos de Kubernetes.

7. Ejecute un shell en el Pod que describió en el paso anterior y reemplace *65b7669776-m6qxz* por el ID de uno de sus Pods.

Linux

```
kubectl exec -it eks-sample-linux-deployment-65b7669776-m6qxz -n eks-sample-app
 -- /bin/bash
```

Windows

```
kubectl exec -it eks-sample-windows-deployment-65b7669776-m6qxz -n eks-sample-
app -- powershell.exe
```

8. Desde el shell del Pod, vea la salida del servidor web que se instaló con la implementación en un paso anterior. Solo tiene que especificar el nombre del servicio. CoreDNS lo resuelve en la dirección IP del servicio, que se implementa con un clúster de Amazon EKS de forma predeterminada.

   Linux

   ```
   curl eks-sample-linux-service
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   <!DOCTYPE html>
   <html>
   <head>
   <title>Welcome to nginx!</title>
   [...]
   ```

   Windows

   ```
   Invoke-WebRequest -uri eks-sample-windows-service/default.html -UseBasicParsing
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   StatusCode        : 200
   StatusDescription : OK
   Content           : < h t m l > < b o d y > < b r / > < b r / > < m a r q u e e
    > < H 1 > H e l l o
   ```

```
                    E K S ! ! ! < H 1 > < m a r q u e e > < / b o d y > < h t
m l >
```

9. Desde el shell del Pod, vea el servidor de DNS del Pod.

   Linux

   ```
   cat /etc/resolv.conf
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   nameserver 10.100.0.10
   search eks-sample-app.svc.cluster.local svc.cluster.local cluster.local us-
   west-2.compute.internal
   options ndots:5
   ```

   En la salida anterior, `10.100.0.10` se asigna automáticamente como el `nameserver` para todos los Pods implementados en el clúster.

   Windows

   ```
   Get-NetIPConfiguration
   ```

   Salida abreviada

   ```
   InterfaceAlias       : vEthernet
   [...]
   IPv4Address          : 192.168.63.14
   [...]
   DNSServer            : 10.100.0.10
   ```

   En la salida anterior, `10.100.0.10` se asigna automáticamente como el servidor de DNS para todos los Pods implementados en el clúster.

10. Puede desconectarse del Pod al escribir `exit`.

11. Una vez que haya terminado de utilizar la aplicación de muestra, puede eliminar el espacio de nombres, el servicio y la implementación de muestra con el siguiente comando.

    ```
    kubectl delete namespace eks-sample-app
    ```

Siguientes pasos

Después de implementar la aplicación de ejemplo, es posible que desee intentar alguno de los siguientes ejercicios:

- the section called “Equilibrio de carga de aplicaciones”
- the section called “Equilibrio de carga de red”

Escalador automático vertical de pods

El escalador automático vertical de pods de Kubernetes ajusta de forma automática las reservas de CPU y memoria de sus Pods para ayudar a “ajustar el tamaño” de las aplicaciones. Este ajuste puede mejorar la utilización de los recursos del clúster y liberar CPU y memoria para otros Pods. Este tema lo ayuda a implementar el escalador automático vertical de pods en el clúster y a verificar que funciona.

Requisitos previos

- Tiene un clúster de Amazon EKS existente. Si no lo tiene, consulte Introducción a Amazon EKS.
- Tiene instalado el servidor de métricas de Kubernetes. Para obtener más información, consulte Instalación del servidor de métricas de Kubernetes.
- Utiliza un cliente `kubectl` que está configurado para comunicarse con el clúster de Amazon EKS.
- OpenSSL `1.1.1` o posterior instalado en su dispositivo.

Implementar el escalador automático vertical de pods

En esta sección, implementará el escalador automático vertical de pods en el clúster.

Para implementar el escalador automático vertical de pods

1. Abra una ventana de terminal y vaya al directorio en el que desee descargar el código fuente del escalador automático vertical de pods.
2. Clone el repositorio kubernetes/autoscaler de GitHub.

   ```
   git clone https://github.com/kubernetes/autoscaler.git
   ```

3. Cambie al directorio de `vertical-pod-autoscaler`.

```
cd autoscaler/vertical-pod-autoscaler/
```

4. (Opcional) Si ya ha implementado otra versión del escalador automático vertical de pods, elimínela con el siguiente comando.

```
./hack/vpa-down.sh
```

5. Si los nodos no tienen acceso a Internet al registro de contenedores `registry.k8s.io`, entonces debe extraer las siguientes imágenes y enviarlas a su propio repositorio privado. Para obtener más información sobre cómo extraer y enviar las imágenes en su propio repositorio privado, consulte Copiar una imagen de contenedor de un repositorio en otro repositorio.

```
registry.k8s.io/autoscaling/vpa-admission-controller:0.10.0
registry.k8s.io/autoscaling/vpa-recommender:0.10.0
registry.k8s.io/autoscaling/vpa-updater:0.10.0
```

   Si va a enviar las imágenes a un repositorio privado de Amazon ECR, sustituya `registry.k8s.io` en los manifiestos por su registro. Reemplace *`111122223333`* por su ID de cuenta. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster. El siguiente comando supone que el nombre del repositorio privado es el mismo que el repositorio de origen en el manifiesto. Si le ha dado un nombre diferente a su repositorio, tendrá que cambiarlo también.

```
sed -i.bak -e 's/registry.k8s.io/111122223333.dkr.ecr.region-
code.amazonaws.com/' ./deploy/admission-controller-deployment.yaml
sed -i.bak -e 's/registry.k8s.io/111122223333.dkr.ecr.region-
code.amazonaws.com/' ./deploy/recommender-deployment.yaml
sed -i.bak -e 's/registry.k8s.io/111122223333.dkr.ecr.region-
code.amazonaws.com/' ./deploy/updater-deployment.yaml
```

6. Implemente el escalador automático vertical de pods en el clúster con el siguiente comando.

```
./hack/vpa-up.sh
```

7. Verifique que se hayan creado correctamente los Pods del escalador automático vertical de pods.

```
kubectl get pods -n kube-system
```

Un ejemplo de salida sería el siguiente.

```
NAME                                        READY   STATUS    RESTARTS   AGE
[...]
metrics-server-8459fc497-kfj8w              1/1     Running   0          83m
vpa-admission-controller-68c748777d-ppspd   1/1     Running   0          7s
vpa-recommender-6fc8c67d85-gljpl            1/1     Running   0          8s
vpa-updater-786b96955c-bgp9d                1/1     Running   0          8s
```

Comprobar la instalación del escalador automático vertical de pods

En esta sección, implementará una aplicación de ejemplo para verificar que el escalador automático vertical de pods funciona.

Para probar la instalación del escalador automático vertical de pods

1. Implemente el ejemplo del escalador automático vertical de pods de `hamster.yaml` con el siguiente comando.

```
kubectl apply -f examples/hamster.yaml
```

2. Obtenga los Pods de la aplicación de ejemplo de `hamster`.

```
kubectl get pods -l app=hamster
```

Un ejemplo de salida sería el siguiente.

```
hamster-c7d89d6db-rglf5   1/1     Running   0          48s
hamster-c7d89d6db-znvz5   1/1     Running   0          48s
```

3. Describa uno de los Pods para ver la reserva de `cpu` y `memory`. Reemplace *c7d89d6db-rglf5* por uno de los ID obtenidos en la salida del paso anterior.

```
kubectl describe pod hamster-c7d89d6db-rglf5
```

Un ejemplo de salida sería el siguiente.

```
[...]
```

```
Containers:
  hamster:
    Container ID:  docker://
e76c2413fc720ac395c33b64588c82094fc8e5d590e373d5f818f3978f577e24
    Image:         registry.k8s.io/ubuntu-slim:0.1
    Image ID:      docker-pullable://registry.k8s.io/ubuntu-
slim@sha256:b6f8c3885f5880a4f1a7cf717c07242eb4858fdd5a84b5ffe35b1cf680ea17b1
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/sh
    Args:
      -c
      while true; do timeout 0.5s yes >/dev/null; sleep 0.5s; done
    State:          Running
      Started:      Fri, 27 Sep 2019 10:35:16 -0700
    Ready:          True
    Restart Count:  0
    Requests:
      cpu:        100m
      memory:     50Mi
[...]
```

Puede ver que el Pod original reserva 100 milicpu de CPU y 50 mebibytes de memoria. En esta aplicación de ejemplo, 100 milicpu es menos de lo que necesita el Pod para ejecutarse, por lo que está limitada a la CPU. También reserva mucha menos memoria de la que necesita. La implementación del escalador automático vertical de `vpa-recommender` analiza los `hamster` de Pods para ver si los requisitos de CPU y memoria son adecuados. Si se necesita realizar ajustes, el `vpa-updater` vuelve a lanzar los Pods con valores actualizados.

4. Espere a que `vpa-updater` lance un nuevo Pod de `hamster`. Esto debería tardar uno o dos minutos. Puede monitorear los Pods con el siguiente comando.

> **Note**
>
> Si no está seguro de que se haya lanzado un nuevo Pod, compare los nombres de los Pod con la lista anterior. Cuando se lance el nuevo Pod, verá un nuevo nombre de Pod.

```
kubectl get --watch Pods -l app=hamster
```

5. Cuando comience un nuevo Pod de `hamster`, descríbalo y vea las reservas de CPU y memoria actualizadas.

```
kubectl describe pod hamster-c7d89d6db-jxgfv
```

Un ejemplo de salida sería el siguiente.

```
[...]
Containers:
  hamster:
    Container ID:
 docker://2c3e7b6fb7ce0d8c86444334df654af6fb3fc88aad4c5d710eac3b1e7c58f7db
    Image:         registry.k8s.io/ubuntu-slim:0.1
    Image ID:      docker-pullable://registry.k8s.io/ubuntu-
slim@sha256:b6f8c3885f5880a4f1a7cf717c07242eb4858fdd5a84b5ffe35b1cf680ea17b1
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/sh
    Args:
      -c
      while true; do timeout 0.5s yes >/dev/null; sleep 0.5s; done
    State:          Running
      Started:      Fri, 27 Sep 2019 10:37:08 -0700
    Ready:          True
    Restart Count:  0
    Requests:
      cpu:        587m
      memory:     262144k
[...]
```

En la salida anterior puede ver que la reserva de `cpu` ha aumentado a 587 milicpu, que es más de cinco veces el valor original. La `memory` ha aumentado a 262 144 kilobytes, lo que equivale a 250 mebibytes, o a cinco veces el valor original. Este Pod no tenía recursos suficientes, y el escalador automático vertical de pods corrigió la estimación con un valor mucho más adecuado.

6. Describa el recurso de `hamster-vpa` para ver la nueva recomendación.

```
kubectl describe vpa/hamster-vpa
```

Un ejemplo de salida sería el siguiente.

```
Name:         hamster-vpa
Namespace:    default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"autoscaling.k8s.io/
v1beta2","kind":"VerticalPodAutoscaler","metadata":{"annotations":
{},"name":"hamster-vpa","namespace":"d...
API Version:  autoscaling.k8s.io/v1beta2
Kind:         VerticalPodAutoscaler
Metadata:
  Creation Timestamp:  2019-09-27T18:22:51Z
  Generation:          23
  Resource Version:    14411
  Self Link:           /apis/autoscaling.k8s.io/v1beta2/namespaces/default/
verticalpodautoscalers/hamster-vpa
  UID:                 d0d85fb9-e153-11e9-ae53-0205785d75b0
Spec:
  Target Ref:
    API Version:  apps/v1
    Kind:         Deployment
    Name:         hamster
Status:
  Conditions:
    Last Transition Time:  2019-09-27T18:23:28Z
    Status:                True
    Type:                  RecommendationProvided
  Recommendation:
    Container Recommendations:
      Container Name:  hamster
      Lower Bound:
        Cpu:     550m
        Memory:  262144k
      Target:
        Cpu:     587m
        Memory:  262144k
      Uncapped Target:
        Cpu:     587m
        Memory:  262144k
      Upper Bound:
        Cpu:     21147m
        Memory:  387863636
Events:          <none>
```

7. Cuando termine de experimentar con la aplicación de ejemplo, elimínela con el siguiente comando.

```
kubectl delete -f examples/hamster.yaml
```

Escalador automático de pods horizontales

El escalador automático de pods horizontales de Kubernetes escala automáticamente el número de Pods en una implementación, un controlador de reproducción o un conjunto de réplicas en función de la utilización de la CPU de ese recurso. Esto puede ayudar a las aplicaciones a escalar horizontalmente para satisfacer el aumento de la demanda o a reducir horizontalmente cuando no se necesitan recursos y, de esa forma, liberar los nodos para otras aplicaciones. Cuando se establece un porcentaje de utilización de CPU objetivo, el escalador automático de pods horizontales escala su aplicación de forma horizontal o la reduce horizontalmente para intentar alcanzar ese objetivo.

El Horizontal Pod Autoscaler es un recurso de la API estándar en Kubernetes que simplemente requiere que un origen de métricas (como el servidor de métricas de Kubernetes) esté instalada en el clúster de Amazon EKS para funcionar. No es necesario implementar ni instalar el Horizontal Pod Autoscaler en el clúster para comenzar a escalar las aplicaciones. Para obtener más información, consulte Horizontal Pod Autoscaler en la documentación del Kubernetes.

Utilice este tema para preparar el Horizontal Pod Autoscaler para el clúster de Amazon EKS y para verificar que funciona con una aplicación de muestra.

> **Note**
>
> Este tema se basa en la explicación de Horizontal Pod autoscaler en la documentación de Kubernetes.

Requisitos previos

- Tiene un clúster de Amazon EKS existente. Si no lo tiene, consulte Introducción a Amazon EKS.
- Tiene instalado el servidor de métricas de Kubernetes. Para obtener más información, consulte Instalación del servidor de métricas de Kubernetes.
- Utiliza un cliente `kubectl` que está configurado para comunicarse con el clúster de Amazon EKS.

Ejecutar una aplicación de prueba del escalador automático de pods horizontales

En esta sección, implementará una aplicación de muestra para verificar que el escalador automático de pods horizontales funciona.

> **Note**
>
> Este ejemplo se basa en la explicación del escalador automático del Pod horizontal en la documentación de Kubernetes.

Para probar la instalación del escalador automático de pods horizontales

1. Implemente una aplicación de servidor web Apache sencilla con el siguiente comando.

   ```
   kubectl apply -f https://k8s.io/examples/application/php-apache.yaml
   ```

   Este Pod de servidor web Apache tiene un límite de CPU 500 milicpu y sirve en el puerto 80.
2. Cree un recurso del escalador automático de pods horizontales para la implementación de `php-apache`.

   ```
   kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
   ```

   Este comando crea un escalador automático que tiene como objetivo el 50 por ciento de utilización de la CPU para la implementación, con un mínimo de un Pod y un máximo de diez Pods. Cuando la carga media de CPU es inferior al 50 %, el escalador automático intenta reducir el número de Pods en la implementación a un mínimo de uno. Cuando la carga es superior al 50 por ciento, el escalador automático intenta aumentar el número de Pods en la implementación, hasta un máximo de diez. Para obtener más información, consulte How does the HorizontalPodAutoscaler work? (¿Cómo funciona el escalador automático de pods horizontales?) en la documentación de Kubernetes.
3. Describa el escalador automático con el siguiente comando para ver los detalles.

   ```
   kubectl get hpa
   ```

   Un ejemplo de salida sería el siguiente.

```
NAME         REFERENCE                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache    0%/50%    1         10        1          51s
```

Como puede ver, la carga actual de la CPU es `0%`, porque todavía no hay carga en el servidor. El recuento de Pod ya se encuentra en su límite más bajo (uno), por lo que no se puede reducir horizontalmente.

4. Cree una carga para el servidor web mediante la ejecución un contenedor.

```
kubectl run -i \
    --tty load-generator \
    --rm --image=busybox \
    --restart=Never \
    -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"
```

5. Para ver que la implementación escale horizontalmente, ejecute de manera periódica el siguiente comando en un terminal independiente desde el terminal en el que ejecutó el paso anterior.

```
kubectl get hpa php-apache
```

Un ejemplo de salida sería el siguiente.

```
NAME         REFERENCE                TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache    250%/50%   1         10        5
 4m44s
```

El recuento de réplicas puede tardar más de un minuto en aumentar. Mientras el porcentaje actual de la CPU sea superior al porcentaje objetivo, el recuerdo de réplicas aumenta hasta 10. En este caso, es `250%`, para que el número de `REPLICAS` siga en aumento.

> **Note**
>
> Pueden pasar unos minutos antes de que vea que el recuento de réplicas alcanza su máximo. Si solo se necesitan 6 réplicas, por ejemplo, para que la carga de la CPU permanezca al o por debajo del 50 %, la carga no superará las 6 réplicas.

6. Detenga la carga. En la ventana del terminal en la que genera la carga, mantenga presionadas las teclas `Ctrl+C` para detener la carga. Puede ver cómo las réplicas vuelven a escalar a 1 al ejecutar de nuevo el siguiente comando en el terminal en el que ve la reducción horizontal.

   ```
   kubectl get hpa
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME         REFERENCE                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
   php-apache   Deployment/php-apache    0%/50%    1         10        1          25m
   ```

   > **Note**
   >
   > El periodo predeterminado para reducir la escala es de cinco minutos, por lo que pasará algún tiempo antes de que vea que el recuento de réplicas es de 1 de nuevo, incluso cuando el porcentaje actual de la CPU es 0 %. El periodo de tiempo es modificable. Para obtener más información, consulte Horizontal Pod Autoscaler (Escalador automático de pods horizontales) en la documentación de Kubernetes.

7. Cuando haya terminado de experimentar con la aplicación de muestra, elimine los recursos `php-apache`.

   ```
   kubectl delete deployment.apps/php-apache service/php-apache
    horizontalpodautoscaler.autoscaling/php-apache
   ```

Equilibrio de carga de red en Amazon EKS

Se equilibra la carga del tráfico de red en `L4` del modelo OSI. Para equilibrar la carga del tráfico de aplicaciones en `L7`, implemente una `ingress` de Kubernetes, que aprovisiona un equilibrador de carga de aplicación de AWS. Para obtener más información, consulte Equilibrio de carga de aplicaciones en Amazon EKS. Para obtener más información sobre las diferencias entre los dos tipos de equilibrio de carga, consulte Características de Elastic Load Balancing en el sitio web de AWS.

Cuando crea un `Service` de Kubernetes de tipo `LoadBalancer`, el controlador del equilibrador de carga del proveedor de nube de AWS crea equilibradores de carga clásicos de AWS de forma predeterminada, pero también puede crear equilibradores de carga de red de AWS. Este controlador solo recibirá correcciones de errores críticos en el futuro. Para obtener más información acerca del

uso del equilibrador de carga del proveedor de nube de AWS, consulte controlador del equilibrador de carga del proveedor de nube de AWS en la documentación de Kubernetes. Su uso no se aborda en este tema.

Le recomendamos usar la versión `2.7.2` o una posterior del AWS Load Balancer Controller en lugar del controlador del equilibrador de carga del proveedor de la nube de AWS. El AWS Load Balancer Controller crea equilibradores de carga de red de AWS, pero no crea equilibradores de carga clásica de AWS. El resto de este tema trata sobre el uso del Controlador del equilibrador de carga de AWS.

Un equilibrador de carga de red de AWS puede equilibrar la carga del tráfico de red hacia los Pods implementados en destinos IP e instancias de Amazon EC2 o hacia destinos IP de AWS Fargate. Para obtener más información, consulte Controlador del equilibrador de carga de AWS en GitHub.

Requisitos previos

Antes de poder equilibrar la carga del tráfico de red con el AWS Load Balancer Controller, debe cumplir los siguientes requisitos.

- Tener un clúster existente. Si no tiene un clúster existente, consulte Introducción a Amazon EKS. Si necesita actualizar la versión de un clúster existente, consulte Actualización de una versión de Kubernetes de clúster de Amazon EKS.
- Implemente AWS Load Balancer Controller en su clúster. Para obtener más información, consulte ¿Qué es el AWS Load Balancer Controller?. Recomendamos la versión `2.7.2` o posterior.
- Tener al menos una subred. Si varias subredes etiquetadas se encuentran en una zona de disponibilidad, el controlador elige la primera subred cuyo ID de subred va primero lexicográficamente. La subred debe tener al menos ocho direcciones IP disponibles.
- Si utiliza la versión `2.1.1` del AWS Load Balancer Controller o anteriores, las subredes deben etiquetarse de la siguiente manera. Si utiliza la versión `2.1.2` o posterior, esta etiqueta es opcional. Es posible que desee etiquetar una subred si tiene varios clústeres que se ejecutan en la misma VPC o varios servicios de AWS que comparten subredes en una VPC y desea tener más control sobre dónde se aprovisionan los equilibradores de carga para cada clúster. Si especifica de manera explícita los ID de subred como una anotación en un objeto de servicio, Kubernetes y el AWS Load Balancer Controller utilizarán esas subredes directamente para crear el equilibrador de carga. El etiquetado de subred no es necesario si elige utilizar este método para aprovisionar los equilibradores de carga y puede omitir los siguientes requisitos de etiquetado de subred privada y pública. Sustituya *`my-cluster`* con el nombre del clúster.
  - Clave: `kubernetes.io/cluster/`*`my-cluster`*

- Valor: `shared` o `owned`

- Las subredes públicas y privadas deben cumplir los siguientes requisitos, a menos que especifique de manera explícita los ID de subred como una anotación en un objeto de servicio o entrada. Si aprovisiona equilibradores de carga mediante la especificación explícita de los ID de subred como una anotación en un objeto de servicio o entrada, Kubernetes y el AWS Load Balancer Controller utilizarán esas subredes directamente para crear el equilibrador de carga, y las siguientes etiquetas no serán necesarias.
  - Subredes privadas: deben etiquetarse con el siguiente formato. De esta manera Kubernetes y el controlador del equilibrador de carga de AWS saben que las subredes se pueden utilizar para equilibradores de carga internos. Si utiliza `eksctl` o una plantilla de AWS AWS CloudFormation de Amazon EKS para crear la VPC después del 26 de marzo de 2020, las subredes se etiquetan correctamente cuando se crean. Para obtener más información sobre las plantillas de VPC de AWS AWS CloudFormation de Amazon EKS, consulte Creación de una VPC para su clúster de Amazon EKS.
    - Clave: `kubernetes.io/role/internal-elb`
    - Valor: `1`
  - Subredes públicas: deben etiquetarse con el siguiente formato. Es así para que Kubernetes sepa que debe utilizar solo esas subredes para los equilibradores de carga externos, en lugar de elegir una subred pública en cada zona de disponibilidad (en orden lexicográfico por ID de subred). Si utiliza `eksctl` o una plantilla de AWS CloudFormation de Amazon EKS para crear la VPC después del 26 de marzo de 2020, las subredes se etiquetan correctamente cuando se crean. Para obtener más información sobre las plantillas de VPC de AWS CloudFormation de Amazon EKS, consulte Creación de una VPC para su clúster de Amazon EKS.
    - Clave: `kubernetes.io/role/elb`
    - Valor: `1`

  Si las etiquetas del rol de subred no se agregan de manera explícita, el controlador de servicio de Kubernetes examinará la tabla de enrutamiento de las subredes de la VPC del clúster para determinar si la subred es privada o pública. Recomendamos que no dependa de este comportamiento y que, en su lugar, agregue de manera explícita las etiquetas de rol públicas o privadas. El AWS Load Balancer Controller no examina las tablas de enrutamiento y requiere que las etiquetas privadas y públicas estén presentes para la detección automática correcta.

Consideraciones

- La configuración del equilibradores de carga se controla mediante anotaciones que se agregan al manifiesto del servicio. Las anotaciones de servicio son diferentes cuando se utiliza el AWS Load Balancer Controller en comparación a cuando se utiliza el controlador del equilibrador de carga del proveedor de nube de AWS. Asegúrese de revisar las anotaciones para el AWS Load Balancer Controller antes de implementar los servicios.
- Cuando se utiliza el Amazon VPC CNI plugin for Kubernetes, el AWS Load Balancer Controller puede equilibrar la carga hacia destinos IP o de instancia de Amazon EC2 y destinos IP de Fargate. Al utilizar complementos de CNI compatibles alternativos, el controlador solo puede equilibrar la carga en los destinos de instancia. Para obtener más información acerca de los tipos de destinos de Network Load Balancer, consulte Tipo de destino en la Guía del usuario para Network Load Balancer.
- Si desea agregar etiquetas al equilibrador de carga durante su creación o después de ella, agregue la siguiente anotación en la especificación del servicio. Para obtener más información, consulte Etiquetas de recursos de AWS en la documentación del AWS Load Balancer Controller.

```
service.beta.kubernetes.io/aws-load-balancer-additional-resource-tags
```

- Para asignar direcciones IP elásticas al Network Load Balancer, agregue la siguiente anotación. Sustituya los *example values* con los `Allocation IDs` de las direcciones IP elásticas. El número de `Allocation IDs` debe coincidir con el número de subredes utilizadas para el equilibrador de carga. Para obtener más información, consulte la documentación de AWS Load Balancer Controller.

```
service.beta.kubernetes.io/aws-load-balancer-eip-allocations:
 eipalloc-xxxxxxxxxxxxxxxxx,eipalloc-yyyyyyyyyyyyyyyyy
```

- Amazon EKS agrega una regla de entrada al grupo de seguridad del nodo para el tráfico del cliente y una regla para cada subred del equilibrador de carga de la VPC a fin de realizar comprobaciones de estado para cada equilibrador de carga de red que cree. La implementación de un servicio de tipo `LoadBalancer` puede fallar si Amazon EKS intenta crear reglas que superen la cuota del número máximo de reglas permitidas para un grupo de seguridad. Para obtener más información, consulte Grupos de seguridad en las cuotas de Amazon VPC en la Guía del usuario de Amazon VPC. Considere las siguientes opciones a fin de minimizar las posibilidades de exceder el número máximo de reglas para un grupo de seguridad:

  - Solicite un aumento de las reglas por cuota de grupo de seguridad. Para obtener más información, consulte Solicitar un aumento de cuota en la Guía del usuario de Service Quotas.
  - Utilice destinos de IP, en lugar de destinos de instancia. Con los destinos de IP, puede compartir reglas para los mismos puertos de destino. Puede especificar manualmente las subredes del equilibrador de carga con una anotación. Para obtener más información, consulte Anotaciones en GitHub.
  - Utilice una entrada en lugar de un servicio de tipo `LoadBalancer` para enviar tráfico al servicio. El AWS Application Load Balancer requiere menos reglas que los Network Load Balancers. Puede compartir un ALB entre varias entradas. Para obtener más información, consulte Equilibrio de carga de aplicaciones en Amazon EKS. No puede compartir un Network Load Balancer entre varios servicios.
  - Implemente sus clústeres en varias cuentas.
- Si los Pods se ejecutan en Windows en un clúster de Amazon EKS, un único servicio con un equilibrador de carga puede admitir hasta 1024 Pods de backend. Cada Pod tiene su propia dirección IP única.
- Recomendamos crear solo equilibradores de carga de red nuevos con el AWS Load Balancer Controller. Intentar sustituir los Network Load Balancers existentes creados con el Controlador del equilibrador de carga del proveedor de nube de AWS puede dar lugar a varios Network Load Balancers que podrían causar tiempo de inactividad en la aplicación.

Crear un equilibrador de carga de red

Puede crear un equilibrador de carga de red con IP o destinos de instancia.

IP targets

Puede utilizar destinos de IP con Pods implementados en nodos de Amazon EC2 o Fargate. El servicio de Kubernetes debe crearse como tipo `LoadBalancer`. Para obtener más información, consulte Tipo de equilibradores de carga en la documentación de Kubernetes.

Para crear un equilibrador de carga que utilice destinos IP, agregue las siguientes anotaciones a un manifiesto de servicio e implemente el servicio. El valor `external` para `aws-load-balancer-type` es lo que lleva al AWS Load Balancer Controller, en lugar del controlador del equilibrador de carga del proveedor de nube de AWS, a crear el equilibrador de carga de red. Puede ver un manifiesto de servicio de ejemplo con los comentarios.

```
service.beta.kubernetes.io/aws-load-balancer-type: "external"
```

```
service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "ip"
```

> **Note**
>
> Si equilibra la carga en Pods de IPv6, agregue la siguiente anotación. Solo puede equilibrar la carga a través de IPv6 a destinos IP, no a destinos de instancias. Sin esta anotación, la tarea de equilibrar la carga se realiza a través de IPv4.
>
> ```
> service.beta.kubernetes.io/aws-load-balancer-ip-address-type: dualstack
> ```

Los Network Load Balancers se crean con el `internal` `aws-load-balancer-scheme` de forma predeterminada. Puede lanzar equilibradores de carga de red en cualquier subred de la VPC del clúster, incluidas las subredes sin especificar cuando creó el clúster.

Kubernetes examina la tabla de enrutamiento de las subredes con el fin de identificar si son públicas o privadas. Las subredes públicas tienen una ruta directa a Internet mediante una puerta de enlace de Internet, pero no así las subredes privadas.

Si desea crear un Network Load Balancer en una subred pública para equilibrar la carga en nodos de Amazon EC2 (Fargate solo puede ser privado), especifique `internet-facing` con la siguiente anotación:

```
service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
```

> **Note**
>
> La anotación `service.beta.kubernetes.io/aws-load-balancer-type: "nlb-ip"` todavía se admite para la compatibilidad con versiones anteriores. Sin embargo, le recomendamos que utilice las anotaciones anteriores para nuevos equilibradores de carga en lugar de `service.beta.kubernetes.io/aws-load-balancer-type: "nlb-ip"`.

> **Important**
>
> No edite los comentarios después de crear el servicio. Si necesita modificarlo, elimine el objeto de servicio y vuelva a crearlo con el valor deseado para este comentario.

Instance targets

El Controlador del equilibrador de carga del proveedor de nube de AWS crea Network Load Balancers solo con destinos de instancia. La versión `2.2.0` y posteriores del Controlador del equilibrador de carga de AWS también crean equilibradores de carga de red con destinos de instancia. Recomendamos utilizarlo, en lugar de utilizar el Controlador del equilibrador de carga del proveedor de nube de AWS, para crear nuevos Network Load Balancers. Puede utilizar destinos de instancia del equilibrador de carga de red con Pods implementados en nodos de Amazon EC2, pero no en Fargate. Para equilibrar la carga del tráfico de red a través de los Pods implementados en Fargate, debe utilizar destinos de IP.

Para implementar un Network Load Balancer en una subred privada, la especificación del servicio debe tener las siguientes anotaciones. Puede ver un manifiesto de servicio de ejemplo con los comentarios. El valor `external` para `aws-load-balancer-type` es lo que lleva al Controlador del equilibrador de carga de AWS, en lugar del controlador del equilibrador de carga del proveedor de nube de AWS, a crear el Network Load Balancer.

```
service.beta.kubernetes.io/aws-load-balancer-type: "external"
service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "instance"
```

Los Network Load Balancers se crean con el `internal` `aws-load-balancer-scheme` de forma predeterminada. Para los equilibradores de carga de red internos, el clúster de Amazon EKS debe estar configurado para utilizar al menos una subred privada en la VPC. Kubernetes examina la tabla de enrutamiento de las subredes con el fin de identificar si son públicas o privadas. Las subredes públicas tienen una ruta directa a Internet mediante una puerta de enlace de Internet, pero no así las subredes privadas.

Si desea crear un Network Load Balancer en una subred pública para equilibrar la carga en nodos de Amazon EC2, especifique `internet-facing` con la siguiente anotación:

```
service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
```

> **Important**
>
> No edite los comentarios después de crear el servicio. Si necesita modificarlo, elimine el objeto de servicio y vuelva a crearlo con el valor deseado para este comentario.

(Opcional) Implementación de una aplicación de muestra

Requisitos previos

- Al menos una subred privada o pública en la VPC del clúster.
- Implemente AWS Load Balancer Controller en su clúster. Para obtener más información, consulte ¿Qué es el AWS Load Balancer Controller?. Recomendamos la versión `2.7.2` o posterior.

Para implementar una aplicación de muestra

1. Si va a implementar en Fargate, asegúrese de tener una subred privada disponible en la VPC y cree un perfil de Fargate. Si no implementa en Fargate, omita este paso. Puede crear el perfil con la ejecución del siguiente comando o en la AWS Management Console con los mismos valores para `name` y `namespace` que los del comando. Reemplace los *`example values`* por los de su propiedad.

   ```
   eksctl create fargateprofile \
       --cluster my-cluster \
       --region region-code \
       --name nlb-sample-app \
       --namespace nlb-sample-app
   ```

2. Implemente una aplicación de muestra.

   a. Cree un espacio de nombres para la aplicación.

      ```
      kubectl create namespace nlb-sample-app
      ```

   b. Guarde el siguiente contenido en un archivo denominado *`sample-deployment`*`.yaml` en el equipo.

      ```
      apiVersion: apps/v1
      kind: Deployment
      ```

```
metadata:
  name: nlb-sample-app
  namespace: nlb-sample-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: public.ecr.aws/nginx/nginx:1.23
          ports:
            - name: tcp
              containerPort: 80
```

c. Aplique el manifiesto al clúster.

```
kubectl apply -f sample-deployment.yaml
```

3. Cree un servicio con un equilibrador de carga de red interno que equilibre la carga en destinos IP.

a. Guarde el siguiente contenido en un archivo denominado *sample-service*.yaml en el equipo. Si va a implementar en nodos Fargate, elimine la línea `service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing`.

```
apiVersion: v1
kind: Service
metadata:
  name: nlb-sample-service
  namespace: nlb-sample-app
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: external
    service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: ip
    service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing
spec:
  ports:
```

```
    - port: 80
      targetPort: 80
      protocol: TCP
  type: LoadBalancer
  selector:
    app: nginx
```

b. Aplique el manifiesto al clúster.

```
kubectl apply -f sample-service.yaml
```

4. Compruebe que el servicio se haya implementado.

```
kubectl get svc nlb-sample-service -n nlb-sample-app
```

Un ejemplo de salida sería el siguiente.

```
NAME             TYPE           CLUSTER-IP         EXTERNAL-IP
                                                  PORT(S)        AGE
sample-service   LoadBalancer   10.100.240.137
 k8s-nlbsampl-nlbsampl-xxxxxxxxxx-xxxxxxxxxxxxxxxx.elb.region-code.amazonaws.com
  80:32400/TCP   16h
```

> **Note**
>
> Los valores de *10.100.240.137* y *xxxxxxxxxx-xxxxxxxxxxxxxxxx* serán diferentes a la salida de ejemplo (serán exclusivos de su equilibrador de carga) y *us-west-2* puede ser diferente para usted según la Región de AWS en la que se encuentre el clúster.

5. Abra la AWS Management Console de Amazon EC2. Seleccione Target Groups (Grupos de destino) (en Load Balancing (Equilibrador de carga) en el panel de navegación izquierdo. En la columna Name (Nombre), seleccione el nombre del grupo de destino donde el valor de la columna Load balancer (Equilibrador de carga) coincida con el nombre de la columna `EXTERNAL-IP` de la salida en el paso anterior. Por ejemplo, debe seleccionar el grupo de destino denominado `k8s-default-samplese-`*xxxxxxxxxx* si la salida es la misma que la salida anterior. El Target type (Tipo de destino) es `IP` porque así se especificó en el manifiesto del servicio de muestra.

6. Seleccione el Grupo de destino y elija la pestaña Targets (Destinos). En Registered targets (Destinos registrados), debería ver tres direcciones IP de las tres réplicas implementadas en un paso anterior. Espere hasta que el estado de todos los destinos sea healthy (bueno) antes de continuar. Pueden pasar varios minutos hasta que todos los destinos sean `healthy`. Los destinos pueden estar en estado `unhealthy` antes de cambiar al estado `healthy`.
7. Envíe tráfico al servicio mediante el reemplazo de *xxxxxxxxxx-xxxxxxxxxxxxxxxx* y *us-west-2* por los valores devueltos en la salida de un paso anterior para `EXTERNAL-IP`. Si ha realizado la implementación en una subred privada, tendrá que ver la página desde un dispositivo dentro de la VPC, como un host bastión. Para obtener más información, consulte Host bastión de Linux en AWS.

   ```
   curl k8s-default-samplese-xxxxxxxxxx-xxxxxxxxxxxxxxxx.elb.region-code.amazonaws.com
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   <!DOCTYPE html>
   <html>
   <head>
   <title>Welcome to nginx!</title>
   [...]
   ```
8. Cuando haya terminado de utilizar la implementación, el servicio y el espacio de nombres de muestra, elimínelos.

   ```
   kubectl delete namespace nlb-sample-app
   ```

Equilibrio de carga de aplicaciones en Amazon EKS

Cuando crea una `ingress` de Kubernetes, se aprovisiona un equilibrador de carga de aplicación (ALB) de AWS que equilibra la carga del tráfico de aplicaciones. Para obtener más información, consulte ¿Qué es un equilibrador de carga de aplicación? en la Guía del usuario de equilibradores de carga de aplicación y Entrada en la documentación de Kubernetes. Los ALB se pueden utilizar con Pods que se implementan en nodos o en AWS Fargate. Puede implementar un ALB en subredes públicas o privadas.

El tráfico de aplicaciones se equilibra en L7 del modelo OSI. Para equilibrar la carga del tráfico de red en L4, implemente un `service` de Kubernetes del tipo `LoadBalancer`. Este tipo aprovisiona un

Network Load Balancer de AWS. Para obtener más información, consulte Equilibrio de carga de red en Amazon EKS. Para obtener más información sobre las diferencias entre los dos tipos de equilibrio de carga, consulte Características de Elastic Load Balancing en el sitio web de AWS.

Requisitos previos

Para poder equilibrar la carga del tráfico de aplicaciones a una aplicación, debe cumplir los siguientes requisitos.

- Tener un clúster existente. Si no tiene un clúster existente, consulte Introducción a Amazon EKS. Si necesita actualizar la versión de un clúster existente, consulte Actualización de una versión de Kubernetes de clúster de Amazon EKS.
- Implemente AWS Load Balancer Controller en su clúster. Para obtener más información, consulte ¿Qué es el AWS Load Balancer Controller?. Recomendamos la versión `2.7.2` o posterior.
- Al menos dos subredes en diferentes zonas de disponibilidad. El AWS Load Balancer Controller elige una subred de cada zona de disponibilidad. Cuando varias subredes etiquetadas se encuentran en una zona de disponibilidad, el controlador elige la subred cuyo ID de subred vaya primero lexicográficamente. Cada subred debe tener al menos ocho direcciones IP disponibles.

  Si utiliza varios grupos de seguridad adjuntos al nodo de trabajo, debe etiquetarse exactamente un grupo de seguridad de la siguiente manera. Sustituya *`my-cluster`* con el nombre del clúster.

  - Clave: `kubernetes.io/cluster/`*`my-cluster`*
  - Valor: `shared` o `owned`
- Si utiliza la versión AWS Load Balancer Controller `2.1.1` o anterior, las subredes deben etiquetarse con el siguiente formato. Si utiliza la versión `2.1.2` o posterior, el etiquetado es opcional. No obstante, recomendamos que etiquete una subred si se da cualquiera de las siguientes situaciones. Tiene varios clústeres que se ejecutan en la misma VPC o que tienen varios servicios de AWS que comparten subredes en una VPC. O bien, desea tener más control sobre dónde se aprovisionan los equilibradores de carga para cada clúster. Sustituya *`my-cluster`* con el nombre del clúster.

  - Clave: `kubernetes.io/cluster/`*`my-cluster`*
  - Valor: `shared` o `owned`
- Las subredes públicas y privadas deben cumplir los siguientes requisitos. Esto es así a menos que especifique de manera explícita los ID de subred como una anotación en un objeto de entrada o servicio. Supongamos que aprovisiona equilibradores de carga mediante la especificación explícita de los ID de subred como una anotación en un objeto de entrada o servicio. En esta situación,

Kubernetes y el controlador del equilibrador de carga de AWS utilizan esas subredes directamente para crear el equilibrador de carga y no se requieren las siguientes etiquetas.

- Subredes privadas: deben etiquetarse con el siguiente formato. De esta manera Kubernetes y el controlador del equilibrador de carga de AWS saben que las subredes se pueden utilizar para equilibradores de carga internos. Si utiliza `eksctl` o una plantilla de AWS CloudFormation de Amazon EKS para crear la VPC después del 26 de marzo de 2020, las subredes se etiquetan correctamente cuando se crean. Para obtener más información sobre las plantillas de VPC de AWS CloudFormation de Amazon EKS, consulte Creación de una VPC para su clúster de Amazon EKS.
  - Clave: `kubernetes.io/role/internal-elb`
  - Valor: 1
- Subredes públicas: deben etiquetarse con el siguiente formato. Esto es para que Kubernetes sepa que debe utilizar solo las subredes que se especificaron para los equilibradores de carga externos. De esta forma, Kubernetes no elige una subred pública en cada zona de disponibilidad (lexicográficamente en función de su ID de subred). Si utiliza `eksctl` o una plantilla de AWS CloudFormation de Amazon EKS para crear la VPC después del 26 de marzo de 2020, las subredes se etiquetan correctamente cuando se crean. Para obtener más información sobre las plantillas de VPC de AWS CloudFormation de Amazon EKS, consulte Creación de una VPC para su clúster de Amazon EKS.
  - Clave: `kubernetes.io/role/elb`
  - Valor: 1

Si las etiquetas de rol de subred no se agregan de manera explícita, el controlador de servicio de Kubernetes examina la tabla de enrutamiento de las subredes de la VPC del clúster. Esto sirve para determinar si la subred es privada o pública. Recomendamos que no dependa de este comportamiento. En su lugar, agregue de manera explícita las etiquetas de rol públicas o privadas. El AWS Load Balancer Controller no examina las tablas de enrutamiento. También requiere que las etiquetas privadas y públicas estén presentes para la detección automática correcta.

Consideraciones

- El Controlador del equilibrador de carga de AWS crea los ALB y los recursos de apoyo de AWS necesarios cada vez que se crea un recurso de entrada de Kubernetes en el clúster con la anotación `kubernetes.io/ingress.class: alb`. El recurso de entrada configura el ALB para dirigir el tráfico HTTP o HTTPS a diferentes Pods dentro del clúster. Para asegurarse de que los objetos de entrada utilizan el AWS Load Balancer Controller, agregue la siguiente anotación a su

especificación de entrada de Kubernetes. Para obtener más información, consulte Especificación de entrada en GitHub.

```
annotations:
    kubernetes.io/ingress.class: alb
```

> **Note**
>
> Si equilibra la carga de los Pods IPv6, agregue la siguiente anotación a su especificación de entrada. Solo puede equilibrar la carga a través de IPv6 a destinos IP, no a destinos de instancias. Sin esta anotación, la tarea de equilibrar la carga se realiza a través de IPv4.
>
> ```
> alb.ingress.kubernetes.io/ip-address-type: dualstack
> ```

- El AWS Load Balancer Controller admite los siguientes modos de tráfico:
  - Instancia: registra los nodos dentro del clúster como destinos para el ALB. El tráfico que llega al ALB se redirige a NodePort para su servicio y, luego, se envía a sus Pods. Este es el modo de tráfico predeterminado. También puede especificarlo de manera explícita con la anotación `alb.ingress.kubernetes.io/target-type: instance`.

    > **Note**
    >
    > Su servicio de Kubernetes debe especificar el NodePort o el tipo de "LoadBalancer" (Equilibrador de carga) necesario para utilizar este modo de tráfico.

  - IP: registra los Pods como destinos para el ALB. El tráfico que llega al ALB se redirige directamente a los Pods para su servicio. Debe especificar la anotación `alb.ingress.kubernetes.io/target-type: ip` necesaria para utilizar este modo de tráfico. El tipo de destino de IP es necesario cuando los Pods de destino se ejecutan en Fargate.
- Para etiquetar ALB creados por el controlador, agregue el siguiente comentario al controlador: `alb.ingress.kubernetes.io/tags`. Para ver una lista de todos los comentarios disponibles compatibles con el AWS Load Balancer Controller, consulte Anotaciones de entrada en GitHub.
- La actualización o degradación de la versión del controlador de ALB puede presentar cambios importantes en las características que dependen de él. Para obtener más información sobre los cambios importantes que se presentan en cada versión, consulte las Notas de la versión del controlador de ALB en GitHub.

Para compartir un Application Load Balancer entre varios recursos de servicio mediante **`IngressGroups`**

Para unir una entrada a un grupo, agregue la siguiente anotación a la especificación de un recurso de entrada de Kubernetes.

```
alb.ingress.kubernetes.io/group.name: my-group
```

El nombre del grupo debe:

- Tener 63 caracteres o menos de longitud.
- Constar de letras minúsculas, números, `-` y `.`
- Comenzar y terminar por un número o una letra.

El controlador fusiona de manera automática las reglas de entrada para todas las entradas del mismo grupo de entradas. Las soporta con un solo ALB. La mayoría de las anotaciones definidas en una entrada solo se aplican a las rutas definidas por esa entrada. De forma predeterminada, los recursos de entrada no pertenecen a ningún grupo de entradas.

> **Warning**
>
> Riesgo potencial de seguridad: especifique un grupo de entradas para una entrada solo cuando todos los usuarios de Kubernetes que tienen el permiso RBAC para crear o modificar recursos de entrada tienen el mismo límite de confianza. Si agrega la anotación con un nombre de grupo, otros usuarios de Kubernetes podrán crear o modificar sus entradas para que pertenezcan al mismo grupo de entradas. Si lo hace, puede provocar comportamientos indeseables, como sobrescribir reglas existentes con reglas de mayor prioridad.

Puede agregar un número de orden para el recurso de entrada.

```
alb.ingress.kubernetes.io/group.order: '10'
```

El número puede ser entre 1 y 1000. El número más bajo para todas las entradas del mismo grupo de entradas se evalúa primero. Todas las entradas sin esta anotación se evalúan con un valor de cero. Las reglas duplicadas con un número superior pueden sobrescribir reglas con un número inferior. De forma predeterminada, el orden de las reglas entre entradas dentro del mismo grupo de entradas se determina lexicográficamente en función del espacio de nombres y del nombre.

> **Important**
>
> Asegúrese de que cada entrada del mismo grupo de entradas tenga un número de prioridad único. No puede tener números de orden duplicados entre entradas.

(Opcional) Implementación de una aplicación de muestra

Requisitos previos

- Al menos una subred privada o pública en la VPC del clúster.
- Implemente AWS Load Balancer Controller en su clúster. Para obtener más información, consulte ¿Qué es el AWS Load Balancer Controller?. Recomendamos la versión `2.7.2` o posterior.

Para implementar una aplicación de muestra

Puede ejecutar la aplicación de muestra en un clúster que tenga nodos de Amazon EC2, Pods de Fargate o ambos.

1. Si no implementa en Fargate, omita este paso. Si va a implementar en Fargate, cree un perfil de Fargate. Puede crear el perfil con la ejecución del siguiente comando o en la AWS Management Console con los mismos valores para `name` y `namespace` que los del comando. Reemplace los *`example values`* por los de su propiedad.

   ```
   eksctl create fargateprofile \
       --cluster my-cluster \
       --region region-code \
       --name alb-sample-app \
       --namespace game-2048
   ```

2. Implemente el videojuego 2048 como una aplicación de muestra para comprobar que el AWS Load Balancer Controller crea un ALB de AWS como resultado del objeto de entrada. Complete los pasos del tipo de subred en la que va a realizar la implementación.

   a. Si va a implementar en Pods de un clúster que creó con la familia `IPv6`, vaya al siguiente paso.

      - Public

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
```

- Private

  1. Descargue el manifiesto.

     ```
     curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
     ```

  2. Edite el archivo y busque la línea que indica `alb.ingress.kubernetes.io/scheme: internet-facing`.
  3. Cambie *`internet-facing`* a **`internal`** y guarde el archivo.
  4. Aplique el manifiesto al clúster.

     ```
     kubectl apply -f 2048_full.yaml
     ```

b. Si va a implementar en Pods de un clúster que creó con la familia `IPv6`, lleve a cabo los pasos siguientes.

1. Descargue el manifiesto.

   ```
   curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
   ```

2. Abra el archivo en un editor y agregue la siguiente línea a las anotaciones de la especificación de entrada.

   ```
   alb.ingress.kubernetes.io/ip-address-type: dualstack
   ```

3. Si equilibra la carga en Pods internos, en lugar de en Pods orientados a Internet, cambie la línea que dice `alb.ingress.kubernetes.io/scheme: internet-facing` por `alb.ingress.kubernetes.io/scheme: internal`.
4. Guarde el archivo.
5. Aplique el manifiesto al clúster.

   ```
   kubectl apply -f 2048_full.yaml
   ```

3. Al cabo de unos minutos, verifique que el recurso de entrada se haya creado con el comando siguiente.

```
kubectl get ingress/ingress-2048 -n game-2048
```

Un ejemplo de salida sería el siguiente.

```
NAME           CLASS    HOSTS   ADDRESS
                       PORTS   AGE
ingress-2048   <none>   *       k8s-game2048-ingress2-xxxxxxxxxx-yyyyyyyyyy.region-
code.elb.amazonaws.com   80      2m32s
```

> **Note**
>
> Si ha creado el equilibrador de carga en una subred privada, el valor de `ADDRESS` en la salida anterior aparece precedido por `internal-`.
> Si la entrada no se creó correctamente después de varios minutos, ejecute el siguiente comando para ver los registros del AWS Load Balancer Controller. Estos registros pueden contener mensajes de error que pueden ayudarlo a diagnosticar cualquier problema con la implementación.
>
> ```
> kubectl logs -f -n kube-system -l app.kubernetes.io/instance=aws-load-
> balancer-controller
> ```

4. Si ha realizado la implementación en una subred pública, abra un navegador y navegue hasta la URL `ADDRESS` de la salida del comando anterior para ver la aplicación de muestra. Si no ve nada, actualice el navegador e inténtelo de nuevo. Si ha realizado la implementación en una subred privada, tendrá que ver la página desde un dispositivo dentro de la VPC, como un host bastión. Para obtener más información, consulte Host bastión de Linux en AWS.
5. Cuando termine de experimentar con la aplicación de muestra, elimínela ejecutando alguno de los siguientes comandos.
   - Si aplicó el manifiesto, en lugar de aplicar una copia que haya descargado, utilice el siguiente comando.

```
kubectl delete -f https://raw.githubusercontent.com/kubernetes-sigs/aws-load-
balancer-controller/v2.7.2/docs/examples/2048/2048_full.yaml
```

- Si descargó y editó el manifiesto, utilice el siguiente comando.

```
kubectl delete -f 2048_full.yaml
```

Restringir las direcciones IP externas que se pueden asignar a un servicio.

Se puede acceder a los servicios de Kubernetes desde el interior de un clúster a través de:

- Una dirección IP de clúster asignada automáticamente por Kubernetes.
- Cualquier dirección IP que especifique para la propiedad `externalIPs` en una especificación de servicio. Las direcciones IP externas no son administradas por Kubernetes y son responsabilidad del administrador del clúster. Las direcciones IP externas especificadas con `externalIPs` son diferentes de la dirección IP externa asignada a un servicio de tipo `LoadBalancer` por un proveedor de nube.

Para obtener más información sobre los servicios de Kubernetes, consulte Servicio en la documentación de Kubernetes. Puede restringir las direcciones IP que se pueden especificar para `externalIPs` en una especificación de servicio.

Para restringir las direcciones IP que se pueden especificar para **`externalIPs`** en una especificación de servicio

1. Implemente `cert-manager` para administrar certificados de webhook. Para obtener más información, consulte la documentación de `cert-manager`.

   ```
   kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v1.5.4/
   cert-manager.yaml
   ```

2. Compruebe que están ejecutando los Pods de `cert-manager`.

   ```
   kubectl get pods -n cert-manager
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME                                       READY   STATUS    RESTARTS   AGE
   cert-manager-58c8844bb8-nlx7q              1/1     Running   0          15s
   ```

```
cert-manager-cainjector-745768f6ff-696h5   1/1     Running   0          15s
cert-manager-webhook-67cc76975b-4v4nk      1/1     Running   0          14s
```

3. Revise sus servicios existentes para asegurarse de que ninguno de ellos tenga asignadas direcciones IP externas que no estén incluidas en el bloque de CIDR al que desea limitar las direcciones.

```
kubectl get services -A
```

Un ejemplo de salida sería el siguiente.

```
NAMESPACE                     NAME                                    TYPE
   CLUSTER-IP       EXTERNAL-IP     PORT(S)         AGE
cert-manager                  cert-manager                            ClusterIP
   10.100.102.137   <none>          9402/TCP        20m
cert-manager                  cert-manager-webhook                    ClusterIP
   10.100.6.136     <none>          443/TCP         20m
default                       kubernetes                              ClusterIP
   10.100.0.1       <none>          443/TCP         2d1h
externalip-validation-system  externalip-validation-webhook-service   ClusterIP
   10.100.234.179   <none>          443/TCP         16s
kube-system                   kube-dns                                ClusterIP
   10.100.0.10      <none>          53/UDP,53/TCP   2d1h
my-namespace                  my-service                              ClusterIP
   10.100.128.10    192.168.1.1     80/TCP          149m
```

Si alguno de los valores son direcciones IP que no están dentro del bloque al que desea restringir el acceso, deberá cambiar las direcciones para que estén dentro del bloque y volver a implementar los servicios. Por ejemplo, el servicio `my-service` de la salida anterior tiene asignada una dirección IP externa que no está dentro del ejemplo de bloque de CIDR en el paso 5.

4. Descargue el manifiesto de webhook de la IP externa. También puede ver el código fuente para el webhook en GitHub.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/docs/externalip-webhook.yaml
```

5. Especifica los bloques de CIDR. Abra el archivo descargado en el editor y elimine el # al comienzo de las siguientes líneas.

```
#args:
#- --allowed-external-ip-cidrs=10.0.0.0/8
```

Reemplace `10.0.0.0/8` por su propio bloque de CIDR. Puede especificar tantos bloques como desee. Si especifica varios bloques, agregue una coma entre bloques.

6. Si su clúster no está en la Región de AWS `us-west-2`, eemplace `us-west-2`, `602401143452` y `amazonaws.com` en el archivo con los siguientes comandos. Antes de ejecutar los comandos, reemplace *`region-code`* y *`111122223333`* con el valor de su Región de AWS de la lista en Registros de imágenes de contenedor de Amazon.

```
sed -i.bak -e 's|602401143452|111122223333|' externalip-webhook.yaml
sed -i.bak -e 's|us-west-2|region-code|' externalip-webhook.yaml
sed -i.bak -e 's|amazonaws.com||' externalip-webhook.yaml
```

7. Aplique el manifiesto al clúster.

```
kubectl apply -f externalip-webhook.yaml
```

Se producirá un error al intentar implementar un servicio en el clúster con una dirección IP especificada para `externalIPs` que no está incluida en los bloques que especificó en el paso Especificar bloques CIDR.

Copiar una imagen de contenedor de un repositorio en otro repositorio

En este tema se describe cómo extraer una imagen de contenedor de un repositorio al que los nodos no tienen acceso y enviar la imagen a un repositorio al que tienen acceso los nodos. Puede enviar la imagen a Amazon ECR o a un repositorio alternativo al que tengan acceso sus nodos.

Requisitos previos

- El motor de Docker instalado y configurado en su equipo. Para ver instrucciones, consulte Instalar motor de Docker en la documentación de Docker.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**.

Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

- Un punto de conexión de VPC de interfaz para Amazon ECR si desea que los nodos extraigan imágenes de contenedor o envíen imágenes de contenedor a un repositorio privado de Amazon ECR a través de la red de Amazon. Para obtener más información, consulte Crear los puntos de conexión de VPC para Amazon ECR en la Guía del usuario de Amazon Elastic Container Registry.

Siga los siguientes pasos para extraer una imagen de contenedor de un repositorio y enviarla a su propio repositorio. En los siguientes ejemplos que se proporcionan en este tema, se extrae la imagen del auxiliar de métricas Amazon VPC CNI plugin for Kubernetes. Cuando siga estos pasos, asegúrese de reemplazar los *`example values`* por sus propios valores.

Para copiar una imagen de contenedor de un repositorio a otro

1. Si todavía no tiene un repositorio de Amazon ECR u otro repositorio, cree uno al que tengan acceso sus nodos. El siguiente comando crea un repositorio privado de Amazon ECR. El nombre de un repositorio privado de Amazon ECR debe comenzar por una letra. Solo puede contener letras minúsculas, números, guiones (-), guiones bajos (_) y barras inclinadas (/). Para obtener más información, consulte Creación de un repositorio privado en la Guía del usuario de Amazon Elastic Container Registry.

   Puede reemplazar *`cni-metrics-helper`* con lo que elija. Como práctica recomendada, cree un repositorio independiente para cada imagen. Recomendamos esto porque las etiquetas de imagen deben ser únicas dentro de un repositorio. Reemplace *`region-code`* por una Región de AWS compatible con Amazon ECR.

   ```
   aws ecr create-repository --region region-code --repository-name cni-metrics-helper
   ```

2. Determine el registro, el repositorio y la etiqueta (opcional) de la imagen que deben extraer los nodos. Esta información se encuentra en el formato `registry/repository[:tag]`.

   Muchos de los temas de Amazon EKS sobre la instalación de imágenes requieren aplicar un archivo de manifiesto o instalar la imagen mediante un gráfico de Helm. Sin embargo, antes de

aplicar un archivo de manifiesto o instalar un gráfico de Helm, consulte primero el contenido del manifiesto o el archivo `values.yaml` del gráfico. Así podrá determinar el registro, el repositorio y la etiqueta que desea extraer.

Por ejemplo, puede encontrar la siguiente línea en el archivo de manifiesto para el auxiliar de métricas Amazon VPC CNI plugin for Kubernetes. El registro es `602401143452.dkr.ecr.us-west-2.amazonaws.com`, que es un registro privado de Amazon ECR. El repositorio es `cni-metrics-helper`.

```
image: "602401143452.dkr.ecr.us-west-2.amazonaws.com/cni-metrics-helper:v1.12.6"
```

Podría ver las siguientes variaciones para la ubicación de una imagen:

- Solo `repository-name:tag`. En este caso, `docker.io` suele ser el registro, pero no se especifica ya que Kubernetes lo antepone a un nombre de repositorio de forma predeterminada si no se especifica ningún registro.
- `repository-name/repository-namespace/repository:tag`. Es opcional definir un espacio de nombres para el repositorio, pero a veces lo especifica el propietario del repositorio para clasificar las imágenes. Por ejemplo, todas las imágenes de Amazon EC2 de la galería pública de Amazon ECR usan el espacio de nombres `aws-ec2`.

Antes de instalar una imagen con Helm, consulte el archivo `values.yaml` de Helm para determinar la ubicación de la imagen. Por ejemplo, el archivo `values.yaml` para el auxiliar de métricas Amazon VPC CNI plugin for Kubernetes incluye las siguientes líneas.

```
image:
  region: us-west-2
  tag: v1.12.6
  account: "602401143452"
  domain: "amazonaws.com"
```

3. Extraiga la imagen del contenedor especificada en el archivo de manifiesto.

   a. Si la extracción es de un registro público, como la galería pública de Amazon ECR, puede pasar al siguiente subpaso porque no es necesaria la autenticación. En este ejemplo, se autentica en un registro privado de Amazon ECR que contiene el repositorio de la imagen auxiliar de métricas de CNI. Amazon EKS mantiene la imagen en cada registro que aparece en Registros de imágenes de contenedor de Amazon. Puede autenticarse en cualquiera

de los registros reemplazando *602401143452* y *region-code* por la información de otro registro. Existe un registro independiente para cada Región de AWS compatible con Amazon EKS.

```
aws ecr get-login-password --region region-code | docker login --username AWS --password-stdin 602401143452.dkr.ecr.region-code.amazonaws.com
```

b. Extraiga la imagen. En este ejemplo, extrae del registro en el que se autenticó en el subpaso anterior. Reemplace *602401143452* y *region-code* con la información proporcionada en el subpaso anterior.

```
docker pull 602401143452.dkr.ecr.region-code.amazonaws.com/cni-metrics-helper:v1.12.6
```

4. Etiquete la imagen extraída con su registro, repositorio y etiqueta. En el siguiente ejemplo se supone que ha extraído la imagen del archivo de manifiesto y la va a enviar al repositorio privado de Amazon ECR que creó en el primer paso. Reemplace *111122223333* por su ID de cuenta. Reemplace *region-code* por su Región de AWS que creó en su repositorio privado de Amazon ECR.

```
docker tag cni-metrics-helper:v1.12.6 111122223333.dkr.ecr.region-code.amazonaws.com/cni-metrics-helper:v1.12.6
```

5. Realice la autenticación del registro. En este ejemplo, se autenticará en el registro privado de Amazon ECR que creó en el primer paso. Para obtener más información, consulte Autenticación de registros en la Guía del usuario de Amazon Elastic Container Registry.

```
aws ecr get-login-password --region region-code | docker login --username AWS --password-stdin 111122223333.dkr.ecr.region-code.amazonaws.com
```

6. Inserte la imagen en el repositorio. En este ejemplo, envía la imagen en el repositorio privado de Amazon ECR que creó en el primer paso. Para obtener más información, consulte Inserción de una imagen de Docker en la Guía del usuario de Amazon Elastic Container Registry.

```
docker push 111122223333.dkr.ecr.region-code.amazonaws.com/cni-metrics-helper:v1.12.6
```

7. Actualice el archivo de manifiesto que utilizó para determinar la imagen en un paso anterior con el `registry/repository:tag` para la imagen que envió. Si hace la instalación con un gráfico de Helm, a menudo hay una opción para especificar el `registry/repository:tag`. Al

instalar el gráfico, especifique el `registry/repository:tag` para la imagen que envió a su repositorio.

Registros de imágenes de contenedor de Amazon

Al implementar complementos de Amazon EKS de AWS en su clúster, sus nodos extraen las imágenes del contenedor necesarias del registro especificado en el mecanismo de instalación del complemento, como un manifiesto de instalación o un archivo Helm `values.yaml`. Las imágenes se extraen de un repositorio privado de Amazon ECR de Amazon EKS. Amazon EKS replica las imágenes en un repositorio de cada Región de AWS compatible con Amazon EKS. Los nodos pueden extraer la imagen del contenedor a través de Internet desde cualquiera de los siguientes registros. Como alternativa, sus nodos pueden extraer la imagen a través de la red de Amazon si ha creado un punto de conexión de VPC de interfaz para Amazon ECR (AWS PrivateLink) en la VPC. Los registros requieren autenticación con una cuenta de AWS IAM. Los nodos se autentican mediante el rol de IAM del nodo de Amazon EKS, que tiene asociados a él los permisos de la política de IAM administrada AmazonEC2ContainerRegistryReadOnly.

Región de AWS	Registro
af-south-1	877085696533.dkr.ecr.af-south-1.amazonaws.com
ap-east-1	800184023465.dkr.ecr.ap-east-1.amazonaws.com
ap-northeast-1	602401143452.dkr.ecr.ap-northeast-1.amazonaws.com
ap-northeast-2	602401143452.dkr.ecr.ap-northeast-2.amazonaws.com
ap-northeast-3	602401143452.dkr.ecr.ap-northeast-3.amazonaws.com
ap-south-1	602401143452.dkr.ecr.ap-south-1.amazonaws.com

Región de AWS	Registro
ap-south-2	900889452093.dkr.ecr.ap-south-2.amazonaws.com
ap-southeast-1	602401143452.dkr.ecr.ap-southeast-1.amazonaws.com
ap-southeast-2	602401143452.dkr.ecr.ap-southeast-2.amazonaws.com
ap-southeast-3	296578399912.dkr.ecr.ap-southeast-3.amazonaws.com
ap-southeast-4	491585149902.dkr.ecr.ap-southeast-4.amazonaws.com
ca-central-1	602401143452.dkr.ecr.ca-central-1.amazonaws.com
ca-west-1	761377655185.dkr.ecr.ca-west-1.amazonaws.com
cn-north-1	918309763551.dkr.ecr.cn-north-1.amazonaws.com.cn
cn-northwest-1	961992271922.dkr.ecr.cn-northwest-1.amazonaws.com.cn
eu-central-1	602401143452.dkr.ecr.eu-central-1.amazonaws.com
eu-central-2	900612956339.dkr.ecr.eu-central-2.amazonaws.com
eu-north-1	602401143452.dkr.ecr.eu-north-1.amazonaws.com
eu-south-1	590381155156.dkr.ecr.eu-south-1.amazonaws.com

Región de AWS	Registro
eu-south-2	455263428931.dkr.ecr.eu-south-2.amazonaws.com
eu-west-1	602401143452.dkr.ecr.eu-west-1.amazonaws.com
eu-west-2	602401143452.dkr.ecr.eu-west-2.amazonaws.com
eu-west-3	602401143452.dkr.ecr.eu-west-3.amazonaws.com
il-central-1	066635153087.dkr.ecr.il-central-1.amazonaws.com
me-south-1	558608220178.dkr.ecr.me-south-1.amazonaws.com
me-central-1	759879836304.dkr.ecr.me-central-1.amazonaws.com
sa-east-1	602401143452.dkr.ecr.sa-east-1.amazonaws.com
us-east-1	602401143452.dkr.ecr.us-east-1.amazonaws.com
us-east-2	602401143452.dkr.ecr.us-east-2.amazonaws.com
us-gov-east-1	151742754352.dkr.ecr.us-gov-east-1.amazonaws.com
us-gov-west-1	013241004608.dkr.ecr.us-gov-west-1.amazonaws.com
us-west-1	602401143452.dkr.ecr.us-west-1.amazonaws.com

Región de AWS	Registro
us-west-2	602401143452.dkr.ecr.us-west-2.amazonaws.com

Complementos de Amazon EKS

Un complemento es un software que proporciona capacidades operativas de soporte para aplicaciones de Kubernetes, pero no es específico de la aplicación. Esto incluye software como agentes de observabilidad o controladores de Kubernetes que permiten al clúster interactuar con recursos subyacentes de AWS para redes, informática y almacenamiento. La comunidad de Kubernetes, proveedores de nube como AWS o proveedores de terceros suelen crear y mantener este tipo de software. Amazon EKS instala de forma automática complementos autoadministrados como Amazon VPC CNI plugin for Kubernetes, `kube-proxy` y CoreDNS para cada clúster. Puede cambiar la configuración predeterminada de los complementos y actualizarlos cuando lo desee.

Los complementos de Amazon EKS proporcionan la instalación y administración de un conjunto seleccionado de complementos para clústeres de Amazon EKS. Todos los complementos de Amazon EKS incluyen los parches de seguridad más recientes, correcciones de errores y están validados por AWS para trabajar con Amazon EKS. Los complementos de Amazon EKS permiten garantizar de forma coherente que los clústeres de Amazon EKS sean seguros y estables, así como reducir la cantidad de trabajo necesario para instalar, configurar y actualizar complementos. Si un complemento autoadministrado, como `kube-proxy`, ya se ejecuta en su clúster y está disponible como complemento de Amazon EKS, a continuación, puede instalar el complemento `kube-proxy` de Amazon EKS para comenzar a beneficiarse de las capacidades de los complementos de Amazon EKS.

Puede actualizar campos de configuración administrados de Amazon EKS específicos para complementos de Amazon EKS a través de la API de Amazon EKS. Además, puede modificar los campos de configuración no administrados por Amazon EKS directamente en el clúster de Kubernetes una vez que se inicie el complemento. Esto incluye la definición de campos de configuración específicos para un complemento cuando corresponda. Amazon EKS no anulará estos cambios una vez realizados. Esto es posible mediante la característica de aplicación del lado del servidor de Kubernetes. Para obtener más información, consulte Administración de campos de Kubernetes.

Puede usar complementos de Amazon EKS con cualquier tipo de nodos de Amazon EKS.

Consideraciones

- Para configurar complementos para el clúster, la entidad principal de IAM debe tener permisos de IAM para trabajar con complementos. Para obtener más información, consulte las acciones con Addon en su nombre en Acciones definidas por Amazon Elastic Kubernetes Service.
- Los complementos de Amazon EKS se ejecutan en los nodos que aprovisiona o configura para el clúster. Los tipos de nodo incluyen instancias de Amazon EC2 y Fargate.
- Puede modificar campos que no estén administrados por Amazon EKS para personalizar la instalación de un complemento de Amazon EKS. Para obtener más información, consulte Administración de campos de Kubernetes.
- Si crea un clúster con la AWS Management Console, el de Amazon EKS `kube-proxy`, Amazon VPC CNI plugin for Kubernetes y CoreDNS de Amazon EKS se agregan automáticamente al clúster. Si utiliza `eksctl` para crear el clúster con un archivo de `config`, `eksctl` también puede crear el clúster con complementos de Amazon EKS. Si crea el clúster con `eksctl` sin un archivo de `config` o con cualquier otra herramienta, el `kube-proxy` autoadministrado, la Amazon VPC CNI plugin for Kubernetes y los complementos CoreDNS están instalados, en lugar de los complementos de Amazon EKS. Puede administrarlos usted mismo o agregar los complementos de Amazon EKS de forma manual después de crear el clúster.
- El `ClusterRoleBinding` de `eks:addon-cluster-admin` une las `ClusterRole` de `cluster-admin` a la identidad Kubernetes de `eks:addon-manager`. El rol tiene los permisos necesarios para que la identidad de `eks:addon-manager` cree espacios de nombres de Kubernetes e instale complementos en los espacios de nombres. Si el `ClusterRoleBinding` `eks:addon-cluster-admin` se elimina, el clúster de Amazon EKS seguirá funcionando; sin embargo, Amazon EKS ya no podrá administrar ningún complemento. Todos los clústeres que comiencen con las siguientes versiones de plataforma utilizan el nuevo `ClusterRoleBinding`.

Versión de Kubernetes	Versión de plataforma de EKS
1.28	eks.2012

Versión de Kubernetes	plataforma de EKS
1.21	eks-14
1.22	eks-9
1.23	eks-5
1.24	eks-3

Puede agregar, actualizar o eliminar complementos de Amazon EKS mediante la API de Amazon EKS, la AWS Management Console, la AWS CLI y `eksctl`. Para obtener más información, consulte Administración de los complementos de Amazon EKS. También puede crear complementos de Amazon EKS con AWS CloudFormation.

Complementos de Amazon EKS disponibles en Amazon EKS

Los siguientes complementos de Amazon EKS están disponibles para crearlos en el clúster. Siempre puede ver la lista más actualizada de complementos disponibles mediante `eksctl`, la AWS Management Console o la AWS CLI. Para ver todos los complementos disponibles o instalar un complemento, consulte Creación de un complemento. Si un complemento requiere permisos de IAM, debe tener un proveedor de IAM OpenID Connect (OIDC) para el clúster. Para determinar si ya tiene uno o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster. Puede actualizar o eliminar un complemento una vez que lo haya instalado.

Elija un complemento para obtener más información sobre él y sus requisitos de instalación.

Amazon VPC CNI plugin for Kubernetes

- Nombre: `vpc-cni`

- Descripción: es un complemento de interfaz de red de contenedores (CNI) de Kubernetes que proporciona redes de VPC nativas para el clúster. El tipo autoadministrado o administrado de este complemento se instala en cada nodo de Amazon EC2 de forma predeterminada.
- Permisos de IAM requeridos: este complemento utiliza la capacidad roles de IAM para cuentas de servicio de Amazon EKS. Si el clúster usa la familia `IPv4`, se requieren los permisos de la AmazonEKS_CNI_Policy. Si el clúster utiliza la familia `IPv6`, entonces debe crear una política de IAM con los permisos del modo IPv6. Puede crear un rol de IAM, asociarle una de las políticas y anotar la cuenta de servicio de Kubernetes utilizada por el complemento con el siguiente comando.

  Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`AmazonEKSVPCCNIRole`* con el nombre para su rol. Si el clúster usa la familia `IPv6`, reemplace *`AmazonEKS_CNI_Policy`* por el nombre de la política que creó. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol, asociarle la política y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

  ```
  eksctl create iamserviceaccount --name aws-node --namespace kube-system --cluster my-cluster --role-name AmazonEKSVPCCNIRole \
      --role-only --attach-policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy --approve
  ```

- Información adicional: para obtener más información sobre los ajustes configurables del complemento, consulte aws-vpc-cni-k8s en GitHub. Para obtener más información sobre el complemento, consulte Proposal: CNI plugin for Kubernetes networking over AWS VPC. Para obtener más información acerca de la creación de un complemento, consulte Creación del complemento de Amazon EKS.
- Información sobre la actualización: solo puede actualizar una versión secundaria a la vez. Por ejemplo, si su versión actual es la `1.28.`*`x`*`-eksbuild.`*`y`* y desea actualizarla a la `1.30.`*`x`*`-eksbuild.`*`y`*, primero debe actualizarla a la `1.29.`*`x`*`-eksbuild.`*`y`* y luego a la `1.30.`*`x`*`-eksbuild.`*`y`*. Para obtener más información acerca de la actualización de complementos, consulte Actualizar el complemento de Amazon EKS.

CoreDNS

- Nombre: `coredns`
- Descripción: es un servidor de DNS flexible y extensible que puede servir como el DNS del clúster de Kubernetes. El tipo autoadministrado o administrado de este complemento se instaló de forma

predeterminada cuando se creó el clúster. Al lanzar un clúster de Amazon EKS con al menos un nodo, se implementan dos réplicas de la imagen de CoreDNS de forma predeterminada, independientemente del número de nodos implementados en el clúster. Los Pods CoreDNS proporcionan resolución de nombres para todos los Pods del clúster. Los Pods CoreDNS se pueden implementar en los nodos de Fargate si el clúster incluye un Perfil de AWS Fargate con un espacio de nombres que coincida con el espacio de nombres para la `deployment` de CoreDNS.

- Permisos de IAM requeridos: este complemento no requiere ningún permiso.
- Información adicional: para obtener más información sobre CoreDNS, consulte Using CoreDNS for Service Discovery (Uso de CoreDNS para la detección de servicios) y Customizing DNS Service (Personalización del servicio DNS) en la documentación de Kubernetes.

`Kube-proxy`

- Nombre: `kube-proxy`
- Descripción: mantiene las reglas de red en cada nodo de Amazon EC2. Permite la comunicación de red con sus Pods. De forma predeterminada, el tipo autoadministrado o administrado de este complemento se instala en cada nodo de Amazon EC2 en el clúster.
- Permisos de IAM requeridos: este complemento no requiere ningún permiso.
- Información adicional: para obtener más información sobre `kube-proxy`, consulte `kube-proxy` en la documentación de Kubernetes.
- Información sobre la actualización: antes de actualizar la versión actual, tenga en cuenta los siguientes requisitos:
  - `Kube-proxy` en un clúster de Amazon EKS tiene la misma política de compatibilidad y sesgo que Kubernetes.
  - `Kube-proxy` debe ser la misma versión secundaria que `kubelet` en los nodos de Amazon EC2.
  - El `Kube-proxy` no puede ser posterior a la versión secundaria del plano de control del clúster.
  - La versión `kube-proxy` de su Amazon EC2 no puede ser más de dos versiones secundarias anteriores a la versión del plano de control. Por ejemplo, si el plano de control se ejecuta con Kubernetes 1.30, la versión secundaria de `kube-proxy` no puede ser anterior a 1.28.
  - Si recientemente actualizó el clúster a una nueva versión secundaria de Kubernetes, actualice los nodos de Amazon EC2 a la misma versión secundaria antes de actualizar el `kube-proxy` a la misma versión secundaria que los nodos.

Controlador CSI de Amazon EBS

- Nombre: `aws-ebs-csi-driver`
- Descripción: es un complemento de interfaz de almacenamiento de contenedores (CSI) de Kubernetes que proporciona almacenamiento de Amazon EBS para el clúster.
- Permisos de IAM requeridos: este complemento utiliza los roles de IAM para cuentas de servicio de Amazon EKS. Los permisos de la política administrada `AmazonEBSCSIDriverPolicy` de AWS son obligatorios. Cree un rol de IAM y adjunte la política administrada del rol de IAM con el siguiente comando. Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`AmazonEKS_EBS_CSI_DriverRole`* con el nombre para su rol. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente o necesita usar una clave de KMS personalizada para el cifrado, consulteCreación del rol de IAM del controlador de CSI de Amazon EBS.

```
eksctl create iamserviceaccount \
    --name ebs-csi-controller-sa \
    --namespace kube-system \
    --cluster my-cluster \
    --role-name AmazonEKS_EBS_CSI_DriverRole \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy
 \
    --approve
```

- Información adicional: para obtener más información sobre el complemento, consulte Controlador CSI de Amazon EBS.

Controlador CSI de Amazon EFS

- Nombre: `aws-efs-csi-driver`
- Descripción: es un complemento de interfaz de almacenamiento de contenedores (CSI) de Kubernetes que proporciona almacenamiento de Amazon EFS para el clúster.
- Permisos de IAM requeridos: este complemento utiliza la capacidad roles de IAM para cuentas de servicio de Amazon EKS. Los permisos de la política administrada `AmazonEFSCSIDriverPolicy` de AWS son obligatorios. Cree un rol de IAM y adjunte la política administrada del rol de IAM con los siguientes comandos. Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`AmazonEKS_EFS_CSI_DriverRole`* con el nombre para su rol. Estos

comandos requieren que tenga `eksctl` instalado en su dispositivo. Si necesita utilizar una herramienta diferente, consulte Creación de un rol de IAM.

```
export cluster_name=my-cluster
export role_name=AmazonEKS_EFS_CSI_DriverRole
eksctl create iamserviceaccount \
    --name efs-csi-controller-sa \
    --namespace kube-system \
    --cluster $cluster_name \
    --role-name $role_name \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEFSCSIDriverPolicy
 \
    --approve
TRUST_POLICY=$(aws iam get-role --role-name $role_name --query
 'Role.AssumeRolePolicyDocument' | \
    sed -e 's/efs-csi-controller-sa/efs-csi-*/' -e 's/StringEquals/StringLike/')
aws iam update-assume-role-policy --role-name $role_name --policy-document
 "$TRUST_POLICY"
```

- Información adicional: para obtener más información sobre el complemento, consulte Controlador CSI de Amazon EFS.

Mountpoint para el controlador CSI de Amazon S3

- Nombre: `aws-mountpoint-s3-csi-driver`
- Descripción: Es un complemento de interfaz de almacenamiento de contenedores (CSI) de Kubernetes que proporciona almacenamiento de Amazon S3 para el clúster.
- Permisos de IAM requeridos: este complemento utiliza la capacidad roles de IAM para cuentas de servicio de Amazon EKS. El rol de IAM que se cree requerirá una política que dé acceso a S3. Siga las recomendaciones de permisos de Mountpoint IAM al crear la política. Como alternativa, puede utilizar la política gestionada de AWS AmazonS3FullAccess, pero esta política gestionada concede más permisos de los necesarios para Mountpoint.

  Cree un rol de IAM y adjunte la política a él con los siguientes comandos. *Reemplace my-cluster por el nombre de su clúster, region-code por el código de Región de AWS correcto, AmazonEKS_S3_CSI_DriverRole por el nombre de su rol y AmazonEKS_S3_CSI_DriverRole_ARN* por el ARN del rol. Estos comandos requieren que tenga

`eksctl` instalado en su dispositivo. Para obtener instrucciones sobre cómo utilizar la consola de IAM o AWS CLI, consulte Creación de un rol de IAM.

```
CLUSTER_NAME=my-cluster
REGION=region-code
ROLE_NAME=AmazonEKS_S3_CSI_DriverRole
POLICY_ARN=AmazonEKS_S3_CSI_DriverRole_ARN
eksctl create iamserviceaccount \
    --name s3-csi-driver-sa \
    --namespace kube-system \
    --cluster $CLUSTER_NAME \
    --attach-policy-arn $POLICY_ARN \
    --approve \
    --role-name $ROLE_NAME \
    --region $REGION \
    --role-only
```

- Información adicional: para obtener más información sobre el complemento, consulte Mountpoint para el controlador CSI de Amazon S3.

Controlador de instantáneas CSI

- Nombre: `snapshot-controller`
- Descripción: El controlador de instantáneas de la interfaz de almacenamiento de contenedores (CSI) permite utilizar la funcionalidad de instantáneas en controladores CSI compatibles, como el controlador CSI de Amazon EBS.
- Permisos de IAM requeridos: este complemento no requiere ningún permiso.
- Información adicional: para obtener más información sobre el complemento, consulte Controlador de instantáneas CSI.

AWS Distro para OpenTelemetry

- Nombre: `adot`
- Descripción: AWS Distro para OpenTelemetry (ADOT), es una distribución segura, lista para la producción y compatible con AWS del proyecto OpenTelemetry.
- Permisos de IAM necesarios: Este complemento solo requiere permisos de IAM si utiliza uno de los recursos personalizados configurados previamente que se pueden utilizar mediante una configuración avanzada.

- Información adicional: Para obtener más información, consulte Introducción a AWS Distro para OpenTelemetry con complementos de EKS en la documentación de AWS Distro para OpenTelemetry.

  ADOT requiere que el `cert-manager` esté implementado en el clúster como requisito previo, de lo contrario, este complemento no funcionará si se implementa directamente mediante la propiedad `cluster_addons` de Amazon EKS Terraform. Para obtener más información sobre los requisitos, consulte Requisitos para iniciar con AWS Distro para OpenTelemetry mediante los complementos de EKS en la AWS Distro para la documentación de OpenTelemetry.

Agente de Amazon GuardDuty

- Nombre: `aws-guardduty-agent`
- Descripción: Amazon GuardDuty es un servicio de monitoreo de seguridad que analiza y procesa los orígenes de datos fundamentales, incluidos los eventos de administración de AWS CloudTrail y los registros de flujo de Amazon VPC. Amazon GuardDuty también procesa características, como los registros de auditoría de Kubernetes y la supervisión del tiempo de ejecución.
- Permisos de IAM requeridos: este complemento no requiere ningún permiso.
- Información adicional: para obtener más información, consulte Runtime Monitoring for Amazon EKS clusters in Amazon GuardDuty.
  - Para detectar posibles amenazas de seguridad en sus clústeres de Amazon EKS, habilite el monitoreo del tiempo de ejecución de Amazon GuardDuty e implemente el agente de seguridad de GuardDuty en sus clústeres de Amazon EKS.

Agente de Amazon CloudWatch Observability

- Nombre: `amazon-cloudwatch-observability`
- Descripción Amazon CloudWatch Agent es el servicio de monitoreo y observabilidad ofrecido por AWS. Este complemento instala el CloudWatch Agent y habilita tanto CloudWatch Application Signals como CloudWatch Container Insights con una observabilidad mejorada para Amazon EKS.
- Permisos de IAM requeridos: este complemento utiliza la capacidad roles de IAM para cuentas de servicio de Amazon EKS. Los permisos de las políticas administradas AWSXrayWriteOnlyAccess y CloudWatchAgentServerPolicy de AWS son obligatorios. Puede crear un rol de IAM, asociarle políticas administradas y anotar la cuenta de servicio de Kubernetes utilizada por el complemento con el siguiente comando. Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`AmazonEKS_Observability_role`* con el nombre para su rol. Este comando requiere que

tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol, asociarle la política y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

```
eksctl create iamserviceaccount \
    --name cloudwatch-agent \
    --namespace amazon-cloudwatch \
    --cluster my-cluster \
    --role-name AmazonEKS_Observability_Role \
    --role-only \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
    --approve
```

- Información adicional: para obtener más información, consulte Install the CloudWatch Agent.

Agente de Pod Identity de Amazon EKS

- Nombre: `eks-pod-identity-agent`
- Descripción: Pod Identity de Amazon EKS ofrece la posibilidad de administrar las credenciales para las aplicaciones, de un modo similar a cómo los perfiles de instancia de Amazon EC2 proporcionan credenciales a instancias de EC2.
- Permisos de IAM requeridos: Este complemento usa los permisos de Rol de IAM de nodo de Amazon EKS.
- Información sobre la actualización: solo puede actualizar una versión secundaria a la vez. Por ejemplo, si su versión actual es la `1.28.`*`x`*`-eksbuild.`*`y`* y desea actualizarla a la `1.30.`*`x`*`-eksbuild.`*`y`*, primero debe actualizarla a la `1.29.`*`x`*`-eksbuild.`*`y`* y luego a la `1.30.`*`x`*`-eksbuild.`*`y`*. Para obtener más información acerca de la actualización de complementos, consulte Actualizar el complemento de Amazon EKS.

Complementos adicionales de Amazon EKS de proveedores de software independientes

Además de la lista anterior de complementos de Amazon EKS, también puede agregar una amplia selección de complementos de Amazon EKS de software operativo de proveedores de software independientes. Elija un complemento para obtener más información sobre él y sus requisitos de instalación.

Busque, adquiera e implemente complementos de AWS Marketplace en Amazon EKS (YouTube).

Accuknox

- Editor: Accuknox
- Nombre: `accuknox_kubearmor`
- Espacio de nombres: `kubearmor`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Getting Started with KubeArmor en la documentación de KubeArmor.

Akuity

- Editor: Akuity
- Nombre: `akuity_agent`
- Espacio de nombres: `akuity`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Instalación del agente Akuity en Amazon EKS con el complemento Akuity EKS en la documentación de la plataforma Akuity.

Calyptia

- Editor: Calyptia
- Nombre: `calyptia_fluent-bit`
- Espacio de nombres: `calytia-fluentbit`
- Nombre de la cuenta de servicio: `clyptia-fluentbit`

- Política de IAM administrada de AWS: AWSMarketplaceMeteringRegisterUsage.
- Comando para crear el rol de IAM requerido: el siguiente comando requiere que tenga un proveedor OpenID Connect (OIDC) de IAM para su clúster. Para determinar si ya tiene uno o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster. Reemplace *my-cluster* con el nombre del clúster y reemplace *my-calyptia-role* con el nombre para su rol. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

```
eksctl create iamserviceaccount --name service-account-name  --namespace calyptia-
fluentbit --cluster my-cluster --role-name my-calyptia-role \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/
AWSMarketplaceMeteringRegisterUsage --approve
```

- Instrucciones de configuración y uso: Consulte Calyptia for Fluent Bit en la documentación de Calyptia.

Cisco Observability Collector

- Editor: Cisco
- Nombre: `cisco_cisco-cloud-observability-collectors`
- Espacio de nombres: `appdynamics`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Uso de los complementos de AWS Marketplace de Cisco Cloud Observability en la documentación de Cisco AppDynamics.

Cisco Observability Operator

- Editor: Cisco
- Nombre: `cisco_cisco-cloud-observability-operators`
- Espacio de nombres: `appdynamics`

- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Uso de los complementos de AWS Marketplace de Cisco Cloud Observability en la documentación de Cisco AppDynamics.

CLOUDSOFT

- Editor: CLOUDSOFT
- Nombre: `cloudsoft_cloudsoft-amp`
- Espacio de nombres: `cloudsoft-amp`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte la Complemento de Amazon EKS en la documentación de CLOUDSOFT.

Cribl

- Editor: Cribl
- Nombre: `cribl_cribledge`
- Espacio de nombres: `cribledge`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the Cribl Amazon EKS Add-on for Edge en la documentación de Cribl.

Dynatrace

- Editor: Dynatrace
- Nombre: `dynatrace_dynatrace-operator`
- Espacio de nombres: `dynatrace`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte la supervisión de Kubernetes en la documentación de dynatrace.

Datree

- Editor: Datree
- Nombre: `datree_engine-pro`
- Espacio de nombres: `datree`
- Nombre de la cuenta de servicio: datree-webhook-server-awsmp
- Política de IAM administrada de AWS: AWSLicenseManagerConsumptionPolicy.
- Comando para crear el rol de IAM requerido: el siguiente comando requiere que tenga un proveedor OpenID Connect (OIDC) de IAM para su clúster. Para determinar si ya tiene uno o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster. Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`my-datree-role`* con el nombre para su rol. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

  ```
  eksctl create iamserviceaccount --name datree-webhook-server-awsmp --namespace datree
   --cluster my-cluster --role-name my-datree-role \
      --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/
  AWSLicenseManagerConsumptionPolicy --approve
  ```

- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.

- Instrucciones de configuración y uso: consulte la integración con Amazon EKS en la documentación de Datree.

Datadog

- Editor: Datadog
- Nombre: `datadog_operator`
- Espacio de nombres: `datadog-agent`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the Datadog Agent on Amazon EKS with the Datadog Operator Add-on en la documentación de Datadog.

Groundcover

- Editor: groundcover
- Nombre: `groundcover_agent`
- Espacio de nombres: `groundcover`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the groundcover Amazon EKS Add-on en la documentación de groundcover.

Grafana Labs

- Editor: Grafana Labs
- Nombre: `grafana-labs_kubernetes-monitoring`
- Espacio de nombres: `monitoring`

- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Configurar la monitorización de Kubernetes como un complemento con Amazon EKS en la documentación de Grafana Labs.

HA Proxy

- Editor: HA Proxy
- Nombre: `haproxy-technologies_kubernetes-ingress-ee`
- Espacio de nombres: `haproxy-controller`
- Nombre de la cuenta de servicio: `customer defined`
- Política de IAM administrada de AWS: AWSLicenseManagerConsumptionPolicy.
- Comando para crear el rol de IAM requerido: el siguiente comando requiere que tenga un proveedor OpenID Connect (OIDC) de IAM para su clúster. Para determinar si ya tiene uno o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster. Reemplace *my-cluster* con el nombre del clúster y reemplace *my-haproxy-role* con el nombre para su rol. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

```
eksctl create iamserviceaccount --name service-account-name  --namespace haproxy-
controller --cluster my-cluster --role-name my-haproxy-role \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/
AWSLicenseManagerConsumptionPolicy --approve
```

- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Instalar Kubernetes de HAProxy Enterprise Ingress Controller en Amazon EKS desde AWS en la documentación de HAProxy.

Kpow

- Editor: Factorhouse

- Nombre: `factorhouse_kpow`
- Espacio de nombres: `factorhouse`
- Nombre de la cuenta de servicio: `kpow`
- Política de IAM administrada de AWS: AWSLicenseManagerConsumptionPolicy
- Comando para crear el rol de IAM requerido: el siguiente comando requiere que tenga un proveedor OpenID Connect (OIDC) de IAM para su clúster. Para determinar si ya tiene uno o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster. Reemplace *my-cluster* con el nombre del clúster y reemplace *my-kpow-role* con el nombre para su rol. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

```
eksctl create iamserviceaccount --name kpow --namespace factorhouse --cluster my-
cluster --role-name my-kpow-role \
    --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/
AWSLicenseManagerConsumptionPolicy --approve
```

- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte la LM de AWS Marketplace en la documentación de Kpow.

Kubecost

- Editor: Kubecost
- Nombre: `kubecost_kubecost`
- Espacio de nombres: `kubecost`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Configuración e instrucciones de uso – Consulte Integración de facturación de AWS Cloud en la documentación de Kubecost.

- Si la versión del clúster es la `1.23` o una posterior, debe haber instalado antes the section called "Controlador CSI de Amazon EBS" en el clúster; de lo contrario recibirá un error.

Kasten

- Editor: Kasten by Veeam
- Nombre: `kasten_k10`
- Espacio de nombres: `kasten-io`
- Nombre de la cuenta de servicio: `k10-k10`
- Política de IAM administrada de AWS: AWSLicenseManagerConsumptionPolicy.
- Comando para crear el rol de IAM requerido: el siguiente comando requiere que tenga un proveedor OpenID Connect (OIDC) de IAM para su clúster. Para determinar si ya tiene uno o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster. Reemplace *`my-cluster`* con el nombre del clúster y reemplace *`my-kasten-role`* con el nombre para su rol. Este comando requiere que tenga `eksctl` instalado en su dispositivo. Si necesita usar una herramienta diferente para crear el rol y anotar la cuenta de servicio de Kubernetes, consulte Configuración de una cuenta de servicio de Kubernetes para que asuma un rol de IAM.

  ```
  eksctl create iamserviceaccount --name k10-k10 --namespace kasten-io --cluster my-cluster --role-name my-kasten-role \
      --role-only --attach-policy-arn arn:aws:iam::aws:policy/service-role/AWSLicenseManagerConsumptionPolicy --approve
  ```

- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Instalación de K10 en AWS con el complemento Amazon EKS en la documentación de Kasten.
- Información adicional: si su clúster de Amazon EKS es de una versión `1.23` de Kubernetes o posterior, debe tener el controlador CSI de Amazon EBS instalado en su clúster con `StorageClass` de forma predeterminada.

Kong

- Editor: Kong
- Nombre: `kong_konnect-ri`
- Espacio de nombres: `kong`

- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the Kong Gateway EKS Add-on en la documentación de Kong.

LeakSignal

- Editor: LeakSignal
- Nombre: `leaksignal_leakagent`
- Espacio de nombres: `leakagent`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Instalación del complemento LeakAgent en la documentación de LeakSignal.

NetApp

- Editor: NetApp
- Nombre: `netapp_trident-operator`
- Espacio de nombres: `trident`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Configurar el complemento Astra Trident EKS en la documentación de NetApp.

New Relic

- Editor: New Relic
- Nombre: `new-relic_kubernetes-operator`
- Espacio de nombres: `newrelic`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the New Relic Add-on for EKS en la documentación de New Relic.

Rafay

- Editor: Rafay
- Nombre: `rafay-systems_rafay-operator`
- Espacio de nombres: `rafay-system`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the Rafay Amazon EKS Add-on en la documentación de Rafay.

Solo.io

- Editor: Solo.io
- Nombre: `solo-io_istio-distro`
- Espacio de nombres: `istio-system`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.

- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Installing Istio en la documentación de Solo.io.

Stormforge

- Editor: Stormforge
- Nombre: `stormforge_optimize-Live`
- Espacio de nombres: `stormforge-system`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: Consulte Installing the StormForge Agent en la documentación de StormForge.

Splunk

- Editor: Splunk
- Nombre: `splunk_splunk-otel-collector-chart`
- Espacio de nombres: `splunk-monitoring`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Instalación del complemento Splunk para Amazon EKS en la documentación de Splunk.

Teleport

- Editor: Teleport
- Nombre: `teleport_teleport`

- Espacio de nombres: `teleport`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte How Teleport Works (Cómo funciona el teletransporte) en la documentación de Teleport.

Tetrate

- >Publicador: Tetrate Io
- Nombre: `tetrate-io_istio-distro`
- Espacio de nombres: `istio-system`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte el sitio web de Tetrate Istio Distro.

Upbound Universal Crossplane

- Editor: Upbound
- Nombre: `upbound_universal-crossplane`
- Espacio de nombres: `upbound-system`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte Upbound Universal Crossplane (UXP) en la documentación de Upbound.

Upwind

- Editor: Upwind
- Nombre: `upwind`
- Espacio de nombres: `upwind`
- Nombre de la cuenta de servicio: no se usa una cuenta de servicio con este complemento.
- Política de IAM administrada de AWS: no se utiliza una política administrada con este complemento.
- Permisos de IAM personalizados: los permisos personalizados no se utilizan con este complemento.
- Instrucciones de configuración y uso: consulte los pasos de la instalación en la documentación de Upwind.

Administración de los complementos de Amazon EKS

Los complementos de Amazon EKS son un conjunto seleccionado de software de complemento para clústeres de Amazon EKS. Todos los complementos de Amazon EKS:

- incluyen los parches de seguridad y correcciones de errores más recientes;
- están validados por AWS para poder trabajar con Amazon EKS;
- reducen la cantidad de trabajo necesaria para administrar el software complementario.

La AWS Management Console le notifica cuando hay una nueva versión disponible para un complemento de Amazon EKS. Solo tiene que iniciar la actualización y Amazon EKS actualiza el software del complemento en su nombre.

Para obtener una lista de los complementos disponibles, consulte Complementos de Amazon EKS disponibles en Amazon EKS. Para obtener más información sobre el campo de administración de Kubernetes, consulte Administración de campos de Kubernetes.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS.

Creación de un complemento

Puede crear un complemento de Amazon EKS mediante `eksctl`, la AWS Management Console o la AWS CLI. Si el complemento requiere un rol de IAM, consulte los detalles del complemento específico en Complementos de Amazon EKS disponibles en Amazon EKS para obtener información sobre cómo crear el rol.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Para crear el complemento de Amazon EKS con **`eksctl`**

1. Vea los nombres de los complementos disponibles para una versión de clúster. Reemplace *`1.30`* por la versión del clúster.

```
eksctl utils describe-addon-versions --kubernetes-version 1.30 | grep AddonName
```

Un ejemplo de salida sería el siguiente.

```
"AddonName": "aws-ebs-csi-driver",
                        "AddonName": "coredns",
                        "AddonName": "kube-proxy",
                        "AddonName": "vpc-cni",
                        "AddonName": "adot",
                        "AddonName": "dynatrace_dynatrace-operator",
                        "AddonName": "upbound_universal-crossplane",
                        "AddonName": "teleport_teleport",
                        "AddonName": "factorhouse_kpow",
                        [...]
```

2. Vea las versiones disponibles para el complemento que le gustaría crear. Reemplace *`1.30`* por la versión del clúster. Reemplace *`name-of-addon`* por el nombre del complemento para el que desea ver las versiones. El nombre debe ser uno de los nombres que obtuvo en los pasos anteriores.

```
eksctl utils describe-addon-versions --kubernetes-version 1.30 --name name-of-addon | grep AddonVersion
```

El siguiente resultado es un ejemplo de lo que se devuelve para el complemento denominado `vpc-cni`. Puede ver que el complemento tiene varias versiones disponibles.

```
"AddonVersions": [
    "AddonVersion": "v1.12.0-eksbuild.1",
    "AddonVersion": "v1.11.4-eksbuild.1",
    "AddonVersion": "v1.10.4-eksbuild.1",
    "AddonVersion": "v1.9.3-eksbuild.1",
```

3. Determine si el complemento que desea crear es uno de Amazon EKS o de AWS Marketplace. AWS Marketplace tiene complementos de terceros que requieren que complete pasos adicionales para crear el complemento.

```
eksctl utils describe-addon-versions --kubernetes-version 1.30 --name name-of-addon | grep ProductUrl
```

Si no se devuelve ningún resultado, el complemento es de Amazon EKS. Si se devuelve algún resultado, el complemento de AWS Marketplace. El siguiente resultado corresponde a un complemento denominado `teleport_teleport`.

```
"ProductUrl": "https://aws.amazon.com/marketplace/pp?
sku=3bda70bb-566f-4976-806c-f96faef18b26"
```

Puede obtener más información sobre el complemento en AWS Marketplace con la URL devuelta. Si el complemento requiere una suscripción, puede suscribirse al complemento a través de AWS Marketplace. Si va a crear un complemento desde AWS Marketplace, la entidad principal de IAM que utilice para crear el complemento debe tener permiso para crear el rol vinculado al servicio AWSServiceRoleForAWSLicenseManagerRole. Para obtener información sobre cómo asignar los permisos a una entidad principal de IAM, consulte Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

4. Cree un complemento de Amazon EKS. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado:

- Reemplace *my-cluster* por el nombre del clúster.
- Reemplace *name-of-addon* por el nombre del complemento que desea crear.
- Si quiere una versión del complemento anterior a la versión más reciente, sustitúyala por *latest* con el número de versión que aparece en el resultado del paso anterior que quieras usar.
- Si el complemento usa un rol de cuenta de servicio, sustitúyalo por *111122223333* con el ID de la cuenta y sustituya *role-name* por el nombre del rol. Para obtener instrucciones sobre cómo crear un rol para su cuenta de servicio, consulte la documentación del complemento que está creando. Para especificar un rol de cuenta de servicio, es necesario disponer de un proveedor de IAM OpenID Connect (OIDC) para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.

  Si el complemento no usa un rol de cuenta de servicio, elimine ***--service-account-role-arn*** **arn:aws:iam::*111122223333*:role/*role-name***.
- Este comando de ejemplo sobrescribe la configuración de cualquier versión autoadministrada existente del complemento, si la hay. Si no quiere sobrescribir la configuración de un complemento autoadministrado existente, elimine la opción *--force*. Si lo hace, y el complemento de Amazon EKS necesita sobrescribir la configuración existente de un complemento autoadministrado, se produce un error en la creación del complemento de Amazon EKS y recibe un mensaje para ayudarlo a resolver el conflicto. Antes de especificar esta opción, asegúrese de que el complemento de Amazon EKS no administra la configuración que necesita administrar, ya que dicha configuración se sobrescribe con esta opción.

  ```
  eksctl create addon --cluster my-cluster --name name-of-addon --version latest
   \
     --service-account-role-arn arn:aws:iam::111122223333:role/role-name --
  force
  ```

Puede ver una lista completa de las opciones disponibles para el comando.

```
eksctl create addon --help
```

Para obtener más información acerca de otras opciones, consulte Addons (Complementos) en la documentación de `eksctl`.

AWS Management Console

Para crear el complemento de Amazon EKS con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clusters (Clústeres) y, luego, seleccione el nombre del clúster para el que desea eliminar el complemento .
3. Elija la pestaña Complementos.
4. Escoja Obtener más complementos.
5. Elija los complementos que desea agregar al clúster. Puede agregar tantos complementos de Amazon EKS y complementos de AWS Marketplace como necesite.

   En el caso de complementos de AWS Marketplace, la entidad principal de IAM que se utilice para crear un complemento debe tener permisos para leer los derechos del complemento desde AWS LicenseManager. AWS LicenseManager requiere el rol vinculado al servicio (SLR) AWSServiceRoleForAWSLicenseManagerRole que permite que los recursos de AWS administren las licencias en su nombre. El SLR es obligatoria una sola vez, por cuenta, y no tendrá que crear SLR independientes para cada complemento ni para cada clúster. Para obtener información sobre cómo asignar los permisos a una entidad principal de IAM, consulte Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

   Si los complementos de AWS Marketplace que desea instalar no aparecen, puede buscar complementos disponibles escribiendo texto en el cuadro de búsqueda. En las Opciones de filtrado, también puede buscar por categoría, proveedor o modelo de precios y, a continuación, seleccionar los complementos en los resultados de la búsqueda. Una vez que haya seleccionado los complementos que desee instalar, seleccione Siguiente.
6. En la página Configurar las opciones de complementos seleccionados:

   - Seleccione Ver opciones de suscripción para abrir el formulario Opciones de suscripción. Revise las secciones Detalles de precios y Legal y, a continuación, pulse el botón Suscribirse para continuar.
   - En Version (Versión), seleccione la versión que desee utilizar. Recomendamos la versión marcada como la más reciente, a menos que el complemento individual que está creando

recomiende una versión diferente. Para determinar si un complemento tiene una versión recomendada, consulte la documentación del complemento que está creando.

- Si todos los complementos que seleccionó tienen la opción Requiere suscripción en Estado, seleccione Siguiente. No puede seguir configurando esos complementos hasta que se haya suscrito a ellos después de crear el clúster. Para los complementos que no tienen Requires subscription (Requiere suscripción) en Status (Estado):
  - En Select IAM role (Seleccionar rol de IAM), acepte la opción predeterminada, a menos que el complemento requiera permisos de IAM. Si el complemento requiere permisos de AWS, puede seleccionar el rol de IAM del nodo (No establecido) o un rol existente que haya creado para usarlo con el complemento. Si no hay ningún rol para seleccionar, no dispone de un rol existente. Independientemente de la opción que elija, consulte la documentación del complemento que está creando para crear una política de IAM y adjuntarla a un rol. Para seleccionar un rol de IAM es necesario que tenga un proveedor de OpenID Connect (OIDC) de IAM para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.
  - Seleccione Valores de configuración opcionales.
    - Si el complemento requiere configuración, introdúzcala en el cuadro Configuration values (Valores de configuración). Para determinar si el complemento requiere información de configuración, consulte la documentación del complemento que está creando.
    - En Conflict resolution method (Método de resolución de conflictos), seleccione una de las opciones disponibles.
  - Elija Siguiente.

7. En la página Revisar y añadir, elija Crear. Una vez finalizada la instalación del complemento, verá los complementos instalados.
8. Si alguno de los complementos que ha instalado requiere una suscripción, siga estos pasos:
   1. Selecciona el botón Subscribe (Suscribirse) en la esquina inferior derecha del complemento. Se lo redirigirá a la página del complemento en AWS Marketplace. Lea la información sobre el complemento, como Product Overview (Descripción general del producto) y Pricing Information (Información sobre precios).
   2. Seleccione el botón Continue to Subscribe (Continuar con la suscripción) en la parte superior derecha de la página del complemento.

3. Lea la sección Terms and Conditions (Condiciones generales). Si está de acuerdo con ellas, seleccione Accept Terms (Aceptar las condiciones). La suscripción puede tardar varios minutos en procesarse. Mientras se procesa la suscripción, el botón Return to Amazon EKS Console (Volver a la consola de Amazon EKS) aparece en gris.
4. Una vez que la suscripción haya terminado de procesarse, el botón Return to Amazon EKS Console (Volver a la consola de Amazon EKS) ya no aparecerá en gris. Elija el botón para volver a la pestaña Add-ons (Complementos) de la consola de Amazon EKS de su clúster.
5. Para el complemento al que se suscribió, seleccione Remove and reinstall (Eliminar y volver a instalar) y, a continuación, elija Reinstalar el complemento (Reinstall add-on). La instalación del complemento puede tardar varios minutos. Cuando finalice la instalación, podrá configurar el complemento.

AWS CLI

Requisito previo

La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

Para crear el complemento de Amazon EKS con la AWS CLI

1. Determine qué complementos están disponibles. Puede ver todos los complementos disponibles, su tipo y su editor. También puede ver la URL de los complementos que están disponibles a través de AWS Marketplace. Reemplace *`1.30`* por la versión del clúster.

```
aws eks describe-addon-versions --kubernetes-version 1.30 \
    --query 'addons[].{MarketplaceProductUrl: marketplaceInformation.productUrl,
 Name: addonName, Owner: owner Publisher: publisher, Type: type}' --output table
```

Un ejemplo de salida sería el siguiente.

```
-------------------------------------------------------------------------------------------
|
 DescribeAddonVersions
       |
+-------------------------------------------------------------
+------------------------------+------------------+-------------
+---------------------+
|                     MarketplaceProductUrl                     |
 Name              |      Owner       |  Publisher   |        Type         |
+-------------------------------------------------------------
+------------------------------+------------------+-------------
+---------------------+
|  None                                                         |  aws-ebs-csi-
driver           |  aws             |  eks         |  storage            |
|  None                                                         |  coredns
                |  aws             |  eks         |  networking         |
|  None                                                         |  kube-proxy
                |  aws             |  eks         |  networking         |
|  None                                                         |  vpc-cni
                |  aws             |  eks         |  networking         |
|  None                                                         |  adot
                |  aws             |  eks         |  observability      |
|  https://aws.amazon.com/marketplace/pp/prodview-brb73nceicv7u |
 dynatrace_dynatrace-operator |  aws-marketplace |  dynatrace   |  monitoring
      |
|  https://aws.amazon.com/marketplace/pp/prodview-uhc2iwi5xysoc |
 upbound_universal-crossplane |  aws-marketplace |  upbound     |  infra-
management   |
|  https://aws.amazon.com/marketplace/pp/prodview-hd2ydsrgqy4li |
 teleport_teleport            |  aws-marketplace |  teleport    |  policy-
management  |
|  https://aws.amazon.com/marketplace/pp/prodview-vgghgqdsplhvc |
 factorhouse_kpow             |  aws-marketplace |  factorhouse |  monitoring
      |
|  [...]                                                        |  [...]
                |  [...]           |  [...]       |  [...]              |
+-------------------------------------------------------------
+------------------------------+------------------+-------------
+---------------------+
```

El resultado puede ser diferente. En el resultado de este ejemplo, hay tres complementos diferentes disponibles de tipo `networking` y cinco complementos con un editor de tipo `eks`. Es posible que los complementos con `aws-marketplace` que aparecen en la columna `Owner` requieran una suscripción antes de poder instalarlos. Puede visitar la URL para obtener más información sobre el complemento y suscribirse a él.

2. Puede ver qué versiones están disponibles para cada complemento. Reemplace la versión *`1.30`* por la versión de su clúster y reemplace *`vpc-cni`* por el nombre de un complemento devuelto en el paso anterior.

```
aws eks describe-addon-versions --kubernetes-version 1.30 --addon-name vpc-cni \
    --query 'addons[].addonVersions[].{Version: addonVersion, Defaultversion:
 compatibilities[0].defaultVersion}' --output table
```

Un ejemplo de salida sería el siguiente.

```
-----------------------------------------
|          DescribeAddonVersions         |
+-----------------+----------------------+
| Defaultversion  |       Version        |
+-----------------+----------------------+
|  False          |  v1.12.0-eksbuild.1  |
|  True           |  v1.11.4-eksbuild.1  |
|  False          |  v1.10.4-eksbuild.1  |
|  False          |  v1.9.3-eksbuild.1   |
+-----------------+----------------------+
```

La versión con `True` que aparece en la columna `Defaultversion` es la versión con la que se creó el complemento, de forma predeterminada.

3. (Opcional) Busque las opciones de configuración del complemento elegido ejecutando el siguiente comando:

```
aws eks describe-addon-configuration --addon-name vpc-cni --addon-
version v1.12.0-eksbuild.1
```

```
{
    "addonName": "vpc-cni",
    "addonVersion": "v1.12.0-eksbuild.1",
```

```
    "configurationSchema": "{\"$ref\":\"#/definitions/VpcCni\",\"$schema
\":\"http://json-schema.org/draft-06/schema#\",\"definitions\":{\"Cri\":
{\"additionalProperties\":false,\"properties\":{\"hostPath\":{\"$ref\":
\"#/definitions/HostPath\"}},\"title\":\"Cri\",\"type\":\"object\"},\"Env
\":{\"additionalProperties\":false,\"properties\":{\"ADDITIONAL_ENI_TAGS
\":{\"type\":\"string\"},\"AWS_VPC_CNI_NODE_PORT_SUPPORT\":{\"format\":
\"boolean\",\"type\":\"string\"},\"AWS_VPC_ENI_MTU\":{\"format\":\"integer
\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_CONFIGURE_RPFILTER\":{\"format
\":\"boolean\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\":
{\"format\":\"boolean\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_EXTERNALSNAT
\":{\"format\":\"boolean\",\"type\":\"string\"},\"AWS_VPC_K8S_CNI_LOGLEVEL
\":{\"type\":\"string\"},\"AWS_VPC_K8S_CNI_LOG_FILE\":{\"type
\":\"string\"},\"AWS_VPC_K8S_CNI_RANDOMIZESNAT\":{\"type\":
\"string\"},\"AWS_VPC_K8S_CNI_VETHPREFIX\":{\"type\":\"string
\"},\"AWS_VPC_K8S_PLUGIN_LOG_FILE\":{\"type\":\"string\"},
\"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\":{\"type\":\"string\"},\"DISABLE_INTROSPECTION
\":{\"format\":\"boolean\",\"type\":\"string\"},\"DISABLE_METRICS\":{\"format
\":\"boolean\",\"type\":\"string\"},\"DISABLE_NETWORK_RESOURCE_PROVISIONING
\":{\"format\":\"boolean\",\"type\":\"string\"},\"ENABLE_POD_ENI\":{\"format
\":\"boolean\",\"type\":\"string\"},\"ENABLE_PREFIX_DELEGATION\":{\"format
\":\"boolean\",\"type\":\"string\"},\"WARM_ENI_TARGET\":{\"format\":\"integer
\",\"type\":\"string\"},\"WARM_PREFIX_TARGET\":{\"format\":\"integer\",
\"type\":\"string\"}},\"title\":\"Env\",\"type\":\"object\"},\"HostPath\":
{\"additionalProperties\":false,\"properties\":{\"path\":{\"type\":\"string\"}},
\"title\":\"HostPath\",\"type\":\"object\"},\"Limits\":{\"additionalProperties
\":false,\"properties\":{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type
\":\"string\"}},\"title\":\"Limits\",\"type\":\"object\"},\"Resources\":
{\"additionalProperties\":false,\"properties\":{\"limits\":{\"$ref\":\"#/
definitions/Limits\"},\"requests\":{\"$ref\":\"#/definitions/Limits\"}},
\"title\":\"Resources\",\"type\":\"object\"},\"VpcCni\":{\"additionalProperties
\":false,\"properties\":{\"cri\":{\"$ref\":\"#/definitions/Cri\"},\"env\":
{\"$ref\":\"#/definitions/Env\"},\"resources\":{\"$ref\":\"#/definitions/
Resources\"}},\"title\":\"VpcCni\",\"type\":\"object\"}}}"
}
```

El resultado es un esquema JSON estándar.

Este es un ejemplo de valores de configuración válidos, en formato JSON, que funcionan con el esquema anterior.

```
{
  "resources": {
    "limits": {
```

```
      "cpu": "100m"
    }
  }
}
```

Este es un ejemplo de valores de configuración válidos, en formato YAML, que funcionan con el esquema anterior.

```
resources:
  limits:
    cpu: 100m
```

4. Determine si el complemento requiere permisos de IAM. Si es así, debe (1) determinar si desea utilizar las Pod Identity de EKS o los roles de IAM para las cuentas de servicio (IRSA), (2) determinar el ARN de la función de IAM que se va a utilizar con el complemento y (3) determinar el nombre de la cuenta de servicio de Kubernetes que utiliza el complemento. Puedes encontrar esta información en la documentación o mediante la API AWS, consulta Cómo recuperar información de IAM sobre un complemento.

   - Amazon EKS sugiere usar las Pod Identity de EKS si el complemento lo admite. Esto requiere que el agente de Pod Identity esté instalado en el clúster. Para obtener más información sobre el uso de las Pod Identity con los complementos, consulte Adjunte un rol de IAM a un complemento de Amazon EKS mediante Pod Identity.
   - Si el complemento o el clúster no están configurados para las Pod Identity de EKS, utilice IRSA. Confirme que IRSA esté configurado en su clúster.
   - Consulte la documentación de complementos de Amazon EKS para determinar si el complemento requiere permisos de IAM y el nombre de la cuenta de servicio de Kubernetes asociada.

5. Cree un complemento de Amazon EKS. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado:

   - Reemplace *my-cluster* por el nombre del clúster.
   - Reemplace *vpc-cni* por un nombre de complemento devuelto en el paso anterior que desea crear.
   - Reemplace *version-number* por una versión devuelta en el paso anterior que desea usar.

- Si el complemento no requiere permisos de IAM, elimine *<service-account-configuration>*.
- Si el complemento (1) requiere permisos de IAM y (2) su clúster usa las Pod Identity de EKS, sustituya *<service-account-configuration>* por la siguiente asociación de Pod Identity. Sustituya *<service-account-name>* por el nombre de la cuenta de servicio que utiliza el complemento. Sustituya *<role-arn>* con el ARN de un rol de IAM. El rol debe tener la política de confianza requerida por las Pod Identity de EKS.
  - `--pod-identity-associations 'serviceAccount=<service-account-name>,roleArn=<role-arn>'`
- Si el complemento (1) requiere permisos de IAM y (2) su clúster usa IRSA, sustituya *<service-account-configuration>* por la siguiente configuración de IRSA. Reemplace *111122223333* por el ID de su cuenta y *role-name* por el nombre del rol de IAM existente que creó. Para obtener instrucciones sobre cómo crear un rol, consulte la documentación del complemento que está creando. Para especificar un rol de cuenta de servicio, es necesario disponer de un proveedor OpenID Connect (OIDC) de IAM para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.
  - `--service-account-role-arn arn:aws:iam::111122223333:role/role-name`
- Estos comandos de ejemplo sobrescriben la opción `--configuration-values` de cualquier versión autoadministrada existente del complemento, si la hay. Sustitúyalo por los valores de configuración que desee, como una cadena o una entrada de archivo. Si no desea proporcionar valores de configuración, elimine la opción `--configuration-values`. Si no quiere que la AWS CLI sobrescriba la configuración de un complemento autoadministrado existente, elimine la opción *--resolve-conflicts OVERWRITE*. Si lo hace, y el complemento de Amazon EKS necesita sobrescribir la configuración existente de un complemento autoadministrado, se produce un error en la creación del complemento de Amazon EKS y recibe un mensaje para ayudarlo a resolver el conflicto. Antes de especificar esta opción, asegúrese de que el complemento de Amazon EKS no administra la configuración que necesita administrar, ya que dicha configuración se sobrescribe con esta opción.

```
aws eks create-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version version-number \
```

```
    <service-account-configuration> --configuration-values '{"resources":
{"limits":{"cpu":"100m"}}}' --resolve-conflicts OVERWRITE
```

```
aws eks create-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version version-number \
    <service-account-configuration> --configuration-values 'file://example.yaml'
 --resolve-conflicts OVERWRITE
```

Para ver una lista completa de las opciones disponibles, consulte `create-addon` en la Referencia de los comandos de Amazon EKS. Si el complemento que ha creado aparece en `aws-marketplace` en la columna `Owner` de un paso anterior, es posible que no se pueda crear y que reciba un mensaje de error similar al siguiente.

```
{
    "addon": {
        "addonName": "addon-name",
        "clusterName": "my-cluster",
        "status": "CREATE_FAILED",
        "addonVersion": "version",
        "health": {
            "issues": [
                {
                    "code": "AddonSubscriptionNeeded",
                    "message": "You are currently not subscribed to this add-
on. To subscribe, visit the AWS Marketplace console, agree to the seller EULA,
 select the pricing type if required, then re-install the add-on"
[...]
```

Si recibe un error similar al del resultado anterior, visite la URL del resultado del paso anterior para suscribirse al complemento. Una vez suscrito, vuelva a ejecutar el comando `create-addon`.

Actualización de un complemento

Amazon EKS no actualiza de forma automática el complemento cuando se lanzan versiones nuevas o después de que actualice el clúster a una nueva versión secundaria de Kubernetes. Para actualizar el complemento en un clúster existente, debe iniciar la actualización. Luego de iniciar la actualización, Amazon EKS actualiza el complemento en su nombre. Antes de actualizar un complemento, consulte la documentación actual del complemento. Para obtener una lista de los

complementos disponibles, consulte Complementos de Amazon EKS disponibles en Amazon EKS. Si el complemento requiere un rol de IAM, consulte los detalles del complemento específico en Complementos de Amazon EKS disponibles en Amazon EKS para obtener información sobre cómo crear el rol.

Puede actualizar un complemento de Amazon EKS mediante `eksctl`, la AWS Management Console o la AWS CLI.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para Instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Para actualizar el complemento de Amazon EKS con **`eksctl`**

1. Determine los complementos y las versiones actuales de los complementos instalados en su clúster. Reemplace *`my-cluster`* por el nombre del clúster.

```
eksctl get addon --cluster my-cluster
```

Un ejemplo de salida sería el siguiente.

```
NAME        VERSION              STATUS  ISSUES  IAMROLE  UPDATE AVAILABLE
coredns     v1.8.7-eksbuild.2    ACTIVE  0
kube-proxy  v1.23.7-eksbuild.1   ACTIVE  0                v1.23.8-eksbuild.2
vpc-cni     v1.10.4-eksbuild.1   ACTIVE  0                v1.12.0-
eksbuild.1,v1.11.4-eksbuild.1,v1.11.3-eksbuild.1,v1.11.2-eksbuild.1,v1.11.0-
eksbuild.1
```

El resultado puede tener un aspecto diferente, según los complementos y las versiones que tenga en su clúster. Puede ver que en el resultado del ejemplo anterior, dos complementos existentes en el clúster tienen versiones más recientes disponibles en la columna `UPDATE AVAILABLE`.

2. Actualice el complemento.

   1. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario:

- Reemplace *my-cluster* por el nombre del clúster.
- Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster.
- Reemplace *vpc-cni* por un nombre de complemento devuelto en el paso anterior que desea actualizar.
- Si quiere una versión del complemento anterior a la versión más reciente, sustitúyala por *latest* con el número de versión que aparece en el resultado del paso anterior que desea usar. Algunos complementos tienen versiones recomendadas. Para obtener más información, consulte la documentación del complemento que está actualizando.
- Si el complemento usa una cuenta de servicio de Kubernetes y un rol de IAM, reemplace *111122223333* por el ID de su cuenta y el *role-name* por el nombre de un rol de IAM existente que haya creado. Para obtener instrucciones sobre cómo crear un rol, consulte la documentación del complemento que está creando. Para especificar un rol de cuenta de servicio, es necesario disponer de un proveedor OpenID Connect (OIDC) de IAM para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.

  Si el complemento no usa una cuenta de servicio de Kubernetes o un rol de IAM, elimine la línea **serviceAccountRoleARN: arn:aws:iam::*111122223333*:role/*role-name***.
- La opción *preserve* conserva los valores existentes del complemento. Si ha establecido valores personalizados para la configuración del complemento y no utiliza esta opción, Amazon EKS sobrescribe los valores con sus valores predeterminados. Si utiliza esta opción, le recomendamos que pruebe cualquier cambio de campo y valor en un clúster que no sea de producción antes de actualizar el complemento de su clúster de producción. Si cambia este valor a `overwrite`, todas las configuraciones cambiarán a los valores predeterminados de Amazon EKS. Si ha establecido valores personalizados para cualquier configuración, es posible que se sobrescriban con los valores predeterminados de Amazon EKS. Si cambia este valor a `none`, Amazon EKS no cambia el valor de ninguna configuración, pero la actualización podría fallar. Si se produce un error en la actualización, recibe un mensaje de error que lo ayuda a resolver el conflicto.

```
cat >update-addon.yaml <<EOF
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
```

```
  name: my-cluster
  region: region-code

addons:
- name: vpc-cni
  version: latest
  serviceAccountRoleARN: arn:aws:iam::111122223333:role/role-name
  resolveConflicts: preserve
EOF
```

2. Ejecute el comando modificado para crear el archivo `update-addon.yaml`.
3. Aplique el archivo de configuración al clúster.

```
eksctl update addon -f update-addon.yaml
```

Para obtener más información acerca de cómo actualizar complementos, consulte Addons (Complementos) en la documentación de `eksctl`.

AWS Management Console

Para actualizar el complemento de Amazon EKS con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clusters (Clústeres) y, a continuación, seleccione el nombre del clúster para el que desea configurar el complemento.
3. Elija la pestaña Complementos.
4. Seleccione la casilla situada en la parte superior derecha del cuadro y, a continuación, elija Edit (Editar).
5. En la página Configurar el ***nombre del complemento***:
   - Seleccione la Version (Versión) que desea utilizar. Es posible que el complemento tenga una versión recomendada. Para obtener más información, consulte la documentación del complemento que está actualizando.
   - En Seleccionar rol de IAM, puede utilizar el rol de IAM del nodo (No establecido) o un rol existente que haya creado para usarlo con el complemento. Si no hay ningún rol para seleccionar, no dispone de un rol existente. Independientemente de la opción que elija, consulte la documentación del complemento que está creando para crear una política

de IAM y adjuntarla a un rol. Para seleccionar un rol de IAM es necesario que tenga un proveedor de OpenID Connect (OIDC) de IAM para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.

- Para `Code editor`, introduzca cualquier información de configuración específica del complemento. Para obtener más información, consulte la documentación del complemento que está actualizando.
- En Conflict resolution method (Método de resolución de conflictos), seleccione una de las opciones. Si ha establecido valores personalizados para la configuración del complemento, le recomendamos que utilice la opción Preserve (Conservar). Si no elige esta opción, Amazon EKS sobrescribe sus valores con los valores predeterminados. Si utiliza esta opción, le recomendamos que pruebe cualquier cambio de campo y valor en un clúster que no sea de producción antes de actualizar el complemento de su clúster de producción.

6. Elija Actualizar.

AWS CLI

Requisito previo

La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.

Para actualizar el complemento de Amazon EKS con la AWS CLI

1. Consulte la lista de complementos instalados. Reemplace *`my-cluster`* por el nombre del clúster.

```
aws eks list-addons --cluster-name my-cluster
```

Un ejemplo de salida sería el siguiente.

```
{
    "addons": [
        "coredns",
        "kube-proxy",
        "vpc-cni"
    ]
}
```

2. Vea la versión actual del complemento que desea actualizar. Reemplace *my-cluster* por el nombre de su clúster y *vpc-cni* por el nombre del complemento que desea actualizar.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni --query
 "addon.addonVersion" --output text
```

Un ejemplo de salida sería el siguiente.

```
v1.10.4-eksbuild.1
```

3. Puede ver qué versiones del complemento se encuentran disponibles para la versión de su clúster. Reemplace *1.30* por la versión de su clúster y *vpc-cni* por el nombre del complemento que desea actualizar.

```
aws eks describe-addon-versions --kubernetes-version 1.30 --addon-name vpc-cni \
    --query 'addons[].addonVersions[].{Version: addonVersion, Defaultversion:
 compatibilities[0].defaultVersion}' --output table
```

Un ejemplo de salida sería el siguiente.

```
-----------------------------------------
|          DescribeAddonVersions         |
+-----------------+----------------------+
| Defaultversion  |       Version        |
+-----------------+----------------------+
|  False          |  v1.12.0-eksbuild.1  |
|  True           |  v1.11.4-eksbuild.1  |
|  False          |  v1.10.4-eksbuild.1  |
|  False          |  v1.9.3-eksbuild.1   |
+-----------------+----------------------+
```

La versión con `True` que aparece en la columna `Defaultversion` es la versión con la que se creó el complemento, de forma predeterminada.

4. Actualice su complemento. Copie el comando que sigue en su dispositivo. Realice las siguientes modificaciones en el comando según sea necesario y, a continuación, ejecute el comando modificado.

   - Reemplace *`my-cluster`* por el nombre del clúster.
   - Reemplace *`vpc-cni`* por el nombre del complemento que desea actualizar y que se ha devuelto en el resultado de un paso anterior.
   - Reemplace *`version-number`* por una versión devuelta en el resultado del paso anterior que a la que desea actualizar. Algunos complementos tienen versiones recomendadas. Para obtener más información, consulte la documentación del complemento que está actualizando.
   - Si el complemento usa una cuenta de servicio de Kubernetes y un rol de IAM, reemplace *`111122223333`* por el ID de su cuenta y el *`role-name`* por el nombre de un rol de IAM existente que haya creado. Para obtener instrucciones sobre cómo crear un rol, consulte la documentación del complemento que está creando. Para especificar un rol de cuenta de servicio, es necesario disponer de un proveedor OpenID Connect (OIDC) de IAM para el clúster. Para determinar si ya tiene uno para su clúster o si debe crearlo, consulte Creación de un proveedor de OIDC de IAM para su clúster.

     Si el complemento no usa una cuenta de servicio de Kubernetes o un rol de IAM, elimine la línea **`serviceAccountRoleARN: arn:aws:iam::`*`111122223333`*`:role/`*`role-name`***.
   - La opción **`--resolve-conflicts`** *`PRESERVE`* (CONSERVAR) conserva los valores existentes del complemento. Si ha establecido valores personalizados para la configuración del complemento y no utiliza esta opción, Amazon EKS sobrescribe los valores con sus valores predeterminados. Si utiliza esta opción, le recomendamos que pruebe cualquier cambio de campo y valor en un clúster que no sea de producción antes de actualizar el complemento de su clúster de producción. Si cambia este valor a `overwrite`, todas las configuraciones cambiarán a los valores predeterminados de Amazon EKS. Si ha establecido valores personalizados para cualquier configuración, es posible que se sobrescriban con los valores predeterminados de Amazon EKS. Si cambia este valor a `none`, Amazon EKS no cambia el valor de ninguna configuración, pero la

actualización podría fallar. Si se produce un error en la actualización, recibe un mensaje de error que lo ayuda a resolver el conflicto.

- Si desea eliminar toda la configuración personalizada, realice la actualización mediante la opción *--configuration-values '{}'*. Esto vuelve a establecer toda la configuración personalizada a los valores predeterminados. Si no quiere cambiar su configuración personalizada, no proporcione el indicador *--configuration-values*. Si desea ajustar una configuración personalizada, sustituya *{}* por los nuevos parámetros. Para ver una lista de parámetros, consulte el paso de visualización del esquema de configuración en la sección sobre cómo crear un complemento.

```
aws eks update-addon --cluster-name my-cluster --addon-name vpc-cni --addon-
version version-number \
    --service-account-role-arn arn:aws:iam::111122223333:role/role-name --
configuration-values '{}' --resolve-conflicts PRESERVE
```

5. Compruebe el estado de la actualización. Reemplace *my-cluster* por el nombre de su clúster y *vpc-cni* por el nombre del complemento que está actualizando.

```
aws eks describe-addon --cluster-name my-cluster --addon-name vpc-cni
```

Un ejemplo de salida sería el siguiente.

```
{
    "addon": {
        "addonName": "vpc-cni",
        "clusterName": "my-cluster",
        "status": "UPDATING",
[...]
```

Cuando la actualización se completa, el estado cambia a `ACTIVE`.

Eliminación de un complemento

Cuando elimina un complemento de Amazon EKS:

- No hay tiempo de inactividad para la funcionalidad que proporciona el complemento.

- Si utiliza los roles de IAM para las cuentas de servicio (IRSA) y el complemento tiene una función de IAM asociada, esta no se elimina.
- Si utiliza las Pod Identity, se eliminarán todas las asociaciones de Pod Identity que sean propiedad del complemento. Si especifica la opción `--preserve` a AWS CLI, las asociaciones se conservan.
- Amazon EKS deja de administrar la configuración del complemento.
- La consola deja de avisarle cuando haya nuevas versiones disponibles.
- No puede actualizar el complemento con ninguna herramienta o API de AWS.
- Puede optar por dejar el software de complemento en el clúster para poder autoadministrar el software de complemento o puede eliminar el software de complemento del clúster. Solo debe eliminar el complemento de software si ninguno de los recursos del clúster depende de la funcionalidad que proporciona el complemento.

Puede eliminar un complemento de Amazon EKS de su clúster mediante `eksctl`, la AWS Management Console o la AWS CLI.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para Instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Para eliminar el complemento de Amazon EKS con **`eksctl`**

1. Determine los complementos instalados en su clúster. Reemplace *`my-cluster`* por el nombre del clúster.

```
eksctl get addon --cluster my-cluster
```

Un ejemplo de salida sería el siguiente.

```
NAME        VERSION              STATUS  ISSUES  IAMROLE  UPDATE AVAILABLE
coredns     v1.8.7-eksbuild.2    ACTIVE  0
kube-proxy  v1.23.7-eksbuild.1   ACTIVE  0
vpc-cni     v1.10.4-eksbuild.1   ACTIVE  0
[...]
```

El resultado puede tener un aspecto diferente, según los complementos y las versiones que tenga en su clúster.

2. Elimine el complemento. Reemplace *my-cluster* por el nombre de su clúster y *name-of-add-on* por el nombre del complemento que obtuvo en la salida del paso anterior que desea eliminar. Si elimina la opción ***--preserve***, además de que Amazon EKS deja de administrar el complemento, se quita el software del complemento se quita el software de complemento del clúster.

```
eksctl delete addon --cluster my-cluster --name name-of-addon --preserve
```

AWS Management Console

Cómo eliminar el complemento de Amazon EKS con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, seleccione Clusters (Clústeres) y, a continuación, seleccione el nombre del clúster para el que desea eliminar el complemento de Amazon EKS.
3. Elija la pestaña Complementos.
4. Seleccione la casilla de verificación situada en la parte superior derecha del cuadro de complemento y, a continuación, elija Remove (Eliminar). Seleccione Preserve on cluster (Conservar en el clúster) si desea que Amazon EKS deje de administrar la configuración del complemento, pero desea retener el software de complemento en el clúster para poder autoadministrar toda la configuración del complemento. Escriba el nombre del complemento y, a continuación, seleccione Remove (Eliminar).

AWS CLI

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para Instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Para eliminar el complemento de Amazon EKS con la AWS CLI

1. Consulte la lista de complementos instalados. Reemplace *my-cluster* por el nombre del clúster.

```
aws eks list-addons --cluster-name my-cluster
```

Un ejemplo de salida sería el siguiente.

```
{
    "addons": [
        "coredns",
        "kube-proxy",
        "vpc-cni",
        "name-of-addon"
    ]
}
```

2. Elimine el complemento instalado. Reemplace *my-cluster* por el nombre de su clúster y *name-of-add-on* por el nombre del complemento que desea eliminar. Al eliminar **--preserve**, se elimina el software de complemento del clúster.

```
aws eks delete-addon --cluster-name my-cluster --addon-name name-of-addon --
preserve
```

A continuación se muestra el ejemplo abreviado de salida.

```
{
    "addon": {
        "addonName": "name-of-add-on",
        "clusterName": "my-cluster",
        "status": "DELETING",
[...]
```

3. Compruebe el estado de la tarea de eliminación. Reemplace *my-cluster* por el nombre de su clúster y *name-of-addon* por el nombre del complemento que está eliminando.

```
aws eks describe-addon --cluster-name my-cluster --addon-name name-of-addon
```

El resultado del ejemplo una vez eliminado el complemento es el siguiente.

```
An error occurred (ResourceNotFoundException) when calling the DescribeAddon
 operation: No addon: name-of-addon found in cluster: my-cluster
```

Recuperación de la compatibilidad de las versiones del complemento

Use la API de `describe-addon-verisions` para enumerar las versiones disponibles de los complementos de EKS y las versiones de Kubernetes que admite cada versión del complemento.

Recuperación de la compatibilidad de las versiones del complemento (AWS CLI)

1. Compruebe que la AWS CLI esté instalada y funcione con `aws sts get-caller-identity`. Si este comando no funciona, obtenga información sobre cómo comenzar a utilizar la AWS CLI.
2. Determine el nombre del complemento del que desea recuperar la información de compatibilidad de versiones; por ejemplo, `amazon-cloudwatch-observability`.
3. Determine la versión de Kubernetes del clúster, como `1.28`.
4. Use la AWS CLI para recuperar versiones de complementos compatibles con la versión de Kubernetes del clúster.

```
aws eks describe-addon-versions --addon-name amazon-cloudwatch-observability --
kubernetes-version 1.29
```

Un ejemplo de salida sería el siguiente.

```
{
    "addons": [
        {
            "addonName": "amazon-cloudwatch-observability",
            "type": "observability",
            "addonVersions": [
                {
                    "addonVersion": "v1.5.0-eksbuild.1",
                    "architecture": [
                        "amd64",
                        "arm64"
                    ],
                    "compatibilities": [
                        {
                            "clusterVersion": "1.28",
                            "platformVersions": [
                                "*"
                            ],
                            "defaultVersion": true
                        }
```

```
                    ],
[...]
```

Este resultado muestra que la versión complementaria `v1.5.0-eksbuild.1` es compatible con la versión `1.28` del clúster de Kubernetes.

Administración de campos de Kubernetes

Los complementos de Amazon EKS se instalan en el clúster mediante configuraciones estándar de prácticas recomendadas. Para obtener más información acerca de cómo agregar un complemento de Amazon EKS al clúster, consulte Complementos de Amazon EKS.

Es posible que desee personalizar la configuración de un complemento de Amazon EKS para habilitar características avanzadas. Amazon EKS utiliza la característica de aplicación del lado del servidor de Kubernetes para habilitar la administración de un complemento por parte de Amazon EKS sin sobrescribir la configuración de los ajustes que Amazon EKS no administra. Para obtener más información, consulte Server-side Apply (Aplicación del lado del servidor) en la documentación de Kubernetes. Para ello, Amazon EKS administra un conjunto mínimo de campos para cada complemento que instala. Puede modificar todos los campos que no estén administrados por Amazon EKS u otro proceso de plano de control de Kubernetes, como `kube-controller-manager`, sin problema.

> **Important**
>
> La modificación de un campo administrado por Amazon EKS impide que Amazon EKS administre el complemento y puede provocar que los cambios se sobrescriban cuando se actualiza un complemento.

Consulte el estado de administración de campos

Puede utilizar `kubectl` para ver qué campos administra Amazon EKS en cualquier complemento de Amazon EKS.

Consulte el estado de administración de un campo

1. Determine el complemento que desea examinar. Para ver todas las `deployments` y los `DaemonSets` implementados en el clúster, consulte Vea los recursos de Kubernetes.

2. Para ver los campos administrados por un complemento, ejecute el siguiente comando:

```
kubectl get type/add-on-name -n add-on-namespace -o yaml
```

Por ejemplo, puede ver los campos administrados para el complemento CoreDNS con el siguiente comando.

```
kubectl get deployment/coredns -n kube-system -o yaml
```

La administración de campos se muestra en la siguiente sección de la salida devuelta.

```
[...]
managedFields:
  - apiVersion: apps/v1
    fieldsType: FieldsV1
    fieldsV1:
[...]
```

> **Note**
>
> Si no ve `managedFields` en la salida, agregue **`--show-managed-fields`** al comando y ejecútelo de nuevo. La versión de `kubectl` que utiliza determina si los campos administrados se devuelven de forma predeterminada.

Descripción de la sintaxis de administración de campos en la API de Kubernetes

Cuando se visualizan los detalles de un objeto de Kubernetes, los campos administrados y los no administrados se devuelven en la salida. Los campos administrados pueden ser de uno de los tipos siguientes:

- Completamente administrado: Amazon EKS administra todas las claves del campo. Las modificaciones de algún valor provocan un conflicto.
- Parcialmente administrado: Amazon EKS administra algunas claves del campo. Solo las modificaciones de las claves administradas explícitamente por Amazon EKS provocan un conflicto.

Ambos tipos de campos se etiquetan con `manager: eks`.

Cada clave es o bien un `.` que representa el campo en sí, que siempre se asigna a un conjunto vacío, o una cadena que representa un subcampo o elemento. La salida para la administración del campo consta de los siguientes tipos de declaraciones:

- `f:`*`name`*, donde *`name`* es el nombre de un campo de una lista.
- `k:`*`keys`*, donde *`keys`* es un mapa de los campos de un elemento de lista.
- `v:`*`value`*, donde *`value`* es el valor exacto con formato JSON de un elemento de lista.
- `i:`*`index`*, donde *`index`* es la posición de un elemento en la lista.

Las siguientes partes de salida para el complemento CoreDNS ilustran las declaraciones anteriores:

- Campos completamente administrados: si para un campo administrado se ha especificado `f:` (campo), pero no `k:` (clave), se administra todo el campo. Las modificaciones a los valores de este campo provocan un conflicto.

  En la siguiente salida, puede ver que el contenedor llamado `coredns` está administrado por `eks`. Los subcampos `args`, `image` y `imagePullPolicy` también están administrados por `eks`. Las modificaciones de algún valor de estos campos provocan un conflicto.

  ```
  [...]
  f:containers:
    k:{"name":"coredns"}:
    .: {}
    f:args: {}
    f:image: {}
    f:imagePullPolicy: {}
  [...]
  manager: eks
  [...]
  ```

- Campos parcialmente administrados: si una clave administrada tiene especificado un valor, se administran las claves declaradas para ese campo. La modificación de las claves especificadas provoca un conflicto.

  En el siguiente resultado, puede ver que `eks` administra los volúmenes `config-volume` y `tmp` establecidos con la clave `name`.

  ```
  [...]
  f:volumes:
  ```

```
  k:{"name":"config-volume"}:
    .: {}
    f:configMap:
      f:items: {}
      f:name: {}
    f:name: {}
  k:{"name":"tmp"}:
    .: {}
    f:name: {}
[...]
manager: eks
[...]
```

- Adición de claves a campos parcialmente administrados: si solo se administra un valor de clave específico, puede agregar claves adicionales, como argumentos, a un campo sin provocar ningún conflicto. Si agrega claves adicionales, asegúrese de que el campo no esté administrado primero. Agregar o modificar cualquier valor administrado provoca un conflicto.

  En el siguiente resultado, puede ver que tanto la clave `name` como el campo `name` están administrados. Agregar o modificar cualquier nombre de contenedor provoca un conflicto con esta clave administrada.

```
[...]
f:containers:
  k:{"name":"coredns"}:
[...]
    f:name: {}
[...]
manager: eks
[...]
```

Adjunte un rol de IAM a un complemento de Amazon EKS mediante Pod Identity

Algunos complementos de Amazon EKS necesitan permisos de rol de IAM para llamar a las API de AWS. Por ejemplo, el complemento CNI de Amazon VPC llama a determinadas API de AWS para configurar los recursos de red de su cuenta. Es necesario conceder permiso a estos complementos mediante el IAM de AWS. Más específicamente, la cuenta de servicio del pod que ejecuta el complemento debe estar asociada a un rol de IAM con una política de IAM suficiente.

La forma recomendada de conceder permisos AWS para agrupar cargas de trabajo es mediante la característica Pod Identity de Amazon EKS. Puede usar una asociación de Pod Identity para asignar la cuenta de servicio de un complemento a un rol de IAM. Si un pod usa una cuenta de servicio que tiene una asociación, Amazon EKS establece las variables de entorno en los contenedores del pod. Las variables de entorno configuran los SDK de AWS, incluida la AWS de CLI, para usar las credenciales de la Pod Identity de EKS. Más información sobre las Pod Identity de EKS.

Los complementos de Amazon EKS pueden ayudar a gestionar el ciclo de vida de las asociaciones de Pod Identity correspondientes al complemento. Por ejemplo, puede crear o actualizar un complemento de Amazon EKS y la asociación de Pod Identity necesaria en una sola llamada a la API. Amazon EKS también proporciona una API para recuperar las políticas de IAM sugeridas.

Uso sugerido:

1. Confirme que el agente de Pod Identity de Amazon EKS esté configurado en su clúster.
2. Determine si el complemento que desea instalar requiere permisos de IAM mediante la operación `describe-addon-versions` AWS CLI. Si el indicador `requiresIamPermissions` es `true`, entonces debe usar la operación `describe-addon-configurations` para determinar los permisos que necesita el complemento. La respuesta incluye una lista de políticas de IAM gestionadas sugeridas.
3. Recupere el nombre de la cuenta de servicio de Kubernetes y la política de IAM sugerida mediante la operación de CLI `describe-addon-configuration`. Evalúe el alcance de la política sugerida en función de sus requisitos de seguridad.
4. Cree un rol de IAM con la política de permisos sugerida y la política de confianza que exige Pod Identity. Para obtener más información, consulte Creación de la asociación de Pod Identity de EKS.
5. Cree o actualice el complemento de Amazon EKS con la CLI. Especifique al menos una asociación de Pod Identity. Una asociación de Pod Identity es (1) el nombre de una cuenta de servicio de Kubernetes y (2) el ARN de un rol de IAM.

Consideraciones:

- Las asociaciones de Pod Identity creadas con las API de los complementos son propiedad del complemento correspondiente. Si elimina el complemento, también se eliminará la asociación de Pod Identity. Para evitar esta eliminación en cascada, utilice la opción `preserve` al eliminar un complemento mediante AWS CLI o la API. Si es necesario, también puede actualizar o eliminar directamente la asociación de Pod Identity. Los complementos no pueden asumir la propiedad de

las asociaciones de Pod Identity existentes. Debe eliminar la asociación existente y volver a crearla mediante una operación de creación o actualización de complementos.

- Amazon EKS recomienda usar asociaciones de Pod Identity para administrar los permisos de IAM para los complementos. El método anterior, los roles de IAM para cuentas de servicio (IRSA), aún se admite. Puede especificar tanto un `serviceAccountRoleArn` de IRSA como una asociación de Pod Identity para un complemento. Si el agente del Pod Identity de EKS está instalado en el clúster, `serviceAccountRoleArn` se ignorará y EKS utilizará la asociación de Pod Identity proporcionada. Si Pod Identity no está habilitada, se utilizará `serviceAccountRoleArn`.
- Si actualiza las asociaciones de Pod Identity de un complemento existente, Amazon EKS inicia un reinicio continuo de los pods del complemento.

Recupere la información de IAM sobre un complemento

Puede utilizar AWS CLI para determinar (1) si un complemento requiere permisos de IAM y (2) una política de IAM sugerida para ese complemento.

Recuperar información de IAM sobre un complemento de Amazon EKS (AWS CLI)

1. Determine el nombre del complemento que quiere instalar y la versión de Kubernetes del clúster. Obtener más información sobre los complementos de Amazon EKS.
2. Utilice AWS CLI para determinar si el complemento requiere permisos de IAM.

   ```
   aws eks describe-addon-versions \
   --addon-name <addon-name> \
   --kubernetes-version <kubernetes-version>
   ```

   Por ejemplo:

   ```
   aws eks describe-addon-versions \
   --addon-name aws-ebs-csi-driver \
   --kubernetes-version 1.30
   ```

   Revise la siguiente salida de ejemplo. Tenga en cuenta que `requiresIamPermissions` es `true` y la versión predeterminada del complemento. Debe especificar la versión del complemento al recuperar la política de IAM recomendada.

   ```
   {
   ```

```
    "addons": [
        {
            "addonName": "aws-ebs-csi-driver",
            "type": "storage",
            "addonVersions": [
                {
                    "addonVersion": "v1.31.0-eksbuild.1",
                    "architecture": [
                        "amd64",
                        "arm64"
                    ],
                    "compatibilities": [
                        {
                            "clusterVersion": "1.30",
                            "platformVersions": [
                                "*"
                            ],
                            "defaultVersion": true
                        }
                    ],
                    "requiresConfiguration": false,
                    "requiresIamPermissions": true
                },
[...]
```

3. Si el complemento requiere permisos de IAM, utilice AWS CLI para recuperar una política de IAM recomendada.

```
aws eks describe-addon-configuration \
--query podIdentityConfiguration \
--addon-name <addon-name> \
--addon-version <addon-version>
```

Por ejemplo:

```
aws eks describe-addon-configuration \
--query podIdentityConfiguration \
--addon-name aws-ebs-csi-driver \
--addon-version v1.31.0-eksbuild.1
```

Revise la siguiente salida. Anote el `recommendedManagedPolicies`.

```
[
    {
        "serviceAccount": "ebs-csi-controller-sa",
        "recommendedManagedPolicies": [
            "arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy"
        ]
    }
]
```

4. Cree un rol de IAM y adjunte la política administrada recomendada. Como alternativa, revise la política administrada y reduzca los permisos según corresponda. Revisar las instrucciones para crear un rol de IAM para usarlo con las Pod Identity de EKS.

Actualice el complemento con el rol de IAM

Actualice un complemento de Amazon EKS para usar una asociación de Pod Identity (AWS CLI)

1. Determine:
   - `cluster-name`— El nombre del clúster de EKS en el que se va a instalar el complemento.
   - `addon-name`— El nombre del complemento de Amazon EKS que se va a instalar.
   - `service-account-name`— El nombre de la cuenta de servicio de Kubernetes utilizada por el complemento.
   - `iam-role-arn`— El ARN de un rol de IAM con permisos suficientes para el complemento. El rol de IAM debe tener la política de confianza requerida para la Pod Identity de EKS.
2. Actualice el complemento mediante la AWS de CLI. También puede especificar las asociaciones de Pod Identity al crear un complemento, utilizando la misma sintaxis `--pod-identity-assocations`. Tenga en cuenta que si especifica las asociaciones de Pod Identity al actualizar un complemento, se sobrescriben todas las anteriores.

   ```
   aws eks update-addon --cluster-name <cluster-name> \
   --addon-name <addon-name> \
   --pod-identity-associations 'serviceAccount=<service-account-name>,roleArn=<role-arn>'
   ```

   Por ejemplo:

```
aws eks update-addon --cluster-name mycluster \
--addon-name aws-ebs-csi-driver \
--pod-identity-associations 'serviceAccount=ebs-csi-controller-
sa,roleArn=arn:aws:iam::123456789012:role/StorageDriver'
```

3. Compruebe que se haya creado la asociación de Pod Identity:

```
aws eks list-pod-identity-associations --cluster-name <cluster-name>
```

Si la operación se realiza correctamente, debería ver un resultado similar al siguiente. Anote el ARN del propietario del complemento EKS.

```
{
    "associations": [
        {
            "clusterName": "mycluster",
            "namespace": "kube-system",
            "serviceAccount": "ebs-csi-controller-sa",
            "associationArn": "arn:aws:eks:us-
west-2:123456789012:podidentityassociation/mycluster/a-4wvljrezsukshq1bv",
            "associationId": "a-4wvljrezsukshq1bv",
            "ownerArn": "arn:aws:eks:us-west-2:123456789012:addon/mycluster/aws-
ebs-csi-driver/9cc7ce8c-2e15-b0a7-f311-426691cd8546"
        }
    ]
}
```

Elimine las asociaciones del complemento

Elimine todas las asociaciones de Pod Identity de un complemento de Amazon EKS (AWS CLI)

1. Determine:
   - `cluster-name`— El nombre del clúster de EKS en el que se va a instalar el complemento.
   - `addon-name`— El nombre del complemento de Amazon EKS que se va a instalar.
2. Actualice el complemento para especificar una matriz vacía de asociaciones de Pod Identity.

```
aws eks update-addon --cluster-name <cluster-name> \
--addon-name <addon-name> \
```

```
--pod-identity-associations "[]"
```

Solucione los problemas de Pod Identity de los complementos de EKS

Si sus complementos encuentran errores al intentar realizar operaciones de la API de AWS, SDK o CLI, confirme lo siguiente:

- El agente de Pod Identity está instalado en el clúster.
  - Revisar cómo configurar el Pod Identity Agent.
- El complemento tiene una asociación de Pod Identity válida.
  - Use AWS CLI para recuperar las asociaciones del nombre de la cuenta de servicio que utiliza el complemento.

```
aws eks list-pod-identity-associations --cluster-name <cluster-name>
```

- El rol de IAM previsto tiene la política de confianza requerida para las Pod Identity de EKS.
  - Use AWS CLI para recuperar la política de confianza de un complemento.

```
aws iam get-role --role-name <role-name> --query Role.AssumeRolePolicyDocument
```

- El rol de IAM previsto tiene los permisos necesarios para el complemento.
  - Use AWS CloudTrail para revisar los eventos `AccessDenied` o `UnauthorizedOperation`.
- El nombre de la cuenta de servicio en la asociación de Pod Identity coincide con el nombre de la cuenta de servicio que utiliza el complemento.
  - Revisar la documentación del complemento para determinar el nombre de la cuenta de servicio.

Verificación de una imagen de contenedor durante la implementación

Si utiliza AWS Signer y desea verificar las imágenes de contenedores firmadas en el momento de la implementación, puede utilizar una de las siguientes soluciones:

- Gatekeeper y Ratify: utilice Gatekeeper como el controlador de admisión y a Ratify configurado con un complemento de AWS Signer como enlace web para validar las firmas.

- Kyverno: un motor de políticas de Kubernetes configurado con un complemento de AWS Signer para validar firmas.

> **Note**
>
> Antes de comprobar las firmas de las imágenes del contenedor, configure el almacén de confianza de Notation y la política de confianza, según lo exija el controlador de admisión seleccionado.

Formación en machine learning con Elastic Fabric Adapter

En este tema se describe cómo integrar Elastic Fabric Adapter (EFA) con los Pods implementados en el clúster de Amazon EKS. Elastic Fabric Adapter (EFA) es una interfaz de red para instancias de Amazon EC2 que le permite ejecutar aplicaciones que requieren altos niveles de comunicaciones entre nodos a escala en AWS. Su interfaz de hardware de bypass del sistema operativo diseñada a medida mejora el rendimiento de las comunicaciones entre instancias, lo que es fundamental para ajustar la escala de estas aplicaciones. Con EFA, las aplicaciones de High Performance Computing (HPC, informática de alto rendimiento) que utilizan la Message Passing Interface (MPI, interfaz de paso de mensajes) y las aplicaciones de Machine Learning (ML) que utilizan NVIDIA Collective Communications Library (NCCL, Biblioteca de comunicación colectiva de NVIDIA) pueden aumentar su escala a miles de CPU o GPU. Como resultado, obtiene el rendimiento de las aplicaciones de los clústeres HPC en las instalaciones con la elasticidad y flexibilidad bajo demanda de la nube de AWS. La integración de Elastic Fabric Adapter (EFA) con aplicaciones que se ejecutan en clústeres de Amazon EKS puede reducir el tiempo necesario para completar cargas de trabajo de formación distribuidas a gran escala sin tener que agregar instancias adicionales al clúster. Para obtener más información sobre EFA, consulte Elastic Fabric Adapter.

El complemento de EFA descrito en este tema es totalmente compatible con las instancias de Amazon EC2 P4d, que representan la tecnología de vanguardia en machine learning distribuido en la nube. Cada instancia `p4d.24xlarge` tiene ocho GPU NVIDIA A100 y GPUDirectRDMA de 400 Gbps sobre EFA. GPUDirectRDMA le permite tener comunicación directa de GPU a GPU a través de nodos con bypass de CPU, lo que aumenta el ancho de banda de comunicación colectiva y reduce la latencia. La integración de Amazon EKS y EFA con instancias P4d proporciona un método perfecto para aprovechar la instancia informática de Amazon EC2 de mayor rendimiento para la formación de machine learning distribuido.

Requisitos previos

- Un clúster existente de Amazon EKS. Si no dispone de un clúster existente, utilice una de nuestras guías Introducción a Amazon EKS para crear uno. El clúster debe implementarse en una VPC que tenga al menos una subred privada con suficientes direcciones IP disponibles para implementar nodos en ella. La subred privada debe tener acceso a Internet saliente proporcionado por un dispositivo externo, como una puerta de enlace NAT.

  Si planea usar `eksctl` para crear su grupo de nodos, `eksctl` también puede crear un clúster por usted.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalación de la AWS CLI en el directorio de inicio en la Guía del usuario de AWS CloudShell.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- Debe tener instalada la versión Amazon VPC CNI plugin for Kubernetes `1.7.10` o posterior antes de lanzar nodos de trabajo que admitan varios Elastic Fabric Adapters, como la `p4d.24xlarge`. Para obtener más información sobre cómo actualizar Amazon VPC CNI plugin for Kubernetes, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS.

Crear un grupo de nodos

El procedimiento siguiente ayuda a crear un grupo de nodos con un grupo de nodos con respaldo de `p4d.24xlarge` con interfaces EFA y RDMA GPUDirect. También lo ayuda a ejecutar una prueba de ejemplo de NVIDIA Collective Communications Library (NCCL) para descubrir el rendimiento

NCCL de varios nodos mediante el uso de EFA. El ejemplo se puede utilizar en una plantilla para la formación de aprendizaje profundo distribuida en Amazon EKS mediante EFA.

1. Determine qué tipos de instancias de Amazon EC2 compatibles con EFA están disponibles en la Región de AWS en la que desea implementar los nodos. Reemplace *region-code* por la Región de AWS en la que desea implementar su grupo de nodos.

```
aws ec2 describe-instance-types --region region-code --filters Name=network-
info.efa-supported,Values=true \
    --query "InstanceTypes[*].[InstanceType]" --output text
```

   Al implementar nodos, el tipo de instancia que desea implementar debe estar disponible en la Región de AWS en el que está su clúster.

2. Determine en qué zonas de disponibilidad está disponible el tipo de instancia que desea implementar. En este tutorial, el tipo de instancia `p4d.24xlarge` se utiliza el tipo de instancia y se debe devolver en la salida de la Región de AWS que especificó en el paso anterior. Cuando implemente nodos en un clúster de producción, sustituya *p4d.24xlarge* con cualquier tipo de instancia que se devolvió en el paso anterior.

```
aws ec2 describe-instance-type-offerings --region region-code --location-type
 availability-zone --filters Name=instance-type,Values=p4d.24xlarge \
    --query 'InstanceTypeOfferings[*].Location' --output text
```

   Un ejemplo de salida sería el siguiente.

```
us-west-2a    us-west-2c    us-west-2b
```

   Tenga en cuenta las zonas de disponibilidad devueltas para su uso en pasos posteriores. Al implementar nodos en un clúster, la VPC debe tener subredes con direcciones IP disponibles en una de las zonas de disponibilidad que se muestran en el resultado.

3. Cree un grupo de nodos mediante `eksctl` o la AWS CLI y AWS CloudFormation.

   eksctl

   Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

1. Copie el siguiente contenido en un archivo denominado *`efa-cluster.yaml`*. Reemplace los *`example values`* por los de su propiedad. Puede reemplazar *`p4d.24xlarge`* por una instancia diferente, pero si lo hace, asegúrese de que los valores de `availabilityZones` sean zonas de disponibilidad que se devolvieron para el tipo de instancia en el paso uno.

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-efa-cluster
  region: region-code
  version: "1.XX"

iam:
  withOIDC: true

availabilityZones: ["us-west-2a", "us-west-2c"]

managedNodeGroups:
  - name: my-efa-ng
    instanceType: p4d.24xlarge
    minSize: 1
    desiredCapacity: 2
    maxSize: 3
    availabilityZones: ["us-west-2a"]
    volumeSize: 300
    privateNetworking: true
    efaEnabled: true
```

2. Cree un grupo de nodos administrados en un clúster existente.

```
eksctl create nodegroup -f efa-cluster.yaml
```

Si no dispone de un clúster existente, puede ejecutar el siguiente comando para crear un clúster y el grupo de nodos.

```
eksctl create cluster -f efa-cluster.yaml
```

> **Note**
>
> Como el tipo de instancia que se usa en este ejemplo tiene GPU,`eksctl` instala automáticamente el complemento de dispositivo NVIDIA Kubernetes en cada instancia por ti.

AWS CLI and AWS CloudFormation

Existen varios requisitos para las redes de EFA, entre ellos la creación de un grupo de seguridad específico de EFA, la creación de un grupo de ubicación de Amazon EC2 y la creación de una plantilla de lanzamiento que especifique una o varias interfaces de EFA e incluya la instalación del controlador de EFA como parte de los datos del usuario de Amazon EC2. Para obtener más información sobre los requisitos de EFA, consulte Introducción a EFA y MPI en la Guía del usuario de Amazon EC2. Los siguientes pasos crean todo esto. Sustituya los *valores de ejemplo* por sus propios valores.

1. Defina algunas variables que se utilizan en pasos posteriores. Sustituya todos los *example values* con los suyos propios. Reemplace *my-cluster* por el nombre de su clúster existente. El valor de `node_group_resources_name` se utiliza más tarde para crear una pila de AWS CloudFormation. El valor de `node_group_name` se utiliza más tarde para crear el grupo de nodos en el clúster.

   ```
   cluster_name="my-cluster"
   cluster_region="region-code"
   node_group_resources_name="my-efa-nodegroup-resources"
   node_group_name="my-efa-nodegroup"
   ```

2. Identifique una subred privada de la VPC que se encuentre en la misma zona de disponibilidad en la que está disponible el tipo de instancia en el que desea implementar.

   a. Recupere la versión del clúster y guárdela en una variable para utilizarla en un paso posterior.

      ```
      cluster_version=$(aws eks describe-cluster \
          --name $cluster_name \
      ```

```
    --query "cluster.version" \
    --output text)
```

b. Recupere el ID de VPC en el que se encuentra el clúster y guárdelo en una variable para utilizarlo en un paso posterior.

```
vpc_id=$(aws eks describe-cluster \
    --name $cluster_name \
    --query "cluster.resourcesVpcConfig.vpcId" \
    --output text)
```

c. Recupere el ID del grupo de seguridad del plano de control del clúster y guárdelo en una variable para utilizarlo en un paso posterior.

```
control_plane_security_group=$(aws eks describe-cluster \
    --name $cluster_name \
    --query "cluster.resourcesVpcConfig.clusterSecurityGroupId" \
    --output text)
```

d. Obtenga la lista de ID de subred de la VPC que se encuentran en una zona de disponibilidad devuelta en el paso uno.

```
aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=$vpc_id" "Name=availability-
zone,Values=us-west-2a" \
    --query 'Subnets[*].SubnetId' \
    --output text
```

Si no se devuelve ninguna salida, pruebe con una zona de disponibilidad diferente que se devuelve en el paso uno. Si ninguna de las subredes se encuentra en una zona de disponibilidad devuelta en el paso 1, deberá crear una subred en una zona de disponibilidad devuelta en el paso uno. Si no tiene espacio en la VPC para crear otra subred, puede agregar un bloque de CIDR a la VPC y crear subredes en el nuevo bloque de CIDR, o bien crear un nuevo clúster en una VPC nueva.

e. Para determinar si la subred es una subred privada, verifique la tabla de enrutamiento de la subred.

```
aws ec2 describe-route-tables \
    --filter Name=association.subnet-id,Values=subnet-0d403852a65210a29 \
    --query "RouteTables[].Routes[].GatewayId" \
```

```
    --output text
```

Un ejemplo de salida sería el siguiente.

```
local
```

Si la salida es `local igw-`*`02adc64c1b72722e2`*, entonces la subred es una subred pública. Debe seleccionar una subred privada en una zona de disponibilidad devuelta en el paso uno. Una vez que ha identificado una subred privada, tenga en cuenta su ID para su uso en un paso posterior.

f. Establezca una variable con el ID de subred privada del paso anterior para utilizarla en pasos posteriores.

```
subnet_id=your-subnet-id
```

3. Descargue la plantilla de AWS CloudFormation.

```
curl -O https://raw.githubusercontent.com/aws-samples/aws-efa-eks/main/
cloudformation/efa-p4d-managed-nodegroup.yaml
```

4. Copie el siguiente texto en el equipo. Reemplace *`p4d.24xlarge`* por un tipo de instancia del paso uno. Reemplace *`subnet-0d403852a65210a29`* con el ID de la subred privada que identificó en el paso 2.b.v. Reemplace *`path-to-downloaded-cfn-template`* por la ruta al archivo `efa-p4d-managed-nodegroup.yaml` que descargó en el paso anterior. Reemplace *`your-public-key-name`* con el nombre de su clave pública. Una vez que haya realizado las sustituciones, ejecute el comando modificado.

```
aws cloudformation create-stack \
    --stack-name ${node_group_resources_name} \
    --capabilities CAPABILITY_IAM \
    --template-body file://path-to-downloaded-cfn-template \
    --parameters \
       ParameterKey=ClusterName,ParameterValue=${cluster_name} \
       ParameterKey=ClusterControlPlaneSecurityGroup,ParameterValue=
${control_plane_security_group} \
       ParameterKey=VpcId,ParameterValue=${vpc_id} \
       ParameterKey=SubnetId,ParameterValue=${subnet_id} \
       ParameterKey=NodeGroupName,ParameterValue=${node_group_name} \
       ParameterKey=NodeImageIdSSMParam,ParameterValue=/aws/service/eks/
optimized-ami/${cluster_version}/amazon-linux-2-gpu/recommended/image_id \
```

```
        ParameterKey=KeyName,ParameterValue=your-public-key-name \
        ParameterKey=NodeInstanceType,ParameterValue=p4d.24xlarge
```

5. Determine cuándo se implementa la pila que implementó en el paso anterior.

```
aws cloudformation wait stack-create-complete --stack-name
 $node_group_resources_name
```

   No hay salida del comando anterior, pero el símbolo del intérprete de comandos no regresa hasta que se crea la pila.

6. Cree su grupo de nodos utilizando los recursos creados por la pila de AWS CloudFormation en el paso anterior.

   a. Recupere información de la pila de AWS CloudFormation implementada y almacénela en variables.

```
node_instance_role=$(aws cloudformation describe-stacks \
    --stack-name $node_group_resources_name \
    --query='Stacks[].Outputs[?OutputKey==`NodeInstanceRole`].OutputValue'
 \
    --output text)
launch_template=$(aws cloudformation describe-stacks \
    --stack-name $node_group_resources_name \
    --query='Stacks[].Outputs[?OutputKey==`LaunchTemplateID`].OutputValue'
 \
    --output text)
```

   b. Cree un grupo de nodos administrados que utilice la plantilla de lanzamiento y el rol de IAM de nodo que se crearon en el paso anterior.

```
aws eks create-nodegroup \
    --cluster-name $cluster_name \
    --nodegroup-name $node_group_name \
    --node-role $node_instance_role \
    --subnets $subnet_id \
    --launch-template id=$launch_template,version=1
```

   c. Confirme que se crearon los nodos.

```
aws eks describe-nodegroup \
   --cluster-name ${cluster_name} \
```

```
    --nodegroup-name ${node_group_name} | jq -r .nodegroup.status
```

No continúe hasta que el estado devuelto por el comando anterior sea `ACTIVE`. Los nodos pueden tardar varios minutos en estar listos.

7. Si elige un tipo de instancia de GPU, debe implementar el Plugin de dispositivo NVIDIA para Kubernetes. Reemplace *vX.X.X* con la versión Plugin de dispositivo NVidia/K8S deseada antes de ejecutar el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/vX.X.X/nvidia-device-plugin.yml
```

4. Implemente el complemento de EFA del dispositivo de Kubernetes.

El complemento de EFA del dispositivo de Kubernetes detecta y anuncia interfaces de EFA como recursos asignables a Kubernetes. Una aplicación puede consumir el tipo de recurso ampliado `vpc.amazonaws.com/efa` en una especificación de solicitud de Pod al igual que la CPU y la memoria. Para obtener más información, consulte Consumir recursos ampliados en la documentación de Kubernetes. Una vez solicitado, el complemento asigna automáticamente una interfaz de EFA al Pod y la monta en él. El uso del complemento del dispositivo simplifica la configuración de EFA y no requiere que un Pod se ejecute en modo privilegiado.

```
helm repo add eks https://aws.github.io/eks-chart
helm install aws-efa-k8s-device-plugin --namespace kube-system eks/aws-efa-k8s-device-plugin
```

(Opcional) Implementar una aplicación de ejemplo compatible con EFA

Implementar el operador MPI de Kubeflow

Para las pruebas NCCL puede aplicar el operador MPI de Kubeflow. El operador Message Passing Interface (MPI, interfaz de paso de mensajes) facilita la ejecución de la formación distribuida de estilo AllReduce en Kubernetes. Para obtener más información, consulte Operador de MPI en GitHub.

```
kubectl apply -f https://raw.githubusercontent.com/kubeflow/mpi-operator/master/deploy/v2beta1/mpi-operator.yaml
```

Ejecutar la prueba de rendimiento NCCL de varios nodos para verificar GPUDirectRDMA/EFA

Para verificar el rendimiento de NCCL con GPUDirectRDMA sobre EFA, ejecute la prueba de rendimiento de NCCL estándar. Para obtener más información, consulte el repositorio oficial Pruebas de NCCL en GitHub. Puede utilizar la muestra Dockerfile que ya está incluida en esta prueba para NVIDIA CUDA `11.2` y la última versión de EFA.

Como alternativa, puede descargar una imagen de Docker de AWS disponible desde un repositorio de Amazon ECR.

> **Important**
>
> Una consideración importante que se requiere para adoptar EFA con Kubernetes es configurar y administrar Huge Pages como un recurso en el clúster. Para obtener más información, consulte Administración de Huge Pages en la documentación de Kubernetes. Las instancias de Amazon EC2 con el controlador de EFA instalado preasignan 5128 Huge Pages de 2 M, que puede solicitar como recursos para utilizar en sus especificaciones de trabajo.

Complete los pasos que se indican a continuación para ejecutar una prueba de rendimiento de NCCL de dos nodos. En el trabajo de prueba NCCL de ejemplo, cada trabajo solicita ocho GPU, 5210 MiB de HugePages de 2 Mi, cuatro EFA y 8000 MiB de memoria, lo que significa que cada trabajo consume todos los recursos de una instancia `p4d.24xlarge`.

1. Cree el trabajo de pruebas NCCL.

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/aws-efa-eks/main/examples/simple/nccl-efa-tests.yaml
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   mpijob.kubeflow.org/nccl-tests-efa creado
   ```

2. Consulte su ejecución Pods.

   ```
   kubectl get pods
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME                              READY   STATUS     RESTARTS   AGE
   ```

```
nccl-tests-efa-launcher-nbql9  0/1     Init:0/1   0          2m49s
nccl-tests-efa-worker-0          1/1     Running    0          2m49s
nccl-tests-efa-worker-1          1/1     Running    0          2m49s
```

El operador MPI crea un Pod de lanzamiento y dos Pods de trabajo (uno en cada nodo).

3. Consulte el registro del Pod de `efa-launcher`. Sustituya *wzr8j* con el valor de su salida.

```
kubectl logs -f nccl-tests-efa-launcher-nbql9
```

Para obtener más ejemplos, consulte Muestras de EFA del repositorio de Amazon EKS en GitHub.

Inferencia de machine learning mediante el uso de AWS Inferentia

En este tema se describe cómo crear un clúster de Amazon EKS con nodos que ejecutan instancias Inf1 de Amazon EC2 e implementar (opcionalmente) una aplicación de ejemplo. Las instancias Inf1 de Amazon EC2 están equipadas con chips de AWS Inferentia creados a medida por AWS para proporcionar un alto rendimiento y una inferencia de menor costo en la nube. Los modelos de machine learning se implementan en contenedores utilizando AWS Neuron, un kit de desarrollo de software (SDK) especializado que consta de un compilador, tiempo de ejecución y herramientas de perfilado que optimizan el rendimiento de inferencia de machine learning de los chips Inferentia. AWS Neuron admite marcos de machine learning populares como TensorFlow, PyTorch y MXNet.

> **Note**
>
> Los identificadores lógicos de los dispositivos Neuron deben ser contiguos. Si un Pod que solicita varios dispositivos Neuron está programado en un tipo de instancia `inf1.6xlarge` o `inf1.24xlarge` (que tiene más de un dispositivo Neuron), ese Pod no se iniciará si el programador de Kubernetes selecciona identificadores de dispositivo no contiguos. Para obtener más información, consulte Device logical IDs must be contiguous (Los ID lógicos de los dispositivos deben ser continuos) en GitHub.

Requisitos previos

- Debe tener `eksctl` instalado en el equipo. Si no lo tiene instalado, consulte la sección de Instalación en la documentación de `eksctl`.

- Debe tener `kubectl` instalado en el equipo. Para obtener más información, consulte Instalación o actualización del `kubectl`.
- (Opcional) Debe tener instalado `python3` en su equipo. Si no lo tiene instalado, consulte Descargas de Python para obtener instrucciones de instalación.

Crear un clúster

Para crear un clúster con nodos Inf1 de instancia de Amazon EC2

1. Cree un clúster con nodos Inf1 de instancias de Amazon EC2. Puede reemplazar *`inf1.2xlarge`* con cualquier tipo de instancia Inf1. La utilidad `eksctl` detecta que va a lanzar un grupo de nodos con un tipo de instancia `Inf1` e iniciará sus nodos utilizando uno de las AMI optimizadas de Amazon Linux acelerado de Amazon EKS.

   > **Note**
   >
   > No puede usar roles de IAM para cuentas de servicio con TensorFlow Serving.

   ```
   eksctl create cluster \
       --name inferentia \
       --region region-code \
       --nodegroup-name ng-inf1 \
       --node-type inf1.2xlarge \
       --nodes 2 \
       --nodes-min 1 \
       --nodes-max 4 \
       --ssh-access \
       --ssh-public-key your-key \
       --with-oidc
   ```

   > **Note**
   >
   > Tenga en cuenta el valor de la siguiente línea de la salida. Se usa en un paso posterior (opcional).

```
[9]  adding identity "arn:aws:iam::111122223333:role/
eksctl-inferentia-nodegroup-ng-in-NodeInstanceRole-FI7HIYS3BS09" to auth
 ConfigMap
```

Al lanzar un grupo de nodos con instancias `Inf1`, `eksctl` instala automáticamente el complemento de dispositivo de Kubernetes de AWS Neuron. Este complemento anuncia dispositivos Neuron como un recurso del sistema al programador de Kubernetes, que puede solicitar un contenedor. Además de las políticas de IAM de nodos de Amazon EKS predeterminadas, se agrega la política de acceso de solo lectura de Amazon S3 para que la aplicación de ejemplo, que se trató en un paso posterior, pueda cargar un modelo entrenado desde Amazon S3.

2. Asegúrese de que todos los Pods se hayan iniciado correctamente.

```
kubectl get pods -n kube-system
```

Salida abreviada:

```
NAME                                   READY   STATUS    RESTARTS   AGE
[...]
neuron-device-plugin-daemonset-6djhp   1/1     Running   0          5m
neuron-device-plugin-daemonset-hwjsj   1/1     Running   0          5m
```

(Opcional) Implementar una imagen de aplicación TensorFlow Serving

Un modelo entrenado debe compilarse en un destino de Inferentia antes de poder implementarlo en instancias Inferentia. Para continuar, necesitará un modelo TensorFlow optimizado para Neuron guardado en Amazon S3. Si aún no tiene un SavedModel, siga el tutorial para crear un modelo ResNet50 compatible con Neuron y cargue el SavedModel resultante a S3. ResNet-50 es un modelo de machine learning popular utilizado para tareas de reconocimiento de imágenes. Para obtener más información sobre cómo compilar modelos de Neuron, consulte The AWS Inferentia Chip with DLAMI en la Guía para desarrolladores de AWS Deep Learning AMI.

El manifiesto de implementación de ejemplo administra un contenedor de servicio de inferencia preconstruido para TensorFlow proporcionado por AWS Deep Learning Containers. Dentro del contenedor está el Runtime (Tiempo de ejecución) de AWS Neuron y la aplicación TensorFlow

Serving. Una lista completa de contenedores de aprendizaje profundo preconstruidos optimizados para Neuron se mantiene en GitHub en Available Images (Imágenes disponibles). Al iniciarse, el DLC obtendrá su modelo de Amazon S3, lanzará Neuron TensorFlow Serving con el modelo guardado y esperará las solicitudes de predicción.

El número de dispositivos de Neuron asignados a su aplicación de servicio se puede ajustar cambiando el recurso `aws.amazon.com/neuron` en el yaml de implementación. Tenga en cuenta que la comunicación entre TensorFlow Serving y el tiempo de ejecución de Neuron ocurre a través de GRPC, lo que requiere pasar la capacidad de `IPC_LOCK` al contenedor.

1. Agregue la política de IAM `AmazonS3ReadOnlyAccess` al rol de instancia de nodo que se creó en el paso 1 de Crear un clúster . Esto es necesario para que la aplicación de muestra pueda cargar un modelo entrenado desde Amazon S3.

   ```
   aws iam attach-role-policy \
       --policy-arn arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
       --role-name eksctl-inferentia-nodegroup-ng-in-NodeInstanceRole-FI7HIYS3BS09
   ```

2. Cree un archivo denominado `rn50_deployment.yaml` con el siguiente contenido. Actualice el código de región y la ruta del modelo para que coincida con la configuración deseada. El nombre del modelo es para fines de identificación cuando un cliente realiza una solicitud al servidor TensorFlow. En este ejemplo se utiliza un nombre de modelo para que coincida con un script de cliente ResNet50 de ejemplo que se utilizará en un paso posterior para enviar solicitudes de predicción.

   ```
   aws ecr list-images --repository-name neuron-rtd --registry-id 790709498068 --
   region us-west-2
   ```

   ```
   kind: Deployment
   apiVersion: apps/v1
   metadata:
     name: eks-neuron-test
     labels:
       app: eks-neuron-test
       role: master
   spec:
     replicas: 2
     selector:
       matchLabels:
         app: eks-neuron-test
   ```

```
      role: master
  template:
    metadata:
      labels:
        app: eks-neuron-test
        role: master
    spec:
      containers:
        - name: eks-neuron-test
          image: 763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference-
neuron:1.15.4-neuron-py37-ubuntu18.04
          command:
            - /usr/local/bin/entrypoint.sh
          args:
            - --port=8500
            - --rest_api_port=9000
            - --model_name=resnet50_neuron
            - --model_base_path=s3://your-bucket-of-models/resnet50_neuron/
          ports:
            - containerPort: 8500
            - containerPort: 9000
          imagePullPolicy: IfNotPresent
          env:
            - name: AWS_REGION
              value: "us-east-1"
            - name: S3_USE_HTTPS
              value: "1"
            - name: S3_VERIFY_SSL
              value: "0"
            - name: S3_ENDPOINT
              value: s3.us-east-1.amazonaws.com
            - name: AWS_LOG_LEVEL
              value: "3"
          resources:
            limits:
              cpu: 4
              memory: 4Gi
              aws.amazon.com/neuron: 1
            requests:
              cpu: "1"
              memory: 1Gi
          securityContext:
            capabilities:
              add:
```

```
                - IPC_LOCK
```

3. Implemente el modelo.

```
kubectl apply -f rn50_deployment.yaml
```

4. Cree un archivo denominado `rn50_service.yaml` con el siguiente contenido. Los puertos HTTP y gRPC están abiertos para aceptar solicitudes de predicción.

```
kind: Service
apiVersion: v1
metadata:
  name: eks-neuron-test
  labels:
    app: eks-neuron-test
spec:
  type: ClusterIP
  ports:
    - name: http-tf-serving
      port: 8500
      targetPort: 8500
    - name: grpc-tf-serving
      port: 9000
      targetPort: 9000
  selector:
    app: eks-neuron-test
    role: master
```

5. Cree un servicio de Kubernetes para su aplicación de distribución de modelos de TensorFlow.

```
kubectl apply -f rn50_service.yaml
```

(Opcional) Haga predicciones contra su servicio de distribución de TensorFlow

1. Para realizar pruebas localmente, reenvíe el puerto gRPC al servicio `eks-neuron-test`.

```
kubectl port-forward service/eks-neuron-test 8500:8500 &
```

2. Cree un script de Python denominado `tensorflow-model-server-infer.py` con el siguiente contenido. Este script ejecuta la inferencia a través de gRPC, que es el marco de trabajo de servicio.

```
import numpy as np
   import grpc
   import tensorflow as tf
   from tensorflow.keras.preprocessing import image
   from tensorflow.keras.applications.resnet50 import preprocess_input
   from tensorflow_serving.apis import predict_pb2
   from tensorflow_serving.apis import prediction_service_pb2_grpc
   from tensorflow.keras.applications.resnet50 import decode_predictions

   if __name__ == '__main__':
       channel = grpc.insecure_channel('localhost:8500')
       stub = prediction_service_pb2_grpc.PredictionServiceStub(channel)
       img_file = tf.keras.utils.get_file(
           "./kitten_small.jpg",
           "https://raw.githubusercontent.com/awslabs/mxnet-model-server/master/
docs/images/kitten_small.jpg")
       img = image.load_img(img_file, target_size=(224, 224))
       img_array = preprocess_input(image.img_to_array(img)[None, ...])
       request = predict_pb2.PredictRequest()
       request.model_spec.name = 'resnet50_inf1'
       request.inputs['input'].CopyFrom(
           tf.make_tensor_proto(img_array, shape=img_array.shape))
       result = stub.Predict(request)
       prediction = tf.make_ndarray(result.outputs['output'])
       print(decode_predictions(prediction))
```

3. Ejecute el script para enviar predicciones a su servicio.

```
python3 tensorflow-model-server-infer.py
```

Un ejemplo de salida sería el siguiente.

```
[[(u'n02123045', u'tabby', 0.68817204), (u'n02127052', u'lynx', 0.12701613),
 (u'n02123159', u'tiger_cat', 0.08736559), (u'n02124075', u'Egyptian_cat',
 0.063844085), (u'n02128757', u'snow_leopard', 0.009240591)]]
```

Administración de clústeres

En este capítulo, se explican los siguientes temas para ayudarlo a administrar el clúster. También puede ver la información de sus Recursos de Kubernetes con la AWS Management Console.

- El Panel de Kubernetes es una interfaz de usuario de uso general basada en la web para clústeres de Kubernetes. Permite a los usuarios administrar las aplicaciones que se ejecutan en el clúster y solucionar sus problemas, así como administrar el propio clúster. Para obtener más información, consulte el repositorio GitHub del Panel de Kubernetes.
- Instalación del servidor de métricas de Kubernetes: el servidor de métricas de Kubernetes es un agregador de datos de uso de recursos en el clúster. No se implementa de forma predeterminada en el clúster, pero lo utilizan los complementos de Kubernetes, como el panel de Kubernetes y Escalador automático de pods horizontales. En este tema aprenderá a instalar el servidor de métricas.
- Utilizar Helm con Amazon EKS: el administrador de paquetes Helm para Kubernetes lo ayuda a instalar y administrar aplicaciones en su clúster de Kubernetes. Este tema lo ayudará a instalar y ejecutar los archivos binarios de Helm para que pueda instalar y administrar gráficos mediante la CLI de Helm en su equipo local.
- Etiquetado de los recursos de Amazon EKS Para ayudarlo a administrar los recursos de Amazon EKS, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. En este tema se describe qué son las etiquetas y cómo crearlas.
- Cuotas de servicio de Amazon EKS La cuenta de AWS tiene cuotas predeterminadas, antes denominadas límites, para cada servicio de AWS. Obtenga más información sobre las cuotas para Amazon EKS y sobre cómo aumentarlas.

Supervisión de costos

La supervisión de los costos es un aspecto esencial de la administración de los clústeres de Kubernetes en Amazon EKS. Al obtener visibilidad de los costos de su clúster, puede optimizar el uso de los recursos, establecer presupuestos y tomar decisiones basadas en datos sobre sus implementaciones. Amazon EKS ofrece dos soluciones de supervisión de costos, cada una con sus propias ventajas únicas, para ayudarle a rastrear y asignar sus costos de manera efectiva:

Datos de asignación de costos divididos de facturación de AWS para Amazon EKS: esta característica nativa se integra perfectamente con la consola de facturación de AWS, lo que le

permite analizar y asignar los costos mediante la misma interfaz y los mismos flujos de trabajo familiares que usa para otros servicios de AWS. Con la asignación de costos divididos, puede obtener información sobre sus costos de Kubernetes directamente junto con sus otros gastos de AWS, lo que le permite optimizar los costos de manera integral en todo su entorno de AWS. También puede aprovechar las características de facturación de AWS existentes, como Categorías de costos y Detección de anomalías en los costos, para mejorar aún más sus capacidades de administración de costos. Para obtener más información, consulte Understanding split cost allocation data en la Guía del usuario de facturación de AWS.

Kubecost: Amazon EKS es compatible con Kubecost, una herramienta de supervisión de costos de Kubernetes. Kubecost ofrece un enfoque nativo de Kubernetes y rico en características para la supervisión de costos, que proporciona desgloses de costos detallados por recursos de Kubernetes, recomendaciones de optimización de costos y paneles e informes listos para usar. Kubecost también recupera datos de precios precisos al integrarlos con el informe de costos y uso de AWS, lo que garantiza que tenga una visión precisa de los costos de Amazon EKS. Obtenga información sobre cómo instalar Kubecost.

Facturación de AWS: asignación de costos divididos

Supervisión de costos mediante datos de asignación de costos de AWS divididos para Amazon EKS

Puede utilizar los datos de asignación de costos de AWS divididos para Amazon EKS a fin de obtener una visibilidad pormenorizada de los costos de sus clústeres de Amazon EKS. Esto le permite analizar, optimizar y reembolsar los costos y el uso de sus aplicaciones de Kubernetes. Los costos de las aplicaciones se asignan a las unidades de negocio y los equipos individuales en función de los recursos de memoria y CPU de Amazon EC2 que consume la aplicación de Kubernetes. Los datos de asignación de costos divididos para Amazon EKS ofrecen visibilidad del costo por pod y le permiten agregar los datos de costos por pod mediante el espacio de nombres, el clúster y otras primitivas de Kubernetes. Los siguientes son ejemplos de primitivas de Kubernetes que puede usar para analizar los datos de asignación de costos de Amazon EKS.

- Cluster name (Nombre del clúster)
- Implementación
- Espacio de nombres
- Nodo
- Nombre de carga de trabajo
- Tipo de carga de trabajo

Para obtener más información sobre el uso de los datos de asignación de costos divididos, consulte Understanding split cost allocation data en la Guía del usuario de facturación de AWS.

Configuración de informes de costos y usos

Puede activar los datos de asignación de costos divididos para ECS en la consola de administración de costos, AWS Command Line Interface, o en los SDK de AWS.

Utilice lo siguiente para los datos de asignación de costos divididos:

1. Active los datos de asignación de costos divididos. Para obtener más información, consulte Enabling split cost allocation data en la Guía del usuario de AWS Cost and Usage Report.
2. Incluya los datos en un informe nuevo o existente.
3. Vea el informe. Puede utilizar la consola de administración de costo y facturación o visualizar los archivos de los informes en Amazon Simple Storage Service.

Kubecost

Amazon EKS admite Kubecost, que puede utilizar para supervisar sus costos desglosados por los recursos de Kubernetes que incluyen Pods, nodos, espacios de nombres y etiquetas. Como administrador de plataforma y líder financiero de Kubernetes, puede usar Kubecost para visualizar un desglose de los cargos de Amazon EKS, asignar costos y aplicar cargos a las unidades organizativas, como los equipos de aplicaciones. Puede proporcionar a sus equipos internos y unidades de negocio datos de costes transparentes y precisos basados en su factura de AWS real. Además, también puede obtener recomendaciones personalizadas para la optimización de costos en función de su entorno de infraestructura y los patrones de uso dentro de sus clústeres. Para obtener más información sobre Kubecost, consulte la documentación de Kubecost.

Amazon EKS ofrece un paquete optimizado por AWS de Kubecost para obtener visibilidad de los costos del clúster. Puede utilizar sus acuerdos de soporte de AWS existentes para obtener soporte.

Requisitos previos

- Un clúster existente de Amazon EKS. Para implementar uno, consulte Introducción a Amazon EKS. El clúster debe tener nodos de Amazon EC2 porque no se puede ejecutar Kubecost en los nodos de Fargate.
- La herramienta de línea de comandos de `kubectl` está instalada en su dispositivo o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a

la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.

- Versión 3.9.0 o posterior de Helm configurada en su dispositivo o AWS CloudShell. Para instalar o actualizar Helm, consulte the section called “Utilizar Helm”.
- Si la versión del clúster es `1.23` o una posterior, debe haber instalado antes the section called “Controlador CSI de Amazon EBS” en el clúster.

Para instalar Kubecost

1. Determine la versión de Kubecost que desea instalar. Puede ver las versiones disponibles en kubecost/cost-analyzer en la galería pública de Amazon ECR. Para obtener más información sobre la compatibilidad de las versiones de Kubecost y Amazon EKS, consulte Requisitos del entorno en la documentación de Kubecost.
2. Instale Kubecost con el siguiente comando. Sustituya *`kubecost-version`* con el valor obtenido de ECR, como *`1.108.1`*.

   ```
   helm upgrade -i kubecost oci://public.ecr.aws/kubecost/cost-analyzer --
   version kubecost-version \
       --namespace kubecost --create-namespace \
       -f https://raw.githubusercontent.com/kubecost/cost-analyzer-helm-chart/develop/
   cost-analyzer/values-eks-cost-monitoring.yaml
   ```

   Kubecost publica nuevas versiones con regularidad. Puede actualizar su versión mediante `helm upgrade`. De manera predeterminada, la instalación incluye un servidor de Prometheus local y `kube-state-metrics`. Puede personalizar su implementación para usar el Servicio administrado de Amazon para Prometheus siguiendo la documentación de Integración con la supervisión de costos de Amazon EKS. Para obtener una lista de todas las opciones que puede configurar, consulte el archivo de configuración de muestra en GitHub.
3. Asegúrese de que los Pods necesarios estén en ejecución.

   ```
   kubectl get pods -n kubecost
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME                                          READY   STATUS    RESTARTS   AGE
   kubecost-cost-analyzer-b9788c99f-5vj5b        2/2     Running   0          3h27m
   ```

```
kubecost-kube-state-metrics-99bb8c55b-bn2br   1/1     Running   0          3h27m
kubecost-prometheus-server-7d9967bfc8-9c8p7   2/2     Running   0          3h27m
```

4. En el dispositivo, habilite el reenvío de puertos para exponer el panel de control de Kubecost.

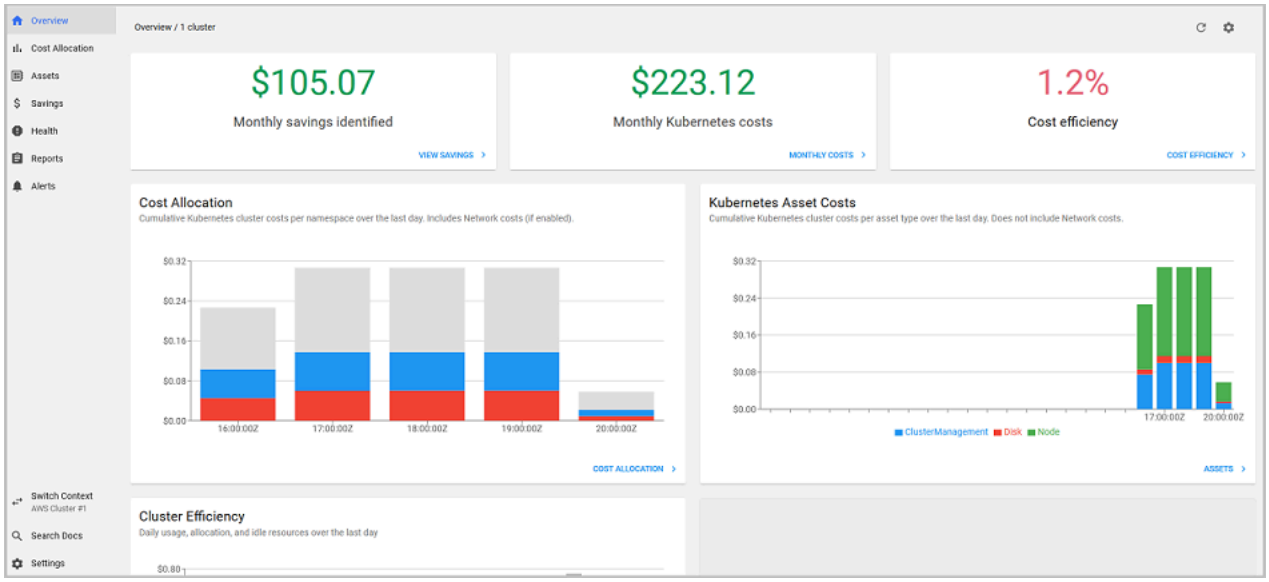
```
kubectl port-forward --namespace kubecost deployment/kubecost-cost-analyzer 9090
```

También puede utilizar el AWS Load Balancer Controller para exponer Kubecost y usar Amazon Cognito para la autenticación, autorización y administración de usuarios. Para obtener más información, consulte Cómo usar el equilibrador de carga de aplicación y Amazon Cognito para autenticar usuarios de las aplicaciones web de Kubernetes.

5. En el mismo dispositivo en el que hizo el paso anterior, abra un navegador web e ingrese la siguiente dirección.

```
http://localhost:9090
```

Verá la página de descripción general de Kubecost en su navegador. Es posible que Kubecost tarde entre 5 y 10 minutos en recopilar las métricas. Puede ver sus gastos de Amazon EKS, incluidos los costos acumulados del clúster, los costos de los activos asociados de Kubernetes y los gastos mensuales agregados.



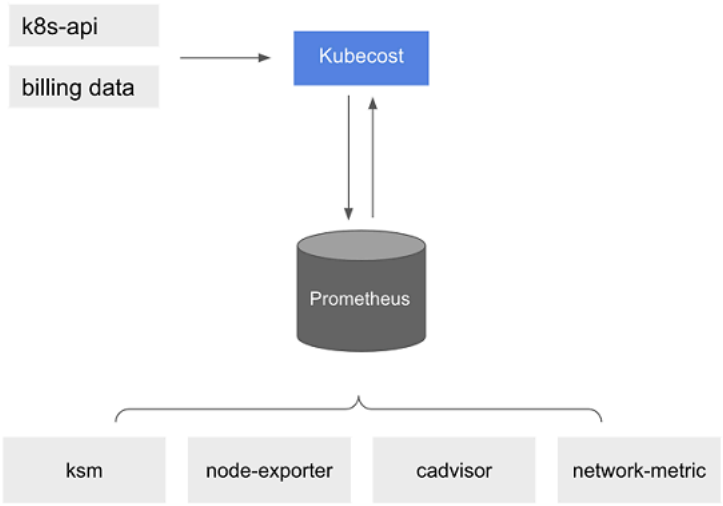
6. Para hacer un seguimiento de los costos del clúster, etiquete sus recursos de Amazon EKS para la facturación. Para obtener más información, consulte Etiquetado de los recursos para facturación.

También puede ver la siguiente información al seleccionarla en el panel izquierdo:

- Cost allocation (Asignación de costos): vea los costos mensuales de Amazon EKS y los costos acumulados para cada uno de sus espacios de nombres y otras dimensiones durante los últimos siete días. Esto es útil para entender qué partes de su aplicación están contribuyendo al gasto de Amazon EKS.
- Assets (Activos): vea los costos de los activos de infraestructura de AWS que se asocian a sus recursos de Amazon EKS.

Características adicionales

- Export cost metrics (Métricas de costos de exportación): la supervisión de costos optimizada de Amazon EKS se implementa con Kubecost y Prometheus, que es un sistema de supervisión de código abierto y una base de datos de serie temporal. Kubecost lee la métrica de Prometheus y luego calcula la asignación de costos y vuelve a escribir las métricas en Prometheus. El módulo frontal de Kubecost lee las métricas de Prometheus y las muestra en la interfaz de usuario de Kubecost. La arquitectura se ilustra en el siguiente diagrama.



Con Prometheus preinstalado, puede escribir consultas para ingerir datos de Kubecost en su actual sistema de inteligencia empresarial para su posterior análisis. También puede utilizarlo como origen de datos para su panel de control actual de Grafana para mostrar los costos del clúster de Amazon EKS con los que sus equipos internos están familiarizados. Para obtener más información sobre cómo escribir consultas de Prometheus, consulte el archivo `readme` de

Configuración de Prometheus en GitHub o utilice los modelos JSON de Grafana de ejemplo en el repositorio de GitHub de Kubecost como referencia.

- Integración de AWS Cost and Usage Report: para calcular la asignación de costos de un clúster de Amazon EKS, Kubecost recupera la información pública de precios de los Servicios de AWS y recursos de AWS desde la API de lista de precios de AWS. También puede integrar Kubecost con AWS Cost and Usage Report para mejorar la precisión de la información de precios específica de su Cuenta de AWS. Esta información incluye programas de descuento para empresas, uso de instancias reservadas, Savings Plans y uso puntual. Para más información sobre el funcionamiento de la integración de AWS Cost and Usage Report, consulte Integración de facturación de AWS Cloud en la documentación de Kubecost.

Remove Kubecost

Puede eliminar Kubecost del clúster con los siguientes comandos.

```
helm uninstall kubecost --namespace kubecost
kubectl delete ns kubecost
```

Preguntas frecuentes

Consulte las siguientes preguntas y respuestas frecuentes sobre el uso de Kubecost con Amazon EKS.

¿Cuál es la diferencia entre el paquete personalizado de Kubecost y la versión gratuita de Kubecost (también conocido como OpenCost)?

AWS y Kubecost han colaborado para ofrecer una versión personalizada de Kubecost. Esta versión incluye un subconjunto de características comerciales sin cargo adicional. Consulte la siguiente tabla para ver las características que se incluyen en el paquete personalizado de Kubecost.

Característica	Nivel gratuito de Kubecost	Paquete personalizado Kubecost de Amazon EKS optimizado	Kubecost Enterprise
Implementación	Alojado por el usuario	Alojado por el usuario	Alojado por el usuario u hospedado por Kubecost (SaaS)

Característica	Nivel gratuito de Kubecost	Paquete personalizado Kubecost de Amazon EKS optimizado	Kubecost Enterprise
Número de clústeres compatibles	Sin límite	Sin límite	Sin límite
Bases de datos admitidas	Prometheus local	Prometheus local o Amazon Managed Service for Prometheus	Prometheus, Amazon Managed Service for Prometheus, Cortex o Thanos
Soporte de retención de bases de datos	15 días	Datos históricos ilimitados	Datos históricos ilimitados
Retención de API de Kubecost (ETL)	15 días	15 días	Datos históricos ilimitados
Visibilidad de costos del clúster	Clústeres individuales	Varios clústeres unificados	Varios clústeres unificados
Visibilidad en la nube híbrida	-	Clústeres de Amazon EKS y Amazon EKS Anywhere	Compatibilidad con multinubes y nubes híbridas
Alertas e informes periódicos	-	Soporte para alertas de eficiencia, alertas de presupuesto, alertas de cambio de gastos y más	Soporte para alertas de eficiencia, alertas de presupuesto, alertas de cambio de gastos y más
Informes guardados	-	Informes con datos de 15 días	Informes que utilizan datos históricos ilimitados

Característica	Nivel gratuito de Kubecost	Paquete personalizado Kubecost de Amazon EKS optimizado	Kubecost Enterprise
Integración de facturación en la nube	Necesario para cada clúster individual	Soporte de precios personalizado para AWS (incluidos varios clústeres y múltiples cuentas)	Soporte de precios personalizado para AWS(incluidos varios clústeres y múltiples cuentas)
Recomendaciones de guardado	Información sobre un único clúster	Información sobre un único clúster	Información sobre múltiples clústeres
Gobernanza: auditorías	-	-	Audite los eventos de costos históricos
Compatibilidad con inicio de sesión único (SSO)	-	Compatible con Amazon Cognito	Okta, Auth0, PingID, KeyCloak
Control de acceso basado en roles (RBAC) con **2.0** SAML	-	-	Okta, Auth0, PingID, Keycloak
Formación e incorporación empresarial	-	-	Servicio completo de formación e incorporación de FinOps

Qué es la característica de retención de API (ETL) de Kubecost?

La característica ETL de Kubecost agrega y organiza las métricas para mostrar la visibilidad de los costos en varios niveles de granularidad (como `namespace-level`, `pod-level`, y `deployment-level`). Para el paquete personalizado de Kubecost, los clientes obtienen datos e información de las métricas de los últimos 15 días.

¿Qué es la característica de alertas e informes periódicos? ¿Qué alertas e informes incluye?

Las alertas de Kubecost permiten a los equipos recibir actualizaciones de gasto en tiempo real de Kubernetes, así como el gasto en la nube. Los informes periódicos permiten a los equipos recibir vistas personalizadas del historial de Kubernetes y gastos en la nube. Ambos se pueden configurar mediante el UI de Kubecost o valores de Helm. Admiten el correo electrónico, Slack y Microsoft Teams.

¿Qué incluyen los informes guardados?

Los informes guardados de Kubecost son vistas predefinidas de las métricas de costes y eficiencia. Incluyen el costo por clúster, espacio de nombres, etiqueta y más.

¿Qué es la integración de facturación en la nube?

La integración con las API de facturación de AWS permite a Kubecost mostrar los costos fuera del clúster (como Amazon S3). Además, permite a Kubecost conciliar las predicciones integradas de Kubecost en el clúster con datos de facturación reales para tener en cuenta el uso puntual, los planes de ahorro y los descuentos empresariales.

¿Qué incluyen las recomendaciones de ahorro?

Kubecost proporciona información y automatización para ayudar a los usuarios a optimizar su infraestructura y gastos de Kubernetes.

¿Se cobra por esta funcionalidad?

No. Puede usar esta versión de Kubecost sin cargo adicional. Si desea capacidades adicionales de Kubecost que no están incluidas en este paquete, puede comprar una licencia empresarial de Kubecost a través de AWS Marketplace o directamente desde Kubecost.

¿Hay soporte disponible?

Sí. Puede abrir un caso de soporte con el equipo de AWS Support en Contacte con AWS.

¿Necesito una licencia para usar las características de Kubecost proporcionadas por la integración de Amazon EKS?

No.

¿Puedo integrar Kubecost con AWS Cost and Usage Report para obtener informes más precisos?

Sí. Puede configurar Kubecost para que capture datos de AWS Cost and Usage Report y así obtener una visibilidad precisa de los costos, incluidos descuentos, precios de spot, precios de instancias

reservadas y otros. Para más información, consulte Integración de facturación de AWS Cloud en la documentación de Kubecost.

¿Esta versión admite la administración de costos de los clústeres de Kubernetes autoadministrados en Amazon EC2?

No. Esta versión solo es compatible con los clústeres de Amazon EKS.

¿Kubecost puede hacer un seguimiento de los costos de Amazon EKS en AWS Fargate?

Kubecost ofrece el mejor esfuerzo para mostrar la visibilidad de los costos de los clústeres de Amazon EKS en Fargate, pero con una precisión inferior a la de Amazon EKS en Amazon EC2. Esto se debe principalmente a la diferencia en la forma en que se le factura el uso. Con Amazon EKS en Fargate, se le facturan los recursos consumidos. Con Amazon EKS en los nodos de Amazon EC2, se le facturan los recursos aprovisionados. Kubecost calcula el costo de un nodo de Amazon EC2 en función de la especificación del nodo, lo cual incluye la CPU, la RAM y el almacenamiento efímero. Con Fargate, los costos se calculan en función de los recursos solicitados para los Pods de Fargate.

¿Cómo puedo obtener actualizaciones y nuevas versiones de Kubecost?

Puede actualizar su versión de Kubecost mediante procedimientos de actualización estándar de Helm. Las versiones más recientes se encuentran en la Galería pública de Amazon ECR.

¿Es **`kubectl-cost`** compatible con la CLI? ¿Cómo se instala?

Sí. `Kubectl-cost` es una herramienta de código abierto de Kubecost (Licencia Apache 2.0) que proporciona acceso de CLI a las métricas de asignación de costos de Kubernetes. Para instalar `kubectl-cost`, consulte Installation (Instalación) en GitHub.

¿La interfaz de usuario de Kubecost es compatible? ¿Cómo puedo acceder a ella?

Kubecost proporciona un panel web al que puede acceder a través del reenvío de puertos `kubectl`, una entrada o un equilibrador de carga. También puede usar el AWS Load Balancer Controller para exponer Kubecost y usar Amazon Cognito para la autenticación, autorización y administración de usuarios. Para obtener más información, consulte Cómo usar el equilibrador de carga de aplicación y Amazon Cognito para autenticar a usuarios de las aplicaciones web de Kubernetes en el blog de AWS.

¿Amazon EKS Anywhere es compatible?

No.

Instalación del servidor de métricas de Kubernetes

El servidor de métricas de Kubernetes es un agregador de datos de uso de recursos en el clúster. No está implementado en los clústeres de Amazon EKS de forma predeterminada. Para obtener más información, consulte Servidor de métricas de Kubernetes en GitHub. El servidor de métricas suelen usarlo otros complementos de Kubernetes, como Escalador automático de pods horizontales o el panel de Kubernetes. Para obtener más información, consulte Canalización de métricas de recursos en la documentación de Kubernetes. En este tema, se explica cómo implementar el servidor de métricas de Kubernetes en el clúster de Amazon EKS.

> **Important**
>
> Las métricas están pensadas para el análisis en un momento dado y no son una fuente precisa para el análisis histórico. No se pueden utilizar como solución de monitorización ni para otros fines que no sean de escalado automático. Para obtener más información sobre las herramientas de monitorización, consulte Observabilidad en Amazon EKS.

Implementar el servidor de métricas

1. Implemente el servidor de métricas con el siguiente comando:

   ```
   kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
   ```

   Si utiliza Fargate, deberá cambiar este archivo. En la configuración predeterminada, el servidor de métricas usa el puerto 10250. Este puerto está reservado en Fargate. Sustituya las referencias al puerto 10250 en components.yaml por otro puerto, como el 10251.

2. Compruebe que la implementación de `metrics-server` está ejecutando la cantidad deseada de Pods con el siguiente comando:

   ```
   kubectl get deployment metrics-server -n kube-system
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   NAME             READY   UP-TO-DATE   AVAILABLE   AGE
   metrics-server   1/1     1            1           6m
   ```

Utilizar Helm con Amazon EKS

El administrador de paquetes Helm para Kubernetes lo ayuda a instalar y administrar aplicaciones en su clúster de Kubernetes. Para obtener más información, consulte la documentación de Helm. Este tema lo ayudará a instalar y ejecutar los archivos binarios de Helm para que pueda instalar y administrar gráficos mediante la CLI de Helm en su sistema local.

> **Important**
>
> Antes de poder instalar gráficos de Helm en el clúster de Amazon EKS, debe configurar `kubectl` para que funcione con Amazon EKS. Si todavía no ha hecho esto, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS antes de continuar. Si el siguiente comando se realiza correctamente para su clúster, entonces tiene la configuración correcta.
>
> ```
> kubectl get svc
> ```

Para instalar los archivos binarios de Helm en su sistema local

1. Ejecute el comando apropiado para el sistema operativo del cliente.
   - Si utiliza MacOS con Homebrew, instale los archivos binarios con el siguiente comando.

     ```
     brew install helm
     ```
   - Si utiliza Windows con Chocolatey, instale los archivos binarios con el siguiente comando.

     ```
     choco install kubernetes-helm
     ```
   - Si utiliza Linux, instale los archivos binarios con los siguientes comandos.

     ```
     curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 >
      get_helm.sh
     chmod 700 get_helm.sh
     ./get_helm.sh
     ```

> **Note**
>
> Si recibe un mensaje indicando que debe instalar `openssl` antes, puede instalarlo mediante el siguiente comando.
>
> ```
> sudo yum install openssl
> ```

2. Para recoger los nuevos archivos binarios en su PATH, cierre la ventana del terminal actual y abra una nueva.
3. Consulte la versión de Helm que instaló.

   ```
   helm version | cut -d + -f 1
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   v3.9.0
   ```

4. En este momento, puede ejecutar cualquier comando de Helm (como `helm install` *chart-name*) para instalar, modificar, eliminar o consultar gráficos de Helm en el clúster. Si es nuevo en Helm y no tiene un gráfico específico que pueda instalar, puede:

   - Experimentar mediante la instalación un gráfico de muestra. Consulte instalación de un gráfico de muestra en la guía de inicio rápido de Helm.
   - Cree un gráfico de ejemplo y envíelo a Amazon ECR. Para obtener más información, consulte Envío de un gráfico de Helm en la Guía del usuario de Amazon Elastic Container Registry.
   - Instale un gráfico de Amazon EKS desde el repositorio de GitHub eks-charts o desde ArtifactHub.

Etiquetado de los recursos de Amazon EKS

Puede usar etiquetas para ayudarlo a administrar sus recursos de Amazon EKS. En este tema se proporciona información general sobre la función de etiquetas y se muestra cómo puede crear etiquetas.

Temas

- Conceptos básicos de etiquetas

- Etiquetado de recursos
- Restricciones de las etiquetas
- Etiquetado de los recursos para facturación
- Uso de etiquetas mediante la consola
- Uso de etiquetas mediante la CLI, la API o eksctl

> **Note**
>
> Las etiquetas son un tipo de metadatos independiente de las etiquetas y anotaciones de Kubernetes. Para obtener más información sobre estos otros tipos de metadatos, consulte las secciones siguientes de la documentación de Kubernetes:
>
> - Etiquetas y selectores
> - Annotations

Conceptos básicos de etiquetas

Una etiqueta es una marca que se asigna a un recurso de AWS. Cada etiqueta consta de una clave y un valor opcional.

Con las etiquetas, puede categorizar sus recursos de AWS. Por ejemplo, puede clasificar los recursos en categorías por objetivo, propietario o entorno. Cuando tiene muchos recursos del mismo tipo, puede utilizar las etiquetas que asignó a un recurso específico para identificarlo rápidamente. Por ejemplo, puede definir un conjunto de etiquetas para los clústeres de Amazon EKS a fin de ayudar a realizar un seguimiento del propietario y del nivel de pila de cada clúster. Le recomendamos que diseñe un conjunto coherente de claves de etiqueta para cada tipo de recurso. Puede buscar y filtrar los recursos en función de las etiquetas que agregue.

Después de agregar una etiqueta, puede editar las claves y los valores de las etiquetas o eliminar etiquetas de un recurso en cualquier momento. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

Las etiquetas no tienen ningún significado semántico para Amazon EKS, por lo que se interpretan estrictamente como cadenas de caracteres. Puede establecer el valor de una etiqueta como una cadena vacía. Sin embargo, no se puede establecer el valor de una etiqueta como nulo. Si

agrega una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al anterior.

Si utiliza AWS Identity and Access Management (IAM), puede controlar qué usuarios de su cuenta de AWS tienen permiso para administrar etiquetas.

Etiquetado de recursos

Las siguientes etiquetas de soporte de recursos de Amazon EKS:

- clústeres
- grupos de nodos administrados
- Perfiles de Fargate

Puede etiquetar estos recursos con lo siguiente:

- Si utiliza la consola de Amazon EKS, puede aplicar etiquetas a recursos nuevos o existentes en cualquier momento. Para ello, puede utilizar la pestaña Tags (Etiquetas) en la página de recursos pertinente. Para obtener más información, consulte Uso de etiquetas mediante la consola.
- Si utiliza `eksctl`, puede aplicar etiquetas a los recursos cuando se crean mediante la opción `--tags`.
- Si utiliza la AWS CLI, la API de Amazon EKS o un SDK de AWS, puede aplicar etiquetas a los recursos nuevos mediante el parámetro `tags` en la acción de la API pertinente. Puede aplicar etiquetas a recursos existentes a través de la acción de la API `TagResource`. Para obtener más información, consulte TagResource.

Cuando se utilizan algunas acciones de creación de recursos, se pueden especificar también etiquetas para el recurso al mismo tiempo que se crea. Si las etiquetas no pueden aplicarse mientras se crea el recurso, este no podrá crearse. Este mecanismo garantiza que los recursos que se pretenden etiquetar se creen con las etiquetas que se especifican o no se creen en absoluto. Si se etiquetan los recursos al crearlos, no es necesario ejecutar scripts de etiquetado personalizados después de crear el recurso.

Las etiquetas no se propagan a otros recursos asociados al recurso que se crea. Por ejemplo, las etiquetas de perfil de Fargate no se propagan a otros recursos asociados al perfil de Fargate, como los Pods que están programados con él.

Restricciones de las etiquetas

Se aplican las siguientes restricciones a las etiquetas:

- Se puede asociar un máximo de 50 etiquetas a un recurso.
- Las claves de etiquetas no se pueden repetir para un recurso. Cada clave de etiqueta debe ser única y solo puede tener un valor.
- Las claves pueden tener hasta 128 caracteres en UTF-8.
- Los valores pueden tener hasta 256 caracteres en UTF-8.
- Si hay múltiples Servicios de AWS y los recursos usan su esquema de etiquetado, limite los tipos de caracteres que usa. Algunos servicios pueden tener restricciones en cuanto a los caracteres permitidos. En general, los caracteres permitidos son letras, números, espacios y los siguientes caracteres: **`+ - = . _ : / @`**.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No utilice `aws:`, `AWS:`, ni ninguna combinación de mayúsculas o minúsculas del mismo como prefijo para claves o valores. Estos están reservados solo para la utilización de AWS. Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas con este prefijo no cuentan para el límite de etiquetas por recurso.

Etiquetado de los recursos para facturación

Cuando aplica etiquetas a los clústeres de Amazon EKS, puede utilizarlas para la asignación de costos en sus Informes de costo y uso. Los datos de medición de sus Informes de costo y uso muestran el uso en todos sus clústeres de Amazon EKS. Para obtener más información, consulte Informe de costos y usos de AWS en la Guía del usuario de AWS Billing.

La etiqueta de asignación de costos generada por AWS, específicamente `aws:eks:cluster-name`, le permite desglosar los costos de las instancias de Amazon EC2 por clúster individual de Amazon EKS en el Explorador de costos. Sin embargo, esta etiqueta no captura los gastos del plano de control. La etiqueta se agrega automáticamente a las instancias de Amazon EC2 que participan en un clúster de Amazon EKS. Este comportamiento se produce independientemente de si las instancias se aprovisionan mediante grupos de nodos administrados de Amazon EKS, Karpenter o directamente con Amazon EC2. Esta etiqueta específica no cuenta para el límite de 50 etiquetas. Para usar la etiqueta, el propietario de la cuenta debe activarla en la consola de AWS Billing o mediante la API. Cuando el propietario de una cuenta de administración de AWS Organizations activa la etiqueta, esta también se activa para todas las cuentas miembro de la organización.

También puede organizar su información de facturación en función de los recursos que tienen los mismos valores de clave de etiqueta. Por ejemplo, puede etiquetar varios recursos con un nombre de aplicación específico y, luego, organizar su información de facturación. De esta manera, puede ver el costo total de la aplicación en distintos servicios. Para obtener más información acerca de la configuración de un informe de asignación de costos con etiquetas, consulte Informe de asignación de costos mensual en la Guía del usuario de AWS Billing.

> **Note**
>
> Si acaba de activar los informes, los datos del mes actual estarán disponibles para su visualización después de 24 horas.

El Explorador de costos es una herramienta de informes que está disponible como parte del nivel gratuito de AWS. Puede utilizar el Explorador de costos para ver los gráficos de sus recursos de Amazon EKS de los últimos 13 meses. También puede prever cuánto va a gastar en los próximos tres meses. Puede ver los patrones de lo que gasta en recursos de AWS a lo largo del tiempo. Por ejemplo, se puede utilizar para identificar aspectos que deben estudiarse más a fondo y observar tendencias que pueden ayudar a comprender los costos. También puede especificar intervalos de tiempo para los datos y ver los datos temporales por día o por mes.

Uso de etiquetas mediante la consola

Con la consola de Amazon EKS puede administrar las etiquetas asociadas a los clústeres nuevos o existentes y a grupos de nodos administrados.

Al seleccionar una página específica de recursos en la consola de Amazon EKS, se muestra una lista de esos recursos. Por ejemplo, si selecciona Clusters (Clústeres) en el panel de navegación izquierda, la consola muestra una lista de los clústeres de Amazon EKS. Al seleccionar un recurso de una de estas listas (por ejemplo, un clúster concreto) que admite etiquetas, puede ver y administrar sus etiquetas en la pestaña Tags (Etiquetas).

También puede utilizar Tag Editor (Editor de etiquetas) en laAWS Management Console, que proporciona una forma unificada de administrar las etiquetas. Para obtener más información, consulte Etiquetar recursos de AWS con el editor de etiquetas en la Guía del usuario del editor de etiquetas deAWS.

Adición de etiquetas a un recurso al crearlo

Puede agregar etiquetas a clústeres de Amazon EKS y grupos de nodos administrados y perfiles de Fargate al crearlos. Para obtener más información, consulte Creación de un clúster de Amazon EKS.

Adición y eliminación de etiquetas en un recurso

Puede agregar o eliminar las etiquetas asociadas a sus clústeres directamente desde la página del recurso.

Para agregar o eliminar una etiqueta en un recurso individual

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En la barra de navegación, seleccione la Región de AWS que utilizará.
3. En el panel de navegación izquierdo, elija Clusters (Clústeres).
4. Elija un clúster específico.
5. Elija la pestaña Etiquetas y, a continuación, elija Administrar etiquetas.
6. En la página Manage tags (Administrar etiquetas), agregue o elimine las etiquetas según sea necesario.
   - Para agregar una etiqueta, elija Add tag (Añadir etiqueta). Especifique la clave y el valor para cada etiqueta.
   - Para eliminar una etiqueta, seleccione Remove tag (Eliminar etiqueta).
7. Repita este proceso para cada etiqueta que desee agregar o eliminar.
8. Elija Update (Actualizar) para finalizar.

Uso de etiquetas mediante la CLI, la API o **`eksctl`**

Utilice los siguientes comandos de la AWS CLI o las operaciones de la API de Amazon EKS para agregar, actualizar, enumerar y eliminar las etiquetas de sus recursos. Solo puede utilizar `eksctl` para agregar etiquetas mientras se crean simultáneamente los nuevos recursos con un comando.

Compatibilidad con el etiquetado de recursos de Amazon EKS

Tarea	AWS CLI	AWS Tools for Windows PowerShell	Acción de la API
Agregar o sobrescribir una o varias etiquetas.	tag-resource	Add-EKSResourceTag	TagResource
Eliminar una o varias etiquetas.	untag-resource	Remove-EKSResourceTag	UntagResource

Los siguientes ejemplos muestran cómo agregar o quitar etiquetas a los recursos mediante la AWS CLI.

Ejemplo 1: Etiquetar un clúster existente

El siguiente comando etiqueta un clúster existente.

```
aws eks tag-resource --resource-arn resource_ARN --tags team=devs
```

Ejemplo 2: Quitar la etiqueta de un clúster existente

El siguiente comando elimina una etiqueta de un clúster existente.

```
aws eks untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Ejemplo 3: enumerar etiquetas de un recurso

El siguiente comando enumera las etiquetas que están asociadas a un recurso existente.

```
aws eks list-tags-for-resource --resource-arn resource_ARN
```

Cuando se utilizan algunas acciones de creación de recursos, se pueden especificar etiquetas al mismo tiempo que se crea el recurso. Las siguientes acciones permiten especificar una etiqueta al crear un recurso.

Tarea	AWS CLI	AWS Tools for Windows PowerShell	Acción de la API	**eksctl**
Crear un clúster	create-cluster	New-EKSCluster	CreateCluster	create cluster
Crear un grupo de nodos administrados*	create-nodegroup	New-EKSNodegroup	CreateNodegroup	create nodegroup
Crear un perfil de Fargate	create-fargate-profile	New-EKSFargateProfile	CreateFargateProfile.html	create fargateprofile

* Si desea etiquetar también las instancias de Amazon EC2 al crear un grupo de nodos administrados, cree el grupo de nodos administrados mediante una plantilla de lanzamiento. Para obtener más información, consulte Etiquetado de instancias de Amazon EC2. Si las instancias ya existen, puede etiquetarlas de forma manual. Para obtener más información, consulte Etiquetado de los recursos en la Guía del usuario de Amazon EC2.

Cuotas de servicio de Amazon EKS

Amazon EKS se ha integrado con Service Quotas, un servicio de AWS que le permite ver y administrar sus cuotas desde una ubicación central. Para obtener más información, consulte ¿Qué son las Service Quotas? en la Guía del usuario de Service Quotas. Con la integración de Service Quotas, puede buscar rápidamente el valor de sus cuotas de servicio de Amazon EKS y AWS Fargate con la AWS Management Console y la AWS CLI.

AWS Management Console

Para ver las cuotas de servicio de Amazon EKS y Fargate con la AWS Management Console

1. Abra la consola de Service Quotas en https://console.aws.amazon.com/servicequotas/.
2. En el panel de navegación izquierdo, elija Servicios de AWS.
3. En la lista Servicios de AWS, busque y seleccione Amazon Elastic Kubernetes Service (Amazon EKS) o AWS Fargate.

En la lista Service Quotas, puede ver el nombre de la cuota de servicio, el valor aplicado (si está disponible), la cuota predeterminada de AWS y si el valor de cuota es ajustable.

4. Para ver información adicional sobre una cuota de servicio, como, por ejemplo, la descripción, elija el nombre de cuota.
5. (Opcional) Para solicitar un aumento de cuota, seleccione la cuota que desea aumentar, seleccione Solicitar aumento de cuota, escriba o seleccione la información necesaria y seleccione Solicitar.

Para trabajar más con cuotas de servicio mediante la AWS Management Console, consulte la Guía del usuario de Service Quotas. Para solicitar un aumento de cuota, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas.

AWS CLI

Para ver las cuotas de servicio de Amazon EKS y Fargate con la AWS CLI

Ejecute el siguiente comando para ver las cuotas de Amazon EKS.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code eks \
    --output table
```

Ejecute el siguiente comando para ver las cuotas de Fargate.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code fargate \
    --output table
```

> **Note**
>
> La cuota devuelta es el número de tareas de Amazon ECS o Pods de Amazon EKS que se ejecutan simultáneamente en Fargate en esta cuenta en la Región de AWS actual.

Para trabajar más con las cuotas de servicio mediante AWS CLI, consulte `service-quotas` en la Referencia de comandos AWS CLI. Para solicitar un aumento de cuota, consulte el `request-service-quota-increase` comando en la Referencia de comandos de la AWS CLI.

Service Quotas

Nombre	Valor predeterminado	Ajusta	Descripción
Entradas de acceso por clúster	Cada región admitida: 3 000	No	Número máximo de entradas de acceso por clúster.
Clústeres	Cada región admitida: 100	Sí	El número máximo de clústeres de EKS en esta cuenta en la región actual.
Grupos de seguridad del plano de control por clúster	Cada región admitida: 4	No	El número máximo de grupos de seguridad del plano de control por clúster (se especifican al crear el clúster).
Suscripciones a EKS Anywhere Enterprise	Cada región admitida: 10	Sí	Número máximo de suscripciones EKS Anywhere Enterprise para esta cuenta en la región actual.
Perfiles de Fargate por clúster	Cada región admitida: 10	Sí	El número máximo de perfiles Fargate por clúster.
Pares de etiquetas por selector de perfil de Fargate	Cada región admitida: 5	Sí	El número máximo de pares de etiquetas por

Nombre	Valor predeterminado	Ajusta	Descripción
			selector de perfiles de Fargate.
Grupos de nodos administrados por clúster	Cada región admitida: 30	Sí	El número máximo de grupos de nodos administrados por clúster.
Nodos por grupo de nodos administrados	Cada región admitida: 450	Sí	El número máximo de nodos por grupo de nodos administrados.
Rangos de CIDR de acceso público al punto de conexión por clúster	Cada región admitida: 40	No	El número máximo de rangos de CIDR de acceso a puntos de conexión públicos por clúster (se especifican al crear o actualizar el clúster).
Clústeres registrados	Cada región admitida: 10	Sí	El número máximo de clústeres registrados de esta cuenta en la región actual.
Selectores por perfil de Fargate	Cada región admitida: 5	Sí	El número máximo de selectores por perfil de Fargate.

Note

Los valores predeterminados son las cuotas iniciales que ha establecido AWS. Estos valores predeterminados son independientes de los valores reales de la cuota aplicada y de las cuotas de servicio máximas posibles. Para obtener más información, consulte Terminología de Service Quotas en la Guía del usuario de Service Quotas.

Estas cuotas de servicio se enumeran en Amazon Elastic Kubernetes Service (Amazon EKS), en la consola de Service Quotas. Para solicitar un aumento de cuota para los valores que se muestran como ajustables, consulte Requesting a quota increase (Solicitud de aumento de cuota) en la Guía del usuario de Service Quotas.

Service Quotas de AWS Fargate

Este servicio de AWS Fargate en la consola de Service Quotas enumeran varias cuotas de servicio. En la siguiente tabla solo se describen las cuotas que se aplican a Amazon EKS. Puede configurar alarmas que le avisen cuando su uso se acerque a una Service Quota. Para obtener más información, consulte Creación de una alarma de CloudWatch para monitorear las métricas de uso de recursos de Fargate.

Las nuevas Cuentas de AWS pueden tener cuotas iniciales más bajas que pueden aumentar con el tiempo. Fargate supervisa constantemente el uso de la cuenta dentro de cada Región de AWS y luego aumenta automáticamente las cuotas en función de su uso. También puede solicitar un aumento de cuota para los valores que se muestran como ajustables. Para obtener más información, consulte este tema acerca de cómo solicitar un aumento de cuota en la Guía del usuario de Service Quotas.

Nombre	Valor predeterminado	Ajustable	Descripción
Recuento de recursos de vCPU de Fargate bajo demanda	6	Sí	El número de vCPU de Fargate que se pueden ejecutar simultáneamente como Fargate bajo demanda en esta cuenta en la región actual.

> **Note**
>
> Los valores predeterminados son las cuotas iniciales que ha establecido AWS. Estos valores predeterminados son independientes de los valores reales de la cuota aplicada y de las

cuotas de servicio máximas posibles. Para obtener más información, consulte Terminología de Service Quotas en la Guía del usuario de Service Quotas.

Note

Fargate aplica adicionalmente las tareas de Amazon ECS y las cuotas de la tasa de lanzamiento de Pods de Amazon EKS. Para obtener más información, consulte AWS Fargate throttling quotas en la Guía de Amazon ECS.

Seguridad en Amazon EKS

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos que se han diseñado para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y el usuario. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad en la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. En Amazon EKS, AWS es responsable del plano de control de Kubernetes, que incluye los nodos del plano de control y la base de datos `etcd`. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de conformidad que se aplican a Amazon EKS, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: las siguientes áreas son su responsabilidad.
  - La configuración de seguridad del plano de datos, incluida la configuración de los grupos de seguridad que permiten que el tráfico pase del plano de control de Amazon EKS a la VPC del cliente
  - La configuración de los nodos y los contenedores
  - El sistema operativo de los nodos (incluidas las actualizaciones y los parches de seguridad)
  - Otros software de aplicaciones asociado:
    - Configuración y administración de controles de red, como las reglas del firewall
    - Administración de identidad y acceso de nivel de plataforma, con o además de IAM
  - La confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables

Esta documentación lo ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon EKS. En los siguientes temas, se mostrará cómo configurar Amazon EKS para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de AWS que ayudan a monitorear y proteger los recursos de Amazon EKS.

> **Note**
>
> Los contenedores de Linux se componen de grupos de control (cgroups) y espacios de nombres que ayudan a limitar el acceso de un contenedor, pero todos los contenedores comparten el mismo kernel de Linux que la instancia de Amazon EC2 host. No se recomienda ejecutar un contenedor como usuario raíz (UID 0) o conceder acceso a un contenedor a recursos o espacios de nombres de anfitrión como la red de anfitrión o el espacio de nombres PID de anfitrión, ya que al hacerlo se reduce la eficacia del aislamiento que proporcionan los contenedores.

Temas

Firma de certificados

La API de certificados de Kubernetes automatiza el aprovisionamiento de credenciales X.509. La API cuenta con una interfaz de línea de comandos para que los clientes API de Kubernetes soliciten y obtengan Certificados X.509 de una entidad de certificación (CA). Se puede usar un recurso `CertificateSigningRequest` (CSR) para solicitar que un firmante indicado firme el certificado. Las solicitudes se aprueban o rechazan antes de firmarlas. Kubernetes admite tanto firmantes integrados como firmantes personalizados con comportamientos bien definidos. De esta forma, los clientes pueden predecir qué sucede con sus CSR. Para obtener más información sobre la firma de certificados, consulte acerca de la firma de solicitudes.

Uno de los firmantes integrados es `kubernetes.io/legacy-unknown`. La API `v1beta1` del recurso de CSR honró a este firmante desconocido de legado. Sin embargo, la API estable de CSR `v1` no permite que el `signerName` se establezca en `kubernetes.io/legacy-unknown`.

La versión `1.21` de Amazon EKS y las versiones anteriores permitían el valor `legacy-unknown` como el `signerName` en la API de CSR `v1beta1`. Esta API permite a la autoridad de certificación (CA) de Amazon EKS generar certificados. Sin embargo, en la versión `1.22` de Kubernetes, la API de CSR `v1beta1` se sustituyó por la API de CSR `v1`. Esta API no admite el signerName de "legado desconocido". Si desea utilizar la CA de Amazon EKS para generar certificados en sus clústers, debe usar un firmante personalizado. Se introdujo en la versión de Amazon EKS `1.22`. Para utilizar la versión de la API de CSR `v1` y generar un nuevo certificado, debe migrar cualquier manifiesto y cliente API existentes. Los certificados existentes creados con las API `v1beta1` anteriores son válidas y funcionan hasta que caduque el certificado. Esta incluye lo siguiente:

- Distribución de confianza: ninguna. No hay confianza o distribución estándar para este firmante en un clúster de Kubernetes.
- Temas permitidos: cualquiera
- Extensiones x509 permitidas: honra las extensiones de uso de subjectAltName y clave y descarta otras extensiones
- Usos de clave permitidos: no debe incluir usos más allá de ["cifrado de clave", "firma digital", "autenticación del servidor"]

> **Note**
>
> No se admite la firma de certificados de cliente.

- Vida útil del certificado/caducidad: 1 año (predeterminado y máximo)
- Bit CA permitido/no permitido: no permitido

Generación de CSR de ejemplo con signerName

En estos pasos se muestra cómo generar un certificado de publicación para el nombre de DNS `myserver.default.svc` con `signerName: beta.eks.amazonaws.com/app-serving`. Úselo como guía para su propio entorno.

1. Ejecute el comando `openssl genrsa -out myserver.key 2048` para generar una clave privada RSA.

```
openssl genrsa -out myserver.key 2048
```

2. Utilice el siguiente comando para generar una solicitud de certificado.

```
openssl req -new -key myserver.key -out myserver.csr -subj "/
CN=myserver.default.svc"
```

3. Genere un valor `base64` para la CSR y almacénelo en una variable para utilizarlo en un paso posterior.

```
base_64=$(cat myserver.csr | base64 -w 0 | tr -d "\n")
```

4. Ejecute el siguiente comando para crear un archivo llamado `mycsr.yaml`. En el siguiente ejemplo, `beta.eks.amazonaws.com/app-serving` es el `signerName`.

```
cat >mycsr.yaml <<EOF
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: myserver
spec:
  request: $base_64
  signerName: beta.eks.amazonaws.com/app-serving
  usages:
    - digital signature
    - key encipherment
    - server auth
EOF
```

5. Envíe la CSR.

```
kubectl apply -f mycsr.yaml
```

6. Apruebe el certificado de entrega.

```
kubectl certificate approve myserver
```

7. Compruebe que se emitió el certificado.

```
kubectl get csr myserver
```

Un ejemplo de salida sería el siguiente.

```
NAME       AGE     SIGNERNAME                            REQUESTOR
 CONDITION
myserver   3m20s   beta.eks.amazonaws.com/app-serving   kubernetes-admin
 Approved,Issued
```

8. Exporte el certificado emitido.

```
kubectl get csr myserver -o jsonpath='{.status.certificate}'| base64 -d
 > myserver.crt
```

Consideraciones sobre la firma de certificados antes de actualizar el clúster a la versión 1.24 de Kubernetes

En la versión `1.23` de Kubernetes y anteriores, los certificados de servicio de `kubelet` con nombres alternativos de asunto (SAN) de IP y DNS no verificables se emitían automáticamente con SAN no verificables. Los SAN se omiten del certificado aprovisionado. En `1.24` y clústeres posteriores, no se emiten certificados de servicio de `kubelet` si no se puede verificar un SAN. Esto impide que los comandos `kubectl exec` y `kubectl logs` funcionen.

Antes de actualizar el clúster a la versión `1.24`, complete los siguientes pasos para determinar si el clúster tiene solicitudes de firma de certificados (CSR) que no se hayan aprobado:

1. Ejecute el siguiente comando de la .

```
kubectl get csr -A
```

Un ejemplo de salida sería el siguiente.

```
NAME        AGE   SIGNERNAME                      REQUESTOR
                          REQUESTEDDURATION   CONDITION
csr-7znmf   90m   kubernetes.io/kubelet-serving
 system:node:ip-192-168-42-149.region.compute.internal      <none>
 Approved
csr-9xx5q   90m   kubernetes.io/kubelet-serving
 system:node:ip-192-168-65-38.region.compute.internal      <none>
 Approved, Issued
```

Si el resultado devuelto muestra una CSR con un firmante de `kubernetes.io/kubelet-serving` que está `Approved`, pero que no se ha `Issued` para un nodo, entonces deberá aprobar la solicitud.

2. Apruebe la CSR de forma manual. Reemplace `csr-7znmf` por su propio valor.

```
kubectl certificate approve csr-7znmf
```

Para aprobar automáticamente las solicitudes de firma de certificados en el futuro, recomendamos escribir un controlador de aprobación que pueda validar y aprobar automáticamente las CSR que contengan SAN IP o DNS que Amazon EKS no pueda verificar.

Administración de identidades y accesos para Amazon EKS

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de Amazon EKS. IAM es un Servicio de AWS que se puede utilizar sin cargo adicional.

Público

La forma en que utiliza AWS Identity and Access Management (IAM) difiere en función del trabajo que realiza en Amazon EKS.

Usuario de servicio: si utiliza el servicio Amazon EKS para realizar su trabajo, su administrador proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon EKS para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de Amazon EKS, consulte Solución de problemas de IAM.

Administrador de servicio: si está a cargo de los recursos de Amazon EKS de su empresa, probablemente tenga acceso completo a Amazon EKS. Su trabajo consiste en determinar a qué características y recursos de Amazon EKS deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo la empresa puede utilizar IAM con Amazon EKS, consulte Cómo funciona Amazon EKS con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información sobre cómo escribir políticas para administrar el acceso a Amazon EKS. Para consultar ejemplos de políticas de Amazon EKS basadas en identidades que puede utilizar en IAM, consulte Ejemplos de políticas de Amazon EKS basadas en identidades.

Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Debe estar autenticado (haber iniciado sesión en AWS) como Usuario raíz de la cuenta de AWS, como un usuario de IAM o asumiendo un rol de IAM.

Puede iniciar sesión en AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (del Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accede a AWS mediante la federación, está asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en AWS Management Console o en el portal de acceso AWS. Para obtener más información sobre el inicio de sesión en AWS, consulte Cómo iniciar sesión en su Cuenta de AWS en la Guía del usuario de AWS Sign-In.

Si accede a AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de la línea de comandos (CLI) para firmar criptográficamente las solicitudes mediante el uso de las credenciales. Si no usa las herramientas de AWS, debe encargarse de firmar las solicitudes. Para obtener más información sobre la firma de solicitudes, consulte Firma de solicitudes API de AWS en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, AWS recomienda el uso de la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Usuario raíz de Cuenta de AWS

Cuando se crea una Cuenta de AWS, se comienza con una identidad de inicio de sesión que tiene acceso completo a todos los recursos y Servicios de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la Cuenta de AWS y se accede a ella iniciando sesión con el email y

la contraseña que utilizó para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Usuarios y grupos de IAM

Un usuario de IAM es una identidad de la Cuenta de AWSque dispone de permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad de la Cuenta de AWSque dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en la AWS Management Console cambiando de roles. Puede asumir un rol llamando a una operación de la AWS CLI o de la API de AWS, o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS Single Sign-On.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. No obstante, con algunos Servicios de AWS se puede adjuntar una política directamente a un recurso (en lugar de utilizar un rol como representante). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan características de otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado a servicios.
  - Reenviar sesiones de acceso (FAS): cuando utiliza un rol o un usuario de IAM para llevar a cabo acciones en AWS, se le considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio

desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

- Rol vinculado a los servicios: un rol vinculado a servicios es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede utilizar un rol de IAM que le permita administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia de EC2 y realizan solicitudes a la AWS CLI o a la API de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar un rol de AWS a una instancia de EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia adjuntado a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se adjuntan a identidades o recursos de AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando una entidad principal (sesión de rol, usuario o usuario raíz) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. Las mayorías de las políticas se almacenan en AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con dicha política puede obtener información del usuario de la AWS Management Console, la AWS CLI o la API de AWS.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS. Las políticas administradas incluyen las políticas administradas por AWS y las políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No se puede utilizar políticas de IAM administradas por AWS en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3, AWS WAF y Amazon VPC son ejemplos de servicios que admiten las ACL. Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite otros tipos de políticas adicionales menos frecuentes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicio (SCP): las SCP son políticas de JSON que especifican los permisos máximos de una organización o una unidad organizativa en AWS Organizations. AWS Organizations es un servicio que le permite agrupar y administrar de manera centralizada varias Cuentas de AWS que posea su empresa. Si habilita todas las características en una empresa, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas. Una SCP limita los permisos para las entidades de las cuentas de miembros, incluido cada Usuario raíz de la cuenta de AWS. Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS decide si permitir o no una

solicitud cuando hay varios tipos de políticas implicados, consulte Lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon EKS con IAM

Antes de utilizar IAM para administrar el acceso a Amazon EKS, debe conocer qué características de IAM se encuentran disponibles con Amazon EKS. Para obtener una perspectiva general sobre cómo funcionan Amazon EKS y otros servicios de AWS con IAM, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Temas

Políticas de Amazon EKS basadas en identidades

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Amazon EKS admite acciones, claves de condiciones y recursos específicos. Para obtener más información acerca de los elementos que utiliza en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Acciones

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones de políticas de Amazon EKS utilizan el siguiente prefijo antes de la acción: `eks:`. Por ejemplo, para conceder permiso a alguien para conseguir información descriptiva sobre un clúster de

Amazon EKS, incluya la acción `DescribeCluster` en su política. Las instrucciones de la política deben incluir un elemento `Action` o un elemento `NotAction`.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": ["eks:action1", "eks:action2"]
```

Puede utilizar caracteres comodín para especificar varias acciones (*). Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "eks:Describe*"
```

Para ver una lista de las acciones de Amazon EKS, consulte Acciones definidas por Amazon Elastic Kubernetes Service en la Referencia de autorizaciones de servicio.

Recursos

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

El recurso del clúster de Amazon EKS tiene el siguiente ARN.

```
arn:aws:eks:region-code:account-id:cluster/cluster-name
```

Para obtener más información acerca del formato de los ARN, consulte Nombres de recursos de Amazon (ARN) y espacios de nombres de servicios de AWS.

Por ejemplo, para especificar el clúster llamado *my-cluster* en su instrucción, utilice el siguiente ARN:

```
"Resource": "arn:aws:eks:region-code:111122223333:cluster/my-cluster"
```

Para especificar todos los clústeres que pertenecen a una cuenta y Región de AWS específicas, utilice el carácter comodín (*):

```
"Resource": "arn:aws:eks:region-code:111122223333:cluster/*"
```

Algunas acciones de Amazon EKS, como las que se utilizan para crear recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

```
"Resource": "*"
```

Para ver una lista de los tipos de recursos de Amazon EKS y sus ARN, consulte Recursos definidos por Amazon Elastic Kubernetes Service en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon Elastic Kubernetes Service.

Claves de condición

Amazon EKS define su propio conjunto de claves de condición y también admite el uso de algunas claves de condición globales. Para ver todas las claves de condición globales de AWS, consulte Claves de contexto de condición globales de AWS en la Guía del usuario de IAM.

Puede establecer claves de condición al asociar un proveedor de OpenID Connect al clúster. Para obtener más información, consulte Política de IAM de ejemplo.

Todas las acciones de Amazon EC2 admiten las claves de condición `aws:RequestedRegion` y `ec2:Region`. Para obtener más información, consulte Ejemplo: restricción del acceso a una Región de AWS específica.

Para obtener una lista de las claves de condición de Amazon EKS, consulte Condiciones de Amazon Elastic Kubernetes Service en la Referencia de autorizaciones de servicio. Para obtener más información sobre las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Elastic Kubernetes Service.

Ejemplos

Para ver ejemplos de políticas de Amazon EKS basadas en identidades, consulte Ejemplos de políticas de Amazon EKS basadas en identidades.

Cuando se crea un clúster de Amazon EKS, la entidad principal de IAM que crea el clúster recibe permisos de `system:masters` de forma automática en la configuración del role-based access control (RBAC, control de acceso basado en roles) del clúster en el plano de control de Amazon EKS. Esta entidad principal no aparece en ninguna configuración visible, así que asegúrese de realizar un seguimiento de la entidad principal que creó el clúster originalmente. Para conceder a entidades principales de IAM la capacidad de interactuar con el clúster, edite el `ConfigMap` de `aws-auth` dentro de Kubernetes y cree un `rolebinding` de Kubernetes o `clusterrolebinding` con el nombre de un `group` que especifique en el `ConfigMap` de `aws-auth`.

Para obtener más información sobre cómo trabajar con el ConfigMap, consulte Concesión de acceso a las API de Kubernetes.

Políticas basadas en recursos de Amazon EKS

Amazon EKS no admite las políticas basadas en recursos.

Autorización basada en etiquetas de Amazon EKS

Puede asociar etiquetas a los recursos de Amazon EKS o transferirlas en una solicitud a Amazon EKS. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`. Para obtener más información sobre el etiquetado de recursos de Amazon EKS, consulte Etiquetado de los recursos de Amazon EKS. Para obtener más información sobre qué acciones puede usar las etiquetas en las claves de condición de, consulte Acciones definidas por Amazon EKS en la Referencia de autorizaciones de servicio.

Roles de IAM de Amazon EKS

Un rol de IAM es una entidad de la cuenta de AWS que dispone de permisos específicos.

Uso de credenciales temporales con Amazon EKS

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen mediante una llamada a operaciones de la API de AWS STS, como AssumeRole o GetFederationToken.

Amazon EKS admite el uso de credenciales temporales.

Roles vinculados al servicio

Los roles vinculados a servicios permiten a los servicios de AWS obtener acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Amazon EKS admite roles vinculados a servicios. Para obtener más información sobre cómo crear o administrar roles vinculados a servicios de Amazon EKS, consulte Utilizar roles vinculados a servicios para Amazon EKS.

Roles de servicio

Esta característica permite que un servicio asuma un rol de servicio en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

Amazon EKS admite roles de servicio. Para obtener más información, consulte Rol de IAM del clúster de Amazon EKS y Rol de IAM de nodo de Amazon EKS.

Elegir un rol de IAM en Amazon EKS

Cuando se crea un recurso de clúster en Amazon EKS, debe elegir un rol para permitir a Amazon EKS acceder a otros recursos de AWS en su nombre. Si ya ha creado una función del servicio, Amazon EKS proporciona una lista de roles para elegir. Es importante que elija un rol que cuente con políticas administradas de Amazon EKS asociadas a él. Para obtener más información, consulte Comprobar si existe un rol de clúster existente y Verificar un rol de nodo existente.

Ejemplos de políticas de Amazon EKS basadas en identidades

De forma predeterminada, los usuarios y roles de IAM no tienen permiso para crear ni modificar recursos de Amazon EKS. Tampoco pueden realizar tareas mediante la AWS Management Console, la AWS CLI, o la API de AWS. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas de JSON de ejemplo, consulte Creación de políticas en la pestaña JSON en la Guía del usuario de IAM.

Cuando se crea un clúster de Amazon EKS, la entidad principal de IAM que crea el clúster recibe permisos de `system:masters` de forma automática en la configuración del role-based access control (RBAC, control de acceso basado en roles) del clúster en el plano de control de Amazon EKS. Esta entidad principal no aparece en ninguna configuración visible, así que asegúrese de realizar un seguimiento de la entidad principal que creó el clúster originalmente. Para conceder a entidades principales de IAM la capacidad de interactuar con el clúster, edite el `ConfigMap` de `aws-auth` dentro de Kubernetes y cree un `rolebinding` de Kubernetes o `clusterrolebinding` con el nombre de un `group` que especifique en el `ConfigMap` de `aws-auth`.

Para obtener más información sobre cómo trabajar con el ConfigMap, consulte Concesión de acceso a las API de Kubernetes.

Temas

- Prácticas recomendadas sobre las políticas
- Utilizar la consola de Amazon EKS
- Permitir a los usuarios de IAM ver sus propios permisos
- Crear un clúster de Kubernetes en la Nube de AWS
- Crea un clúster local de Kubernetes en un Outpost
- Actualice un clúster de Kubernetes
- Enumeración o descripción de todos los clústeres

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon EKS de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas administradas por AWSy continúe con los permisos de privilegio mínimo: a fin de comenzar a conceder permisos a los usuarios y las cargas de trabajo, utilice las políticas administradas por AWS, que conceden permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Se recomienda definir políticas administradas por el

cliente de AWS específicas para sus casos de uso a fin de reducir aún más los permisos. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puede usar condiciones para conceder acceso a acciones de servicios si se emplean a través de un Servicio de AWS determinado, como por ejemplo AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Solicite la autenticación multifactor (MFA): si se encuentra en una situación en la que necesita usuarios raíz o de IAM en su Cuenta de AWS, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Utilizar la consola de Amazon EKS

Para acceder a la consola de Amazon EKS, una entidad principal de IAM debe tener un conjunto mínimo de permisos. Estos permisos permiten que la entidad principal enumere y vea detalles sobre los recursos de Amazon EKS en su cuenta de AWS. Si crea una política basada en identidad

que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades principales que tengan esa política adjunta.

Para asegurarse de que las entidades principales de IAM puedan seguir utilizando la consola de Amazon EKS, cree una política con su propio nombre, como `AmazonEKSAdminPolicy`. Adjunte la política a las entidades principales. Para obtener más información, consulte Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

> **Important**
>
> La siguiente política de ejemplo permite que una entidad principal vea información en la pestaña Configuración en la consola. Para ver información en las pestañas Resumen y Recursos en la AWS Management Console, la entidad principal también necesita permisos de Kubernetes. Para obtener más información, consulte Permisos necesarios.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "eks.amazonaws.com"
                }
            }
        }
    ]
}
```

No es necesario conceder permisos mínimos para la consola a las entidades principales que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Permitir a los usuarios de IAM ver sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para llevar a cabo esta acción en la consola o mediante programación con la AWS CLI o la API de AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

Crear un clúster de Kubernetes en la Nube de AWS

En esta política de ejemplo se incluyen los permisos mínimos necesarios para crear un clúster de Amazon EKS denominado *my-cluster* en la Región de AWS *us-west-2*. Puede reemplazar Región de AWS por la Región de AWS en la que desea implementar un clúster. Si ve una advertencia que diga Las acciones de su política no admiten permisos de nivel de recursos y requieren que elija **`All resources`** en la AWS Management Console, puede omitirla con toda tranquilidad. Si su cuenta ya tiene el rol *AWSServiceRoleForAmazonEKS*, puede quitar la acción `iam:CreateServiceLinkedRole` de la política. Si alguna vez creó un clúster de Amazon EKS en su cuenta, este rol ya existe, a menos que lo haya eliminado.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "eks:CreateCluster",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-cluster"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
eks.amazonaws.com/AWSServiceRoleForAmazonEKS",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iam:AWSServiceName": "eks"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/cluster-role-name"
        }
    ]
}
```

Crea un clúster local de Kubernetes en un Outpost

En esta política de ejemplo, se incluyen los permisos mínimos necesarios para crear un clúster local de Amazon EKS denominado *my-cluster* en un Outpost en la Región de AWS *us-west-2*. Puede reemplazar Región de AWS por la Región de AWS en la que desea implementar un clúster. Si ve una advertencia que diga Las acciones de su política no admiten permisos de nivel de recursos y requieren que elija **All resources** en la AWS Management Console, puede omitirla con toda tranquilidad. Si su cuenta ya tiene el rol `AWSServiceRoleForAmazonEKSLocalOutpost`, puede quitar la acción `iam:CreateServiceLinkedRole` de la política. Si alguna vez creó un clúster local de Amazon EKS en un Outpost en su cuenta, este rol ya existe, a menos que lo haya eliminado.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "eks:CreateCluster",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-cluster"
        },
        {
            "Action": [
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "iam:GetRole"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/outposts.eks-
local.amazonaws.com/AWSServiceRoleForAmazonEKSLocalOutpost"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole",
                "iam:ListAttachedRolePolicies"
            ]
            "Resource": "arn:aws:iam::111122223333:role/cluster-role-name"
        },
```

```
        {
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:TagInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile"
            ],
            "Resource": "arn:aws:iam::*:instance-profile/eks-local-*",
            "Effect": "Allow"
        },
    ]
}
```

Actualice un clúster de Kubernetes

Esta política de ejemplo incluye el permiso mínimo necesario para actualizar un clúster denominado *my-cluster* en la Región de AWS us-west-2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "eks:UpdateClusterVersion",
            "Resource": "arn:aws:eks:us-west-2:111122223333:cluster/my-cluster"
        }
    ]
}
```

Enumeración o descripción de todos los clústeres

En esta política de ejemplo se incluyen los permisos mínimos necesarios para enumerar y describir todos los clústeres de su cuenta. Una entidad principal de IAM tiene que ser capaz de enumerar y describir clústeres para usar el comando AWS CLI de `update-kubeconfig`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "eks:DescribeCluster",
                "eks:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

Utilizar roles vinculados a servicios para Amazon EKS

Amazon Elastic Kubernetes Service utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon EKS. Los roles vinculados a servicios se encuentran predefinidos por Amazon EKS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Temas

- Utilizar roles para clústeres de Amazon EKS
- Utilizar roles para grupos de nodos de Amazon EKS
- Utilizar roles para perfiles de Fargate de Amazon EKS
- Uso de roles para conectar un clúster de Kubernetes a Amazon EKS
- Uso de roles para clústeres locales de Amazon EKS en Outpost

Utilizar roles para clústeres de Amazon EKS

Amazon Elastic Kubernetes Service utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon EKS. Los roles vinculados a servicios se encuentran predefinidos por Amazon EKS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a servicios simplifica la configuración de Amazon EKS porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon EKS define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon EKS puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon EKS, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked role (Rol vinculado a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon EKS

Amazon EKS utiliza el rol vinculado a servicios denominado `AWSServiceRoleForAmazonEKS`: el rol permite a Amazon EKS administrar clústeres en su cuenta. Las políticas adjuntas permiten que el rol administre los siguientes recursos: interfaces de red, grupos de seguridad, registros y VPC.

> **Note**
>
> El rol vinculado al servicio `AWSServiceRoleForAmazonEKS` es distinto del rol requerido para la creación de clústeres. Para obtener más información, consulte Rol de IAM del clúster de Amazon EKS.

El rol vinculado al servicio `AWSServiceRoleForAmazonEKS` confía en los siguientes servicios para asumir el rol:

- `eks.amazonaws.com`

La política de permisos del rol permite que Amazon EKS realice las siguientes acciones en los recursos especificados:

- `AmazonEKSServiceRolePolicy`

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a servicios para Amazon EKS

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un clúster en la AWS Management Console, la AWS CLI, o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al crear un clúster, Amazon EKS se encarga de crear de nuevo el rol vinculado a servicios en su nombre.

Editar un rol vinculado a servicios para Amazon EKS

Amazon EKS no permite editar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKS`. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editing a service-linked role (Editar un rol vinculado a servicios) en la Guía del usuario de IAM.

Eliminar un rol vinculado a servicios para Amazon EKS

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si el servicio de Amazon EKS utiliza el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Amazon EKS que utiliza el rol de **`AWSServiceRoleForAmazonEKS`**.

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).

3. Si el clúster tiene grupos de nodos o perfiles de Fargate, debe eliminarlos para poder eliminar el clúster. Para obtener más información, consulte Eliminación de un grupo de nodos administrados y Eliminación de un perfil de Fargate.
4. En la página Clusters (Clústeres), elija el clúster que desea eliminar y elija Delete (Eliminar).
5. Escriba el nombre del clúster en la ventana de confirmación de eliminación y, a continuación, elija Delete (Eliminar).
6. Repita este procedimiento para el resto de los clústeres de la cuenta. Espere a que finalicen todas las operaciones de eliminación.

Elimine manualmente el rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKS`. Para obtener más información, consulte Eliminar un rol vinculado a un servicio en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Amazon EKS

Amazon EKS admite el uso de roles vinculados a servicios en todas las regiones en las que se encuentra disponible el servicio. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon EKS.

Utilizar roles para grupos de nodos de Amazon EKS

Amazon EKS utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon EKS. Los roles vinculados a servicios se encuentran predefinidos por Amazon EKS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a servicios simplifica la configuración de Amazon EKS porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon EKS define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon EKS puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon EKS, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked role (Rol vinculado a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon EKS

Amazon EKS utiliza el rol vinculado a servicios denominado `AWSServiceRoleForAmazonEKSNodegroup`: el rol permite a Amazon EKS administrar grupos de nodos en su cuenta. Las políticas adjuntas permiten que el rol administre los siguientes recursos: grupos de Auto Scaling, grupos de seguridad, plantillas de lanzamiento y perfiles de instancias de IAM.

El rol vinculado al servicio `AWSServiceRoleForAmazonEKSNodegroup` depende de los siguientes servicios para asumir el rol:

- `eks-nodegroup.amazonaws.com`

La política de permisos del rol permite que Amazon EKS realice las siguientes acciones en los recursos especificados:

- `AWSServiceRoleForAmazonEKSNodegroup`

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a servicios para Amazon EKS

No necesita crear manualmente un rol vinculado a servicios. Cuando ejecuta CreateNodegroup en la AWS Management Console, la AWS CLI, o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

> **Important**
>
> Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Si utilizaba el servicio de Amazon EKS antes del 1 de enero de 2017, fecha en que comenzó a admitir los roles vinculados a servicios, Amazon EKS creó el rol AWSServiceRoleForAmazonEKSNodegroup

en su cuenta. Para obtener más información, consulte Un nuevo rol ha aparecido en mi cuenta de IAM.

Crear un rol vinculado a servicios en Amazon EKS (API de AWS)

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un grupo de nodos administrados en la AWS Management Console, la AWS CLI o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea otro grupo de nodos administrado, Amazon EKS vuelve a crear el rol vinculado a servicios.

Editar un rol vinculado a servicios para Amazon EKS

Amazon EKS no permite editar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKSNodegroup`. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editing a service-linked role (Editar un rol vinculado a servicios) en la Guía del usuario de IAM.

Eliminar un rol vinculado a servicios para Amazon EKS

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si el servicio de Amazon EKS utiliza el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Amazon EKS que utiliza el rol de **`AWSServiceRoleForAmazonEKSNodegroup`**.

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. Seleccione la pestaña Compute (Informática).
4. En la sección de Node Groups (Grupos de nodos), elija el grupo de nodos que desea eliminar.
5. Escriba el nombre del grupo de nodos en la ventana de confirmación de eliminación y, a continuación, elija Delete (Eliminar).
6. Repita este procedimiento para los demás grupos de nodos del clúster. Espere a que finalicen todas las operaciones de eliminación.

Elimine manualmente el rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKSNodegroup`. Para obtener más información, consulte Eliminar un rol vinculado a un servicio en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Amazon EKS

Amazon EKS admite el uso de roles vinculados a servicios en todas las regiones en las que se encuentra disponible el servicio. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon EKS.

Utilizar roles para perfiles de Fargate de Amazon EKS

Amazon EKS utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon EKS. Los roles vinculados a servicios se encuentran predefinidos por Amazon EKS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a servicios simplifica la configuración de Amazon EKS porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon EKS define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon EKS puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon EKS, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked role (Rol vinculado a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon EKS

Amazon EKS usa el rol vinculado a servicios denominado `AWSServiceRoleForAmazonEKSForFargate`: el rol permite a Fargate de Amazon EKS configurar las redes de VPC necesarias para los Pods de Fargate. Las políticas asociadas permiten que el rol cree y elimine interfaces de red elásticas y describa los recursos y las interfaces de red elásticas.

El rol vinculado al servicio `AWSServiceRoleForAmazonEKSForFargate` depende de los siguientes servicios para asumir el rol:

- `eks-fargate.amazonaws.com`

La política de permisos del rol permite que Amazon EKS realice las siguientes acciones en los recursos especificados:

- `AmazonEKSForFargateServiceRolePolicy`

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a servicios para Amazon EKS

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un perfil de Fargate en la AWS Management Console, la AWS CLI, o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

> **Important**
>
> Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol.

Si utilizaba el servicio de Amazon EKS antes del 13 de diciembre de 2019, fecha en que comenzó a admitir los roles vinculados a servicios, Amazon EKS creó el rol AWSServiceRoleForAmazonEKSForFargate en su cuenta. Para obtener más información, consulte Un nuevo rol ha aparecido en mi cuenta de IAM.

Crear un rol vinculado a servicios en Amazon EKS (API de AWS)

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un perfil de Fargate en la AWS Management Console, la AWS CLI, o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea otro grupo de nodos administrado, Amazon EKS vuelve a crear el rol vinculado a servicios.

Editar un rol vinculado a servicios para Amazon EKS

Amazon EKS no permite editar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKSForFargate`. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editing a service-linked role (Editar un rol vinculado a servicios) en la Guía del usuario de IAM.

Eliminar un rol vinculado a servicios para Amazon EKS

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

Note

Si el servicio de Amazon EKS utiliza el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Amazon EKS que utiliza el rol de **`AWSServiceRoleForAmazonEKSForFargate`**.

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. En la página Clusters (Clústeres), seleccione el clúster.
4. Seleccione la pestaña Compute (Informática).
5. Si hay algún perfil de Fargate en la sección de Fargate Profiles (Perfiles de Fargate), seleccione cada uno de forma individual y, a continuación, elija Delete (Eliminar).
6. Escriba el nombre del perfil en la ventana de confirmación de eliminación y, a continuación, elija Delete (Eliminar).
7. Repita este procedimiento para cualquier otro perfil de Fargate del clúster y cualquier otro clúster de su cuenta.

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios AWSServiceRoleForAmazonEKSForFargate. Para obtener más información, consulte Eliminar un rol vinculado a un servicio en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Amazon EKS

Amazon EKS admite el uso de roles vinculados a servicios en todas las regiones en las que se encuentra disponible el servicio. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon EKS.

Uso de roles para conectar un clúster de Kubernetes a Amazon EKS

Amazon EKS utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon EKS. Los roles vinculados a servicios se encuentran predefinidos por Amazon EKS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a servicios simplifica la configuración de Amazon EKS porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon EKS define los permisos de sus roles

vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon EKS puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon EKS, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked role (Rol vinculado a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon EKS

Amazon EKS utiliza el rol vinculado a servicios denominado `AWSServiceRoleForAmazonEKSConnector`: el rol permite a Amazon EKS conectar clústeres de Kubernetes. Las políticas adjuntas permiten que el rol administre los recursos necesarios para conectarse al clúster de Kubernetes registrado.

El rol vinculado al servicio `AWSServiceRoleForAmazonEKSConnector` depende de los siguientes servicios para asumir el rol:

- `eks-connector.amazonaws.com`

La política de permisos del rol permite que Amazon EKS realice las siguientes acciones en los recursos especificados:

- `AmazonEKSConnectorServiceRolePolicy`

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a servicios para Amazon EKS

No necesita crear un rol vinculado a servicios de forma manual para conectar un clúster. Cuando conecta un clúster en la AWS Management Console, la AWS CLI, `eksctl` o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al conectar un clúster, Amazon EKS crea de nuevo el rol vinculado a servicios en su nombre.

Editar un rol vinculado a servicios para Amazon EKS

Amazon EKS no permite editar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKSConnector`. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editing a service-linked role (Editar un rol vinculado a servicios) en la Guía del usuario de IAM.

Eliminar un rol vinculado a servicios para Amazon EKS

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si el servicio de Amazon EKS utiliza el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Amazon EKS que utiliza el rol de **`AWSServiceRoleForAmazonEKSConnector`**.

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Clusters (Clústeres).
3. En la página Clusters (Clústeres), seleccione el clúster.
4. Seleccione la pestaña Deregister (Anular registro) y, a continuación, la pestaña OK (Aceptar).

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios AWSServiceRoleForAmazonEKSConnector. Para obtener más información, consulte Eliminación de un rol vinculado a un servicio en la Guía del usuario de IAM.

Uso de roles para clústeres locales de Amazon EKS en Outpost

Amazon Elastic Kubernetes Service utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que se encuentra vinculado directamente a Amazon EKS. Los roles vinculados a servicios se encuentran predefinidos por Amazon EKS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a servicios simplifica la configuración de Amazon EKS porque ya no tendrá que agregar de forma manual los permisos necesarios. Amazon EKS define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon EKS puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon EKS, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Yes (Sí) en la columna Service-linked role (Rol vinculado a servicios). Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon EKS

Amazon EKS usa el rol vinculado a servicios denominado `AWSServiceRoleForAmazonEKSLocalOutpost`: el rol permite a Amazon EKS administrar clústeres locales en su cuenta. Las políticas asociadas permiten que el rol administre los siguientes recursos: interfaces de red, grupos de seguridad, registros e instancias de Amazon EC2.

> **Note**
>
> El rol vinculado al servicio `AWSServiceRoleForAmazonEKSLocalOutpost` es distinto del rol requerido para la creación de clústeres. Para obtener más información, consulte Rol de IAM del clúster de Amazon EKS.

El rol vinculado al servicio `AWSServiceRoleForAmazonEKSLocalOutpost` confía en los siguientes servicios para asumir el rol:

- `outposts.eks-local.amazonaws.com`

La política de permisos del rol permite que Amazon EKS realice las siguientes acciones en los recursos especificados:

- `AmazonEKSServiceRolePolicy`

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a servicios para Amazon EKS

No necesita crear manualmente un rol vinculado a servicios. Cuando crea un clúster en la AWS Management Console, la AWS CLI, o la API de AWS, Amazon EKS crea el rol vinculado a servicios en su nombre.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al crear un clúster, Amazon EKS se encarga de crear de nuevo el rol vinculado a servicios en su nombre.

Editar un rol vinculado a servicios para Amazon EKS

Amazon EKS no permite editar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKSLocalOutpost`. Después de crear un rol vinculado a un servicio, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia a él. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editing a service-linked role (Editar un rol vinculado a servicios) en la Guía del usuario de IAM.

Eliminar un rol vinculado a servicios para Amazon EKS

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si el servicio de Amazon EKS utiliza el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Amazon EKS que utiliza el rol de **`AWSServiceRoleForAmazonEKSLocalOutpost`**.

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En el panel de navegación izquierdo, elija Amazon EKS Clusters (Clústeres de Amazon EKS).
3. Si el clúster tiene grupos de nodos o perfiles de Fargate, debe eliminarlos para poder eliminar el clúster. Para obtener más información, consulte Eliminación de un grupo de nodos administrados y Eliminación de un perfil de Fargate.
4. En la página Clusters (Clústeres), elija el clúster que desea eliminar y elija Delete (Eliminar).
5. Escriba el nombre del clúster en la ventana de confirmación de eliminación y, a continuación, elija Delete (Eliminar).
6. Repita este procedimiento para el resto de los clústeres de la cuenta. Espere a que finalicen todas las operaciones de eliminación.

Elimine manualmente el rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios de `AWSServiceRoleForAmazonEKSLocalOutpost`. Para obtener más información, consulte Eliminar un rol vinculado a un servicio en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Amazon EKS

Amazon EKS admite el uso de roles vinculados a servicios en todas las regiones en las que se encuentra disponible el servicio. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon EKS.

Rol de IAM del clúster de Amazon EKS

El rol de IAM del clúster de Amazon EKS es obligatorio para cada clúster. Los clústeres de Kubernetes administrados por Amazon EKS utilizan este rol para administrar los nodos, y el proveedor de nube heredado lo emplea para crear equilibradores de carga con Elastic Load Balancing para los servicios.

Para poder crear clústeres de Amazon EKS, debe crear un rol de IAM con una de las siguientes políticas de IAM:

- `AmazonEKSClusterPolicy`
- Una política de IAM personalizada. Los permisos mínimos que aparecen a continuación permiten que el clúster de Kubernetes administre los nodos, pero no que el proveedor de nube heredado cree equilibradores de carga con Elastic Load Balancing. La política de IAM personalizada debe disponer al menos de los siguientes permisos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "ForAnyValue:StringLike": {
          "aws:TagKeys": "kubernetes.io/cluster/*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
```

```
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeAvailabilityZones",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Antes del 3 de octubre de 2023, se requería AmazonEKSClusterPolicy en el rol de IAM de cada clúster.
> Antes del 16 de abril de 2020, se requería AmazonEKSServicePolicy y AmazonEKSClusterPolicy y el nombre sugerido era `eksServiceRole`. Con el rol vinculado a servicios de `AWSServiceRoleForAmazonEKS`, la política AmazonEKSServicePolicy ya no es necesaria para clústeres creados a partir del 16 de abril de 2020.

Comprobar si existe un rol de clúster existente

Puede utilizar el siguiente procedimiento para verificar y conocer si la cuenta ya dispone del rol de clúster de Amazon EKS.

Para verificar el **`eksClusterRole`** en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la lista de roles, busque `eksClusterRole`. Si no existe un rol que incluya `eksClusterRole`, consulte Crear el rol de clúster de Amazon EKS para crearlo. Si existe un rol que incluye `eksClusterRole`, seleccione el rol para ver las políticas asociadas.
4. Elija Permissions.
5. Asegúrese de que la política administrada AmazonEKSClusterPolicyse ha asociado al rol.
   Si la política se ha adjuntado, entonces el rol de clúster de Amazon EKS se ha configurado correctamente.

6. Elija Trust relationships (Relaciones de confianza) y, a continuación, Edit trust policy (Editar política de confianza).
7. Verifique que la relación de confianza contiene la siguiente política. Si la relación de confianza coincide con la política a continuación, seleccione Cancelar. Si la relación de confianza no coincide, copie la política en la ventana Editar política de confianza y elija Actualizar política.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Crear el rol de clúster de Amazon EKS

Para crear el rol de clúster, puede utilizar la AWS Management Console o AWS CLI.

AWS Management Console

Para crear el rol de clúster de Amazon EKS en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Elija Roles y, a continuación, Create role (Crear rol).
3. En Tipo de entidad de confianza, seleccione Servicio de AWS.
4. En la lista desplegable Casos de uso para otros Servicios de AWS, elija EKS.
5. Elija EKS: clúster para su caso de uso y, luego, elija Siguiente.
6. En la pestaña Agregar permisos, seleccione Siguiente.
7. En Nombre del rol, ingrese un nombre único para su rol, por ejemplo, **`eksClusterRole`**.
8. En Description (Descripción), ingrese texto descriptivo, como **`Amazon EKS - Cluster role`**.
9. Elija Crear rol.

AWS CLI

1. Copie el siguiente contenido en un archivo denominado *cluster-trust-policy.json*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Cree el rol. Puede reemplazar ***eksClusterRole*** con el nombre que usted elija.

```
aws iam create-role \
  --role-name eksClusterRole \
  --assume-role-policy-document file://"cluster-trust-policy.json"
```

3. Adjunte la política de IAM necesaria al rol de IAM

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
  --role-name eksClusterRole
```

Rol de IAM de nodo de Amazon EKS

El daemon de `kubelet` del nodo de Amazon EKS realiza llamadas a las API de AWS en su nombre. Los nodos reciben permisos de dichas llamadas de API a través de políticas asociadas y de un perfil de instancias de IAM. Antes de poder lanzar nodos y registrarlos en un clúster, debe crear un rol de IAM para dichos nodos, para utilizarlo cuando se lancen. Este requisito se aplica a nodos lanzados con la AMI optimizada para Amazon EKS proporcionada por Amazon o con cualquier otra AMI de nodo que pretenda utilizar. Además, este requisito se aplica tanto a los grupos de nodos administrados como a los administrados por el usuario.

> **Note**
>
> No se puede usar el mismo rol que se usa para crear clústeres.

Antes de crear un nodo, debe crear un rol de IAM con los siguientes permisos:

- Permisos para que el `kubelet` pueda describir los recursos de Amazon EC2 en la VPC, tal y como los proporciona la política `AmazonEKSWorkerNodePolicy`. Esta política también proporciona los permisos para el agente de Pod Identity de Amazon EKS.
- Permisos para que el `kubelet` pueda utilizar imágenes de contenedor de Amazon Elastic Container Registry (Amazon ECR), según lo dispuesto en la política `AmazonEC2ContainerRegistryReadOnly`. Los permisos para utilizar imágenes de contenedor de Amazon Elastic Container Registry (Amazon ECR) son necesarios porque los complementos integrados para redes ejecutan pods que utilizan imágenes de contenedor de Amazon ECR.
- (Opcional) Permisos para que el agente de Pod Identity de Amazon EKS utilice la acción `eks-auth:AssumeRoleForPodIdentity` para recuperar las credenciales de los pods. Si no usa AmazonEKSWorkerNodePolicy, debe proporcionar este permiso además de los permisos de EC2 para utilizar Pod Identity de EKS.
- (Opcional) Si no utiliza Pod Identity de EKS e IRSA para otorgar permisos a los pods de CNI de VPC, debe proporcionar permisos para la CNI de la VPC en el rol de instancia. Puede utilizar la política administrada `AmazonEKS_CNI_Policy` (si creó el clúster con la familia `IPv4`) o una política de IPv6 que usted cree (si creó el clúster con la familia `IPv6`). Sin embargo, en lugar de adjuntar la política a este rol, le recomendamos adjuntar la política a un rol independiente utilizado específicamente para el complemento Amazon VPC CNI. Para obtener más información acerca de cómo crear un rol independiente para el complemento Amazon VPC CNI, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

> **Note**
>
> Antes del 3 de octubre de 2023, se requerían `AmazonEKSWorkerNodePolicy` y `AmazonEC2ContainerRegistryReadOnly` en el rol de IAM de cada grupo de nodos administrados.

Los grupos de nodos de Amazon EC2 deben tener un rol de IAM diferente al perfil de Fargate. Para obtener más información, consulte Rol de IAM de ejecución de Pod de Amazon EKS.

Verificar un rol de nodo existente

Puede utilizar el siguiente procedimiento para verificar y conocer si la cuenta ya dispone del rol de nodo de Amazon EKS.

Para verificar el **`eksNodeRole`** en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la lista de roles, busque `eksNodeRole`, `AmazonEKSNodeRole` o `NodeInstanceRole`. Si no existe un rol con alguno de esos nombres, consulte Crear el rol de IAM de nodo de Amazon EKS para crear el rol. Si existe un rol que contiene `eksNodeRole`, `AmazonEKSNodeRole` o `NodeInstanceRole`, seleccione el rol para ver las políticas adjuntas.
4. Elija Permisos.
5. Asegúrese de que las políticas administradas AmazonEKSWorkerNodePolicy and AmazonEC2ContainerRegistryReadOnly estén asociadas al rol, o que haya una política personalizada asociada con los permisos mínimos.

   > **Note**
   >
   > Si la política AmazonEKS_CNI_Policy se encuentra adjunta al rol, se recomienda eliminarla y adjuntarla a un rol de IAM asignado a la cuenta de servicio de `aws-node` de Kubernetes en su lugar. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

6. Elija Relaciones de confianza y, a continuación, Editar política de confianza.
7. Verifique que la relación de confianza contiene la siguiente política. Si la relación de confianza coincide con la política a continuación, seleccione Cancelar. Si la relación de confianza no coincide, copie la política en la ventana Editar política de confianza y elija Actualizar política.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Crear el rol de IAM de nodo de Amazon EKS

Puede crear el rol de IAM del nodo con la AWS Management Console o la AWS CLI.

AWS Management Console

Para crear el rol de nodo de Amazon EKS en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la página Roles, elija Crear rol.
4. En la página Seleccionar entidad de confianza, haga lo siguiente:
   a. En la sección Tipo de entidad de confianza, elija Servicio de AWS.
   b. En Caso de uso, elija EC2.
   c. Elija Siguiente.
5. En la página Agregar permisos, asocie una política personalizada o haga lo siguiente:
   a. En el cuadro Filtrar políticas, escriba **`AmazonEKSWorkerNodePolicy`**.
   b. A continuación, marque la casilla situada a la izquierda de AmazonEKSWorkerNodePolicy en los resultados de la búsqueda.
   c. Elija Borrar filtros.
   d. En el cuadro Filtrar políticas, escriba **`AmazonEC2ContainerRegistryReadOnly`**.
   e. Marque la casilla situada a la izquierda de AmazonEC2ContainerRegistryReadOnly en los resultados de la búsqueda.

La política administrada AmazonEKS_CNI_Policy o una política de IPv6 que usted cree también se tiene que adjuntar a este rol o a un rol diferente asignado a la cuenta de servicio de Kubernetes de `aws-node`. Se recomienda asignar la política al rol asociado a la cuenta de servicio de Kubernetes en lugar de asignarla a este rol. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

f. Elija Siguiente.

6. En la página Nombrar, revisar y crear, haga lo siguiente:

   a. En Nombre del rol, ingrese un nombre único para su rol, por ejemplo, **`AmazonEKSNodeRole`**.

   b. En Descripción, sustituya el texto actual por un texto descriptivo, como **`Amazon EKS - Node role`**.

   c. En Agregar etiquetas (Opcional), de manera opcional, agregue metadatos al rol asociando etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetado de recursos de IAM en la Guía de usuario de IAM.

   d. Seleccione Crear rol.

AWS CLI

1. Ejecute el siguiente comando para crear un archivo `node-role-trust-relationship.json`.

```
cat >node-role-trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
EOF
```

2. Cree el rol de IAM.

```
aws iam create-role \
  --role-name AmazonEKSNodeRole \
  --assume-role-policy-document file://"node-role-trust-relationship.json"
```

3. Adjunte al rol de IAM las dos políticas administradas por IAM necesarias.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name AmazonEKSNodeRole
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name AmazonEKSNodeRole
```

4. Adjunte una de las siguientes políticas de IAM al rol de IAM en función de la familia de IP con la que haya creado el clúster. La política debe adjuntarse a este rol o a un rol asociado a la cuenta de servicio de Kubernetes de `aws-node` que se utiliza para el Amazon VPC CNI plugin for Kubernetes. Se recomienda asignar la política al rol asociado a la cuenta de servicio de Kubernetes. Para asignar la política al rol asociado a la cuenta de servicio de Kubernetes, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

   - IPv4

     ```
     aws iam attach-role-policy \
       --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \
       --role-name AmazonEKSNodeRole
     ```

   - IPv6

     1. Copie el siguiente texto y guárdelo en un archivo llamado *vpc-cni-ipv6-policy*.json.

        ```
        {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": [
        ```

```
                "ec2:AssignIpv6Addresses",
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstanceTypes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ]
        }
    ]
}
```

2. Cree la política de IAM.

```
aws iam create-policy --policy-name AmazonEKS_CNI_IPv6_Policy --policy-
document file://vpc-cni-ipv6-policy.json
```

3. Adjunte la política de IAM al rol de IAM. Reemplace *111122223333* por su ID de cuenta.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::111122223333:policy/AmazonEKS_CNI_IPv6_Policy \
  --role-name AmazonEKSNodeRole
```

Rol de IAM de ejecución de Pod de Amazon EKS

Se requiere el rol de ejecución de Pod de Amazon EKS para ejecutar Pods en la infraestructura de AWS Fargate.

Cuando su clúster crea Pods en infraestructura de AWS Fargate, los componentes que se ejecutan en la infraestructura de Fargate deben hacer llamadas a las API de AWS en su nombre. Es así para que puedan realizar acciones, como extraer imágenes de contenedores de Amazon ECR o enrutar

registros a otros servicios de AWS. El rol de ejecución de Pod de Amazon EKS proporciona los permisos de IAM para esta tarea.

Al crear un perfil de Fargate, debe especificar un rol de ejecución de Pod para los componentes de Amazon EKS que se ejecutan en la infraestructura de Fargate con el perfil. Este rol se agrega al control de acceso basado en roles (RBAC) de Kubernetes del clúster para su autorización. Esto permite al `kubelet` que se está ejecutando en la infraestructura de Fargate registrarse en el clúster de Amazon EKS para que pueda aparecer en el clúster como un nodo.

> **Note**
>
> El perfil de Fargate debe tener un rol de IAM diferente a los grupos de nodos de Amazon EC2.

> **Important**
>
> Los contenedores que se ejecutan en el Pod de Fargate no pueden asumir los permisos de IAM asociados a un rol de ejecución de Pod. Para brindar permisos a los contenedores de su Pod de Fargate para acceder a otros servicios de AWS, debe utilizar Roles de IAM para cuentas de servicio.

Antes de crear un perfil de Fargate, debe crear un rol de IAM con la `AmazonEKSFargatePodExecutionRolePolicy`.

Compruebe si hay un rol de ejecución de Pod existente correctamente

Puede utilizar el siguiente procedimiento para verificar y ver si su cuenta ya dispone del rol de ejecución de Pod de Amazon EKS correctamente configurado. Para evitar un problema de seguridad adjunto confuso, es importante que la función restrinja el acceso basándose en `SourceArn`. Puede modificar el rol de ejecución según sea necesario para incluir compatibilidad con perfiles Fargate en otros clústeres.

Para verificar si hay un rol de ejecución de Pod de Amazon EKS en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.

3. En la página Roles, busque la lista de roles para AmazonEKSFargatePodExecutionRole. Si el rol no existe, consulte Crear el rol de ejecución de Pod de Amazon EKS para crear el rol. Si el rol existe, elija el rol.
4. En la página AmazonEKSFargatePodExecutionRole, haga lo siguiente:
   a. Elija Permissions.
   b. Asegúrese de que la política AmazonEKSFargatePodExecutionRolePolicy administrada por Amazon esté asociada al rol.
   c. Seleccione Trust Relationships.
   d. Elija Edit trust policy (Editar la política de confianza).
5. En la página Editar la política de confianza, verifique que la relación de confianza contiene la siguiente política y que tenga una línea para los perfiles de Fargate en su clúster. Si es así, elija Cancel (Cancelar).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

   Si la política coincide pero no tiene una línea que especifique los perfiles de Fargate en su clúster, puede agregar la siguiente línea en la parte superior del objeto `ArnLike`. Reemplace *`region-code`* por el Región de AWS en que está su clúster, *`111122223333`* por el ID de su cuenta y *`my-cluster`* por el nombre de su clúster.

```
"aws:SourceArn": "arn:aws:eks:region-code:111122223333:fargateprofile/my-cluster/
*",
```

Si la política no coincide, copie la política anterior completa en el formulario y elija Update policy (Política de actualización). Reemplace *region-code* por la Región de AWS en la que se encuentra el clúster. Si desea utilizar el mismo rol en todas las Regiones de AWS de su cuenta, reemplace *region-code* por *. Reemplace *111122223333* por el nombre del clúster y *my-cluster* por el ID de la cuenta. Si quiere utilizar el mismo rol para todos los clústeres de su cuenta, reemplace *my-cluster* con *.

Crear el rol de ejecución de Pod de Amazon EKS

Si aún no tiene la función de ejecución de Pod de Amazon EKS para su clúster, puede utilizar la AWS Management Console o la AWS CLI para crearlo.

AWS Management Console

Para crear un rol de ejecución de Pod de AWS Fargate con la AWS Management Console

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la página Roles, elija Crear rol.
4. En la página Seleccionar entidad de confianza, haga lo siguiente:
    a. En la sección Tipo de entidad de confianza, elija Servicio de AWS.
    b. En la lista desplegable Casos de uso para otros Servicios de AWS, elija EKS.
    c. Elija EKS - Fargate Pod.
    d. Elija Siguiente.
5. Elija Next (Siguiente) en la página Add permissions (Agregar permisos).
6. En la página Name, review, and create (Nombre, revisar y crear), haga lo siguiente:
    a. En Nombre del rol, ingrese un nombre único para su rol, por ejemplo, **AmazonEKSFargatePodExecutionRole**.
    b. En Agregar etiquetas (Opcional), de manera opcional, agregue metadatos al rol asociando etiquetas como pares de clave-valor. Para obtener más información sobre el

uso de etiquetas en IAM, consulte Etiquetado de recursos de IAM en la Guía de usuario de IAM.

c. Seleccione Crear rol.

7. En la página Roles, busque la lista de roles para AmazonEKSFargatePodExecutionRole. Elija el rol .

8. En la página AmazonEKSFargatePodExecutionRole, haga lo siguiente:

a. Seleccione Trust Relationships.

b. Elija Edit trust policy (Editar la política de confianza).

9. En la página Edit trust policy (Editar política de confianza), lleve a cabo las siguientes operaciones:

a. Copie y pegue los siguientes contenidos en el formulario Edit trust policy (Editar política de confianza). Reemplace *region-code* con la Región de AWS en la que está su clúster. Si desea utilizar el mismo rol en todas las Regiones de AWS de su cuenta, reemplace *region-code* por *. Reemplace *111122223333* por el nombre del clúster y *my-cluster* por el ID de la cuenta. Si quiere utilizar el mismo rol para todos los clústeres de su cuenta, reemplace *my-cluster* con *.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Elija Actualizar política.

AWS CLI

Para crear un rol de ejecución de Pod de AWS Fargate con la AWS CLI

1. Copie y pegue los siguientes contenidos en un archivo denominado *pod-execution-role-trust-policy*.json. Reemplace *region-code* con la Región de AWS en la que está su clúster. Si desea utilizar el mismo rol en todas las Regiones de AWS de su cuenta, reemplace *region-code* por *. Reemplace *111122223333* por el nombre del clúster y *my-cluster* por el ID de la cuenta. Si quiere utilizar el mismo rol para todos los clústeres de su cuenta, reemplace *my-cluster* con *.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Condition": {
         "ArnLike": {
            "aws:SourceArn": "arn:aws:eks:region-
code:111122223333:fargateprofile/my-cluster/*"
         }
      },
      "Principal": {
        "Service": "eks-fargate-pods.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Cree un rol de IAM de ejecución de Pod.

```
aws iam create-role \
  --role-name AmazonEKSFargatePodExecutionRole \
  --assume-role-policy-document file://"pod-execution-role-trust-policy.json"
```

3. Adjunte la política administrada de IAM por Amazon EKS requerida al rol.

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSFargatePodExecutionRolePolicy \
  --role-name AmazonEKSFargatePodExecutionRole
```

Rol de IAM conector de Amazon EKS

Puede conectar clústeres de Kubernetes para verlos en la AWS Management Console. Para conectarse a un clúster de Kubernetes, cree un rol de IAM.

Verificar si hay un rol de EKS Conector existente

Puede utilizar el siguiente procedimiento para verificar y ver si la cuenta ya dispone del rol conector de Amazon EKS.

Para verificar el **`AmazonEKSConnectorAgentRole`** en la consola de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, seleccione Roles.
3. En la lista de roles, busque `AmazonEKSConnectorAgentRole`. Si no existe un rol que incluya `AmazonEKSConnectorAgentRole`, consulte Creación del rol de agente conector de Amazon EKS para crearlo. Si existe un rol que incluye `AmazonEKSConnectorAgentRole`, seleccione el rol para ver las políticas asociadas.
4. Elija Permissions.
5. Asegúrese de que la política administrada AmazonEKSConnectorAgentPolicy se haya adjuntado al rol. Si la política se ha adjuntado, entonces el rol de Amazon EKS Connector se ha configurado correctamente.
6. Elija Relaciones de confianza y, a continuación, Editar política de confianza.
7. Verifique que la relación de confianza contiene la siguiente política. Si la relación de confianza coincide con la política a continuación, seleccione Cancelar. Si la relación de confianza no coincide, copie la política en la ventana Editar política de confianza y elija Actualizar política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
```

```
    ]
}
```

Creación del rol de agente conector de Amazon EKS

Puede utilizar la AWS Management Console o AWS CloudFormation para crear un rol de agente conector.

AWS CLI

1. Cree un archivo con el nombre `eks-connector-agent-trust-policy.json`, que contenga el siguiente JSON que se va a utilizar para el rol de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Cree un archivo con el nombre `eks-connector-agent-policy.json` que contenga el siguiente JSON que se va a utilizar para el rol de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SsmControlChannel",
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel"
            ],
            "Resource": "arn:aws:eks:*:*:cluster/*"
```

```
        },
        {
            "Sid": "ssmDataplaneOperations",
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenDataChannel",
                "ssmmessages:OpenControlChannel"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Cree el rol de agente de Amazon EKS Connector con la política de confianza y la política que creó en la lista de elementos anterior.

```
aws iam create-role \
     --role-name AmazonEKSConnectorAgentRole \
     --assume-role-policy-document file://eks-connector-agent-trust-policy.json
```

4. Adjunte la política a su rol de agente de Amazon EKS Connector.

```
aws iam put-role-policy \
     --role-name AmazonEKSConnectorAgentRole \
     --policy-name AmazonEKSConnectorAgentPolicy \
     --policy-document file://eks-connector-agent-policy.json
```

AWS CloudFormation

Para crear el rol de agente conector de Amazon EKS con AWS CloudFormation.

1. Guarde la siguiente plantilla de AWS CloudFormation en un archivo de texto en su sistema local.

> **Note**
>
> Esta plantilla también crea el rol vinculado al servicio que de otro modo se crearía cuando se llama a la API `registerCluster`. Para obtener más información, consulte Uso de roles para conectar un clúster de Kubernetes a Amazon EKS.

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Provisions necessary resources needed to register clusters in EKS'
Parameters: {}
Resources:
  EKSConnectorSLR:
    Type: AWS::IAM::ServiceLinkedRole
    Properties:
      AWSServiceName: eks-connector.amazonaws.com

  EKSConnectorAgentRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action: [ 'sts:AssumeRole' ]
            Principal:
              Service: 'ssm.amazonaws.com'

  EKSConnectorAgentPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: EKSConnectorAgentPolicy
      Roles:
        - {Ref: 'EKSConnectorAgentRole'}
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: 'Allow'
            Action: [ 'ssmmessages:CreateControlChannel' ]
            Resource:
            - Fn::Sub: 'arn:${AWS::Partition}:eks:*:*:cluster/*'
          - Effect: 'Allow'
            Action: [ 'ssmmessages:CreateDataChannel',
 'ssmmessages:OpenDataChannel', 'ssmmessages:OpenControlChannel' ]
            Resource: "*"
Outputs:
  EKSConnectorAgentRoleArn:
    Description: The agent role that EKS connector uses to communicate with
 Servicios de AWS.
```

```
    Value: !GetAtt EKSConnectorAgentRole.Arn
```

2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. Elija Create stack (Crear pila) (ya sea con recursos nuevos o existentes).
4. Para Specify template (Especificar plantilla), seleccione Upload a template file (Actualizar un archivo de plantilla) y, a continuación, elija Choose file (Elegir archivo).
5. Elija el archivo que creó anteriormente y, a continuación, elija Next (Siguiente).
6. En Stack name (Nombre de pila), escriba un nombre para el rol, por ejemplo `eksConnectorAgentRole` y, a continuación, elija Next (Siguiente).
7. En la página Configurar opciones de pila, elija Siguiente.
8. En la página Review (Revisar), revise la información, confirme que la pila puede crear recursos de IAM y elija Create stack (Crear pila).

Políticas administradas por AWS para Amazon Elastic Kubernetes Service

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas por AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas de AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

Política administrada por AWS: AmazonEKS_CNI_Policy

No puede adjuntar la `AmazonEKS_CNI_Policy` a sus entidades de IAM. Antes de crear un grupo de nodos de Amazon EC2, esta política debe estar asociada al rol de IAM de nodo, o a un rol de IAM utilizado de forma específica por el Amazon VPC CNI plugin for Kubernetes. Esto es para que pueda realizar acciones en su nombre. Recomendamos que adjunte la política a un rol que solo utilice el complemento. Para obtener más información, consulte Trabajar con el complemento Amazon VPC CNI plugin for Kubernetes de Amazon EKS y Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

Detalles sobre los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas:

- **`ec2:*NetworkInterface`** y **`ec2:*PrivateIpAddresses`**: permite que el complemento CNI de Amazon VPC realice acciones como el aprovisionamiento de interfaces de red elásticas y direcciones IP de los Pods a fin de proporcionar redes para aplicaciones que se ejecutan en Amazon EKS.
- acciones de lectura **`ec2`**: permite que el complemento CNI de Amazon VPC realice acciones como describir instancias y subredes para ver la cantidad de direcciones IP libres en las subredes de Amazon VPC. La CNI de la VPC puede usar las direcciones IP libres de cada subred para seleccionar las subredes con la mayor cantidad de direcciones IP libres y utilizarlas al crear una interfaz de red elástica.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKS_CNI_Policy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSClusterPolicy

Puede adjuntar la `AmazonEKSClusterPolicy` a sus entidades de IAM. Antes de crear un clúster, debe tener un Rol de IAM de clúster con esta política adjunta. Los clústeres de Kubernetes administrados por Amazon EKS realizan llamadas a otros servicios de AWS en su nombre. Lo hacen para administrar los recursos que utiliza con el servicio.

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas:

- **`autoscaling`**: lee y actualiza la configuración de un grupo de Auto Scaling. Amazon EKS no utiliza estos permisos, pero permanecen en la política de compatibilidad con versiones anteriores.
- **`ec2`**: trabaja con volúmenes y recursos de red asociados a nodos de Amazon EC2. Esto es necesario para que el plano de control de Kubernetes pueda unir instancias a un clúster y aprovisionar y administrar de forma dinámica los volúmenes de Amazon EBS solicitados por los volúmenes persistentes de Kubernetes.
- **`elasticloadbalancing`**: trabaja con los Elastic Load Balancer y les agrega nodos como destinos. Esto es necesario para que el plano de control de Kubernetes pueda aprovisionar de forma dinámica los equilibradores de carga elástica solicitados por los servicios de Kubernetes.
- **`iam`**: crea un rol vinculado a servicios. Esto es necesario para que el plano de control de Kubernetes pueda aprovisionar de forma dinámica los Elastic Load Balancer solicitados por los servicios de Kubernetes.
- **`kms`**: lee una clave de AWS KMS. Esto es necesario para que el plano de control de Kubernetes admita el cifrado de secretos de Kubernetes de secretos de Kubernetes almacenados en `etcd`.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSClusterPolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSFargatePodExecutionRolePolicy

Puede adjuntar la `AmazonEKSFargatePodExecutionRolePolicy` a sus entidades de IAM. Antes de crear un perfil de Fargate, debe crear un rol de ejecución de Pod de Fargate y adjuntarle esta política. Para obtener más información, consulte Crear un rol de ejecución de Pod de Fargate y Perfil de AWS Fargate.

Esta política concede al rol los permisos que proporcionan acceso a otros recursos de servicio de AWS necesarios para ejecutar Pods de Amazon EKS en Fargate.

Detalles sobre los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas:

- **`ecr`**: permite que los pods que se ejecutan en Fargate extraigan imágenes de contenedor almacenadas en Amazon ECR.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSFargatePodExecutionRolePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSForFargateServiceRolePolicy

No puede adjuntar `AmazonEKSForFargateServiceRolePolicy` a sus entidades IAM. Esta política se encuentra adjunta a un rol vinculado a servicios que permite a Amazon EKS realizar acciones en su nombre. Para obtener más información, consulte AWSServiceRoleforAmazonEKSForFargate.

Esta política concede los permisos necesarios a Amazon EKS para ejecutar tareas de Fargate. Solo se utiliza la política si cuenta con nodos de Fargate.

Detalles de los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas.

- **`ec2`**: crea y elimina interfaces de red elásticas y describe los recursos y las interfaces de red elásticas. Esto es necesario a fin de que el servicio Fargate de Amazon EKS pueda configurar las redes VPC necesarias para los pods de Fargate.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSForFargateServiceRolePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSServicePolicy

Puede adjuntar la `AmazonEKSServicePolicy` a sus entidades de IAM. Los clústeres creados antes del 16 de abril de 2020 requerían que creara un rol de IAM y le adjuntara esta política. Los clústeres creados a partir del 16 de abril de 2020 no requieren que cree un rol ni que asigne esta política. Cuando crea un clúster mediante una entidad principal de IAM que tiene el permiso `iam:CreateServiceLinkedRole`, el rol vinculado al servicio AWSServiceRoleforAmazonEKS se crea de forma automática. El rol vinculado a servicios cuenta con la Política administrada por AWS: AmazonEKSServiceRolePolicy adjunta.

Esta política permite a Amazon EKS crear y administrar los recursos necesarios para operar clústeres de Amazon EKS.

Detalles de los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas.

- **`eks`**: actualiza la versión de Kubernetes de su clúster después de iniciar una actualización. Amazon EKS no utiliza este permiso, pero permanece en la política de compatibilidad con versiones anteriores.
- **`ec2`**: trabaja con interfaces de red elásticas y otros recursos y etiquetas de red. Amazon EKS lo requiere para configurar redes que faciliten la comunicación entre nodos y el plano de control de Kubernetes.
- **`route53`**: asocia una VPC con una zona alojada. Amazon EKS lo requiere a fin de habilitar las redes privadas de punto de conexión para su servidor de API de clúster de Kubernetes.
- **`logs`**: registra eventos. Esto es necesario para que Amazon EKS pueda enviar registros de planos de control de Kubernetes a CloudWatch.
- **`iam`**: crea un rol vinculado a servicios. Esto es necesario para que Amazon EKS can cree el rol vinculado al servicio `AWSServiceRoleForAmazonEKS` en su nombre.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSServicePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSServiceRolePolicy

No puede adjuntar `AmazonEKSServiceRolePolicy` a sus entidades IAM. Esta política se encuentra adjunta a un rol vinculado a servicios que permite a Amazon EKS realizar acciones en su nombre. Para obtener más información, consulte Permisos de roles vinculados a servicios para Amazon EKS. Al crear un clúster mediante una entidad principal de IAM que tiene el permiso `iam:CreateServiceLinkedRole`, el rol vinculado a servicios AWSServiceRoleforAmazonEKS se crea de forma automática en su nombre y se le asocia esta política.

Esta política permite al rol vinculado a servicios llamar a servicios de AWS en su nombre.

Detalles de los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas.

- **`ec2`**: crea y describe las interfaces de red elásticas y las instancias de Amazon EC2, el grupo de seguridad de clúster y la VPC necesarios para la creación de clústeres.

- **`iam`**: enumera todas las políticas administradas que se adjuntan a un rol de IAM. Esto es necesario para que Amazon EKS pueda enumerar y validar todos los permisos y políticas administrados necesarios para crear un clúster.
- Asocie un VPC con una zona alojada: Amazon EKS lo requiere a fin de habilitar las redes privadas de punto de conexión para su servidor de API de clúster de Kubernetes.
- Evento de registro: esto es necesario para que Amazon EKS pueda enviar registros de planos de control de Kubernetes a CloudWatch.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSServiceRolePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSVPCResourceController

Puede adjuntar la política `AmazonEKSVPCResourceController` a las identidades de IAM. Si utiliza grupos de seguridad de Pods, debe asociar esta política a su Rol de IAM del clúster de Amazon EKS para realizar acciones en su nombre.

Esta política concede permisos al rol de clúster para administrar las interfaces de red elásticas y las direcciones IP de los nodos.

Detalles sobre los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas:

- **`ec2`**: administra interfaces de red elásticas y direcciones IP para admitir grupos de seguridad de Pod y nodos de Windows.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSVPCResourceController en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEKSWorkerNodePolicy

No puede adjuntar la `AmazonEKSWorkerNodePolicy` a sus entidades de IAM. Debe asociar esta política a un rol de IAM de nodo que especifique al crear nodos de Amazon EC2 que permiten a Amazon EKS realizar acciones en su nombre. Si crea un grupo de nodos con `eksctl`, crea el rol de IAM de nodo y adjunta esta política al rol de forma automática.

Esta política concede permisos a los nodos de Amazon EC2 de Amazon EKS para conectarse a los clústeres de Amazon EKS.

Detalles sobre los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas:

- **`ec2`**: lee el volumen de la instancia y la información de red. Esto es necesario para que los nodos de Kubernetes puedan describir información sobre los recursos de Amazon EC2 necesarios a fin de que el nodo se una al clúster de Amazon EKS.
- **`eks`**: opcionalmente, describe el clúster como parte del arranque de nodos.
- **`eks-auth:AssumeRoleForPodIdentity`**: Permite la recuperación de credenciales para las cargas de trabajo de EKS en el nodo. Esto es necesario para que la Pod Identity de EKS funcione correctamente.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSWorkerNodePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada de AWS: AWSServiceRoleForAmazonEKSNodegroup

No puede adjuntar `AWSServiceRoleForAmazonEKSNodegroup` a sus entidades IAM. Esta política se encuentra adjunta a un rol vinculado a servicios que permite a Amazon EKS realizar acciones en su nombre. Para obtener más información, consulte Permisos de roles vinculados a servicios para Amazon EKS.

Esta política otorga permisos de rol de `AWSServiceRoleForAmazonEKSNodegroup` que permiten crear y administrar grupos de nodos de Amazon EC2 en su cuenta.

Detalles sobre los permisos

Esta política incluye los siguientes permisos que permiten a Amazon EKS completar las siguientes tareas:

- **`ec2`**: trabaja con grupos de seguridad, etiquetas y plantillas de lanzamiento. Esto es necesario para que los grupos de nodos administrados por Amazon EKS habiliten la configuración de acceso remoto. Además, los grupos de nodos administrados por Amazon EKS crean una plantilla de lanzamiento en su nombre. Esto es para configurar el grupo de Amazon EC2 Auto Scaling que respalda cada grupo de nodos administrados.

- **`iam`**: crea un rol vinculado a servicios y transfiere un rol. Los grupos de nodos administrados por Amazon EKS lo requieren para administrar los perfiles de instancias del rol que se transfiere al crear un grupo de nodos administrados. Las instancias de Amazon EC2 lanzadas como parte de un grupo de nodos administrados utilizan este perfil de instancias. Amazon EKS necesita crear roles vinculadas a servicios para otros servicios, como los grupos de Amazon EC2 Auto Scaling. Estos permisos se usan en la creación de un grupo de nodos administrados.
- **`autoscaling`**: trabaja con grupos de Auto Scaling de seguridad. Los grupos de nodos administrados por Amazon EKS lo requieren para administrar el grupo de Amazon EC2 Auto Scaling que respalda cada grupo de nodos administrados. También se utiliza para admitir funciones como expulsar Pods cuando los nodos se terminan o reciclan durante las actualizaciones de grupos de nodos.

Para ver la versión más reciente del documento de política JSON, consulte AWSServiceRoleForAmazonEKSNodegroup en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEBSCSIDriverPolicy

La política `AmazonEBSCSIDriverPolicy` permite que el controlador de la Interfaz de almacenamiento de contenedores (CSI) de Amazon EBS cree, modifique, asocie, desasocie y elimine volúmenes en su nombre. También otorga al controlador CSI de EBS permisos para crear y eliminar instantáneas, así como para enumerar instancias, volúmenes e instantáneas.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEBSCSIDriverServiceRolePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada por AWS: AmazonEFSCSIDriverPolicy

La política de `AmazonEFSCSIDriverPolicy` permite a la interfaz de almacenamiento de contenedores (CSI) de Amazon EFS crear y eliminar puntos de acceso en su nombre. También otorga permisos al controlador CSI de Amazon EFS para enumerar sus puntos de acceso, sistemas de archivos, destinos de montaje y zonas de disponibilidad de Amazon EC2.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEFSCSIDriverServiceRolePolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada de AWS: AmazonEKSLocalOutpostClusterPolicy

Puede asociar esta política a entidades de IAM. Antes de crear un clúster local, debe asociar esta política a su rol de clúster. Los clústeres de Kubernetes administrados por Amazon EKS realizan llamadas a otros servicios de AWS en su nombre. Lo hacen para administrar los recursos que utiliza con el servicio.

La `AmazonEKSLocalOutpostClusterPolicy` incluye los siguientes permisos:

- **`ec2`**: permisos necesarios para que las instancias de Amazon EC2 se unan correctamente al clúster como instancias del plano de control.
- **`ssm`**: permite la conexión de Amazon EC2 Systems Manager a la instancia del plano de control, que Amazon EKS usa para comunicar y administrar el clúster local de su cuenta.
- **`logs`**: permite que las instancias envíen registros a Amazon CloudWatch.
- **`secretsmanager`**: permite a las instancias obtener y eliminar los datos de arranque de las instancias del plano de control de forma segura desde AWS Secrets Manager.
- **`ecr`**: permite que los Pods y contenedores que se ejecutan en las instancias del plano de control extraigan imágenes de contenedor almacenadas en Amazon Elastic Container Registry.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSLocalOutpostClusterPolicy en la Guía de referencia de políticas administradas de AWS.

Política administrada de AWS: AmazonEKSLocalOutpostServiceRolePolicy

No puede adjuntar esta política a sus entidades de IAM. Al crear un clúster mediante una entidad principal de IAM que tiene el permiso `iam:CreateServiceLinkedRole`, Amazon EKS crea de forma automática el rol vinculado a servicios `AWSServiceRoleforAmazonEKSLocalOutpost` en su nombre y le asocia esta política. Esta política permite al rol vinculado a servicios llamar a servicios de AWS para clústeres locales en su nombre.

La `AmazonEKSLocalOutpostServiceRolePolicy` incluye los siguientes permisos:

- **`ec2`**: permite a Amazon EKS trabajar con la seguridad, la red y otros recursos para lanzar y administrar correctamente las instancias del plano de control en su cuenta.
- **`ssm`**: permite la conexión de Amazon EC2 Systems Manager a las instancias del plano de control, que Amazon EKS usa para comunicar y administrar el clúster local de su cuenta.
- **`iam`**: permite a Amazon EKS administrar el perfil de instancia asociado a las instancias del plano de control.

- **`secretsmanager`**: permite a Amazon EKS colocar los datos de arranque de las instancias del plano de control en AWS Secrets Manager para que pueda consultarse de forma segura durante el arranque de la instancia.
- **`outposts`**: permite a Amazon EKS obtener información de Outpost de su cuenta para lanzar correctamente un clúster local en un Outpost.

Para ver la versión más reciente del documento de política JSON, consulte AmazonEKSLocalOutpostServiceRolePolicy en la Guía de referencia de políticas administradas de AWS.

Actualizaciones de Amazon EKS en las políticas administradas por AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas por AWS para Amazon EKS debido a que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de Amazon EKS.

Cambio	Descripción	Fecha
Amazoneks_CNI_Policy: actualización de una política existente	Amazon EKS ha añadido nuevos permisos `ec2:DescribeSubnets` para Amazon VPC CNI plugin for Kubernetes para poder ver la cantidad de direcciones IP libres en las subredes de Amazon VPC. La CNI de la VPC puede usar las direcciones IP libres de cada subred para seleccionar las subredes con la mayor cantidad de direcciones IP libres y utilizarlas al crear una interfaz de red elástica.	4 de marzo de 2024

Cambio	Descripción	Fecha
AmazonEksWorkerNodePolicy: Actualización de una política existente	Amazon EKS agregó nuevos permisos para permitir las Pod Identities de EKS. El agente de Pod Identity de Amazon EKS usa el rol de nodo.	26 de noviembre de 2023
Se presentó AmazonEFSCSIDriverPolicy.	AWS presentó la `AmazonEFSCSIDriverPolicy` .	26 de julio de 2023
Se agregaron permisos a AmazonEKSClusterPolicy.	Se agregó un permiso de `ec2:DescribeAvailabilityZones` para permitir que Amazon EKS obtenga los detalles de AZ durante la detección automática de subredes al crear equilibradores de carga.	7 de febrero de 2023
Condiciones de la política actualizadas en AmazonEBSCSIdRiverPolicy.	Eliminación de condiciones de política no válidas con caracteres comodín en el campo clave de `StringLike` . También se agregó una nueva condición `ec2:ResourceTag/kubernetes.io/created-for/pvc/name: "*"` a `ec2:DeleteVolume` , que permite al controlador CSI de EBS eliminar los volúmenes creados por el complemento integrado en el árbol.	17 de noviembre de 2022

Cambio	Descripción	Fecha
Se agregaron permisos a AmazonEKSLocalOutpostServiceRolePolicy.	Se agregó `ec2:DescribeVPCAttribute` , `ec2:GetConsoleOutput` y `ec2:DescribeSecret` para permitir una mejor validación de los requisitos previos y un control del ciclo de vida administrado. También se agregó `ec2:DescribePlacementGroups` y `"arn:aws:ec2:*:*:placement-group/*"` a `ec2:RunInstances` para admitir el control de ubicación de las instancias de Amazon EC2 del plano de control en Outposts.	24 de octubre de 2022
Actualice los permisos del registro de Amazon Elastic Container Registry en AmazonEKSLocalOutpostClusterPolicy.	Se movió la acción `ecr:GetDownloadUrlForLayer` de todas las secciones de recursos a una sección específica. Se agregó el recurso `arn:aws:ecr:*:*:repository/eks/*` . Se eliminó el recurso `arn:aws:ecr:*:*:repository/eks/eks-certificates-controller-public` . Este recurso está cubierto por el recurso agregado `arn:aws:ecr:*:*:repository/eks/*` .	20 de octubre de 2022
Se agregaron permisos a AmazonEKSLocalOutpostClusterPolicy.	Se agregó el repositorio de Amazon Elastic Container Registry `arn:aws:ecr:*:*:repository/kubelet-config-updater` para que las instancias del plano de control del clúster puedan actualizar algunos argumentos de `kubelet`.	31 de agosto de 2022

Cambio	Descripción	Fecha
Se presentó AmazonEKS LocalOutpostClusterPolicy.	AWS presentó la `AmazonEKS LocalOutpostClusterPolicy` .	24 de agosto de 2022
Se presentó AmazonEKS LocalOutpostServiceRolePolicy.	AWS presentó la `AmazonEKS LocalOutpostServiceRolePolicy` .	23 de agosto de 2022
Se presentó AmazonEBS CSIDriverPolicy.	AWS presentó la `AmazonEBS CSIDriverPolicy` .	4 de abril de 2022
Se agregaron permisos a AmazonEKSWorkerNodePolicy.	Se agregó `ec2:DescribeInstanceTypes` para habilitar las AMI optimizadas de Amazon EKS que pueden detectar en forma automática las propiedades de nivel de instancia.	21 de marzo de 2022
Se agregaron permisos a AWSServiceRoleForAmazonEKSNodegroup.	Se agregó el permiso `autoscaling:EnableMetricsCollection` para permitir que Amazon EKS habilite la recopilación de métricas.	13 de diciembre de 2021
Se agregaron permisos a AmazonEKSClusterPolicy.	Se han agregados permisos de `ec2:DescribeAccountAttributes` , `ec2:DescribeAddresses` y `ec2:DescribeInternetGateways` a fin de permitir que Amazon EKS cree un rol vinculado a servicios para un equilibrador de carga de red.	17 de junio de 2021
Amazon EKS comenzó a realizar el seguimiento de los cambios.	Amazon EKS comenzó a realizar el seguimiento de los cambios de las políticas administradas por AWS.	17 de junio de 2021

Solución de problemas de IAM

En este tema se tratan algunos errores habituales que pueden aparecer al utilizar Amazon EKS con IAM y cómo solucionarlos.

AccessDeniedException

Si recibe una `AccessDeniedException` al llamar a una operación de API de AWS, las credenciales de la entidad principal de IAM que utiliza no tienen los permisos necesarios para hacer esa llamada.

```
An error occurred (AccessDeniedException) when calling the DescribeCluster operation:
User: arn:aws:iam::111122223333:user/user_name is not authorized to perform:
eks:DescribeCluster on resource: arn:aws:eks:region:111122223333:cluster/my-cluster
```

En el mensaje de ejemplo anterior, el usuario no tiene permisos para llamar a la operación `DescribeCluster` de la API de Amazon EKS. Para proporcionar permisos de administrador de Amazon EKS a una entidad principal de IAM, consulte Ejemplos de políticas de Amazon EKS basadas en identidades.

Para obtener información general sobre IAM, consulte Control del acceso con políticas en la Guía del usuario de IAM.

No puedo ver Nodos en la pestaña Compute (Informática) o cualquier cosa de la pestaña Resources (Recursos) y recibe un error en la AWS Management Console

Es posible que aparezca un mensaje de error en la consola que dice `Your current user or role does not have access to Kubernetes objects on this EKS cluster`. Asegúrese de que el usuario principal de IAM con el que está utilizando la AWS Management Console tenga los permisos necesarios. Para obtener más información, consulte Permisos necesarios.

El **`ConfigMap`** de aws-auth no concede acceso al clúster

El autenticador de IAM de AWS no permite una ruta de acceso en el ARN de rol utilizado en el `ConfigMap`. Por lo tanto, antes de especificar `rolearn`, elimine la ruta de acceso. Por ejemplo, cambie `arn:aws:iam::111122223333:role/team/developers/eks-admin` a `arn:aws:iam::111122223333:role/eks-admin`.

No tengo autorización para realizar la operación iam:PassRole

Si recibe un error que indica que no tiene autorización para llevar a cabo la acción `iam:PassRole`, sus políticas deben actualizarse para permitirle pasar un rol a Amazon MQ.

Algunos servicios de Servicios de AWS le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon EKS. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi cuenta de AWS accedan a mis recursos de Amazon EKS.

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si Amazon EKS admite estas características, consulte Cómo funciona Amazon EKS con IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuenta de AWS de su propiedad, consulte Proporcionar acceso a un usuario de IAM a otra cuenta de AWS de la que es propietario en la Guía del usuario de IAM.

- Para obtener información acerca de cómo proporcionar acceso a tus recursos a Cuentas de AWS de terceros, consulte Proporcionar acceso a cuentas de AWS que son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Los contenedores de pods muestran el siguiente error: **`An error occurred (SignatureDoesNotMatch) when calling the GetCallerIdentity operation: Credential should be scoped to a valid region`**

Los contenedores reciben este error si la aplicación realiza solicitudes explícitamente al punto de conexión AWS STS global (`https://sts.amazonaws`) y su cuenta de servicio de Kubernetes está configurada para utilizar un punto de conexión regional. Puede resolver el problema con una de las siguientes opciones:

- Actualice el código de la aplicación para eliminar las llamadas explícitas al punto de conexión global de AWS STS.
- Actualice el código de la aplicación para realizar llamadas explícitas a puntos de conexión regionales, como `https://sts.us-west-2.amazonaws.com`. Su aplicación debe tener redundancia integrada para seleccionar una Región de AWS diferente en caso de producirse un error del servicio en la Región de AWS. Para obtener más información, consulte Administración de AWS STS en una Región de AWS en la guía del usuario de IAM.
- Configure sus cuentas de servicio para utilizar el punto de conexión global. Todas las versiones anteriores a `1.22` utilizaron el punto de conexión global de forma predeterminada, pero versión `1.22` y los clústeres posteriores utilizan el punto de conexión regional de forma predeterminada. Para obtener más información, consulte Configure el punto de conexión AWS Security Token Service de una cuenta de servicio.

Roles y usuarios predeterminados de Kubernetes creados por Amazon EKS

Al crear un clúster de Kubernetes, se crean varias identidades de Kubernetes predeterminadas en ese clúster para el correcto funcionamiento de Kubernetes. Amazon EKS crea identidades de Kubernetes para cada uno de sus componentes predeterminados. Las identidades proporcionan un control de autorización de Kubernetes basado en roles (RBAC) para los componentes del clúster. Para obtener más información, consulte Uso de la autorización de RBAC en la documentación de Kubernetes.

Al instalar complementos opcionales en el clúster, es posible que se agreguen identidades de Kubernetes adicionales al clúster. Para obtener más información sobre las identidades no abordadas en este tema, consulte la documentación del complemento.

Puede ver la lista de identidades de Kubernetes creadas por Amazon EKS en el clúster mediante la herramienta de línea de comandos de AWS Management Console o `kubectl`. Todas las identidades de usuario aparecen en los registros de auditoría de `kube` disponibles para los clientes a través de Amazon CloudWatch.

AWS Management Console

Requisito previo

La entidad principal de IAM que utilice debe tener los permisos que se describen en Permisos necesarios.

Para ver las identidades creadas por Amazon EKS mediante la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En la lista Clusters (Clústeres), seleccione el clúster que contiene las identidades que desea ver.
3. Elija la pestaña Recursos.
4. En Resource types (Tipos de recursos), elija Authorization (Autorización).
5. Elija ClusterRoles, ClusterRoleBindings, Roles o RoleBindings. Amazon EKS crea todos los recursos con el prefijo eks. Los recursos de identidad adicionales creados por Amazon EKS son los siguientes:

    - El ClusterRole y el ClusterRoleBinding, que se denominan aws-node. Los recursos de aws-node admiten el Amazon VPC CNI plugin for Kubernetes, que Amazon EKS instala en todos los clústeres.

- Un ClusterRole llamado vpc-resource-controller-role y un ClusterRoleBinding llamado vpc-resource-controller-rolebinding. Estos recursos son compatibles con el controlador de recursos de Amazon VPC, que Amazon EKS instala en todos los clústeres.

Además de los recursos que ve en la consola, existen las siguientes identidades de usuario especiales en su clúster, aunque no están visibles en la configuración del clúster:

- **`eks:cluster-bootstrap`**: se utiliza para operaciones de `kubectl` durante el arranque del clúster.
- **`eks:support-engineer`**: se utiliza para operaciones de administración de clústeres.

6. Elija un recurso específico para ver detalles sobre él. De forma predeterminada, se muestra la información en la vista estructurada. En la esquina superior derecha de la página de detalles de la, elija la vista de sin procesar para ver toda la información del recurso.

Kubectl

Requisito previo

La entidad que utilice (AWS Identity and Access Management [IAM] o OpenID Connect [OIDC]) para enumerar los recursos de Kubernetes del clúster debe estar autenticada por IAM o por su proveedor de identidad de OIDC. Se deben conceder permisos a la entidad para usar los verbos de Kubernetes `get` y `list` de los recursos del clúster `Role`, `ClusterRole`, `RoleBinding` y `ClusterRoleBinding` con los que desea que trabaje la entidad. Para obtener más información sobre cómo conceder acceso de entidades de IAM a su clúster, consulte the section called “Concesión de acceso a las API de Kubernetes”. Para obtener más información sobre cómo conceder acceso de entidades auntenticadas mediante su propio proveedor de OIDC a su clúster, consulte Autenticación de usuarios para el clúster desde un proveedor de identidad de OpenID Connect.

Para ver las identidades creadas por Amazon EKS mediante **`kubectl`**

Ejecute el comando para el tipo de recurso que desea ver. Todos los recursos devueltos que llevan el prefijo eks son creados por Amazon EKS. Además de los recursos devueltos en la salida de los comandos, existen las siguientes identidades de usuario especiales en su clúster, aunque no están visibles en la configuración del clúster:

- **`eks:cluster-bootstrap`**: se utiliza para operaciones de `kubectl` durante el arranque del clúster.

- **`eks:support-engineer`**: se utiliza para operaciones de administración de clústeres.

ClusterRoles: `ClusterRoles` está dentro del ámbito de su clúster, por lo que cualquier permiso otorgado a un rol se aplica a los recursos de cualquier espacio de nombres de Kubernetes del clúster.

El siguiente comando devuelve todos los `ClusterRoles` de Kubernetes de Amazon EKS creados en su clúster.

```
kubectl get clusterroles | grep eks
```

Además de los `ClusterRoles` devueltos en la salida con el prefijo, existen los siguientes `ClusterRoles`.

- **`aws-node`**: este `ClusterRole` es compatible con Amazon VPC CNI plugin for Kubernetes, que Amazon EKS instala en todos los clústeres.
- **`vpc-resource-controller-role`**: este `ClusterRole` es compatible con el controlador de recursos de Amazon VPC, que Amazon EKS instala en todos los clústeres.

Para ver la especificación de un `ClusterRole`, sustituya *`eks:k8s-metrics`* en el siguiente comando por el `ClusterRole` devuelto en el resultado del comando anterior. El siguiente ejemplo devuelve la especificación de `ClusterRole` *`eks:k8s-metrics`*.

```
kubectl describe clusterrole eks:k8s-metrics
```

Un ejemplo de salida sería el siguiente.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources         Non-Resource URLs  Resource Names  Verbs
  ---------         -----------------  --------------  -----
                    [/metrics]         []              [get]
  endpoints         []                 []              [list]
  nodes             []                 []              [list]
  pods              []                 []              [list]
```

```
  deployments.apps  []                []              [list]
```

ClusterRoleBindings: `ClusterRoleBindings` está dentro del ámbito de su clúster.

El siguiente comando devuelve todos los `ClusterRoleBindings` de Kubernetes de Amazon EKS creados en su clúster.

```
kubectl get clusterrolebindings | grep eks
```

Además de los `ClusterRoleBindings` devueltos en la salida, existen los siguientes `ClusterRoleBindings`.

- **`aws-node`**: este `ClusterRoleBinding` es compatible con Amazon VPC CNI plugin for Kubernetes, que Amazon EKS instala en todos los clústeres.
- **`vpc-resource-controller-rolebinding`**: este `ClusterRoleBinding` es compatible con el controlador de recursos de Amazon VPC, que Amazon EKS instala en todos los clústeres.

Para ver la especificación de un `ClusterRoleBinding`, sustituya *`eks:k8s-metrics`* en el siguiente comando por el `ClusterRoleBinding` devuelto en el resultado del comando anterior. El siguiente ejemplo devuelve la especificación de `ClusterRoleBinding` *`eks:k8s-metrics`*.

```
kubectl describe clusterrolebinding eks:k8s-metrics
```

Un ejemplo de salida sería el siguiente.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  ClusterRole
  Name:  eks:k8s-metrics
Subjects:
  Kind  Name             Namespace
  ----  ----             ---------
  User  eks:k8s-metrics
```

Roles: `Roles` se limitan a un espacio de nombres de Kubernetes. Todos los `Roles` de Amazon EKS creados están incluidos en el espacio de nombres de `kube-system`.

El siguiente comando devuelve todos los `Roles` de Kubernetes de Amazon EKS creados en su clúster.

```
kubectl get roles -n kube-system | grep eks
```

Para ver la especificación de un `Role`, sustituya *`eks:k8s-metrics`* en el siguiente comando por el nombre del `Role` devuelto en el resultado del comando anterior. El siguiente ejemplo devuelve la especificación de `Role` *`eks:k8s-metrics`*.

```
kubectl describe role eks:k8s-metrics -n kube-system
```

Un ejemplo de salida sería el siguiente.

```
Name:         eks:k8s-metrics
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources         Non-Resource URLs  Resource Names             Verbs
  ---------         -----------------  --------------             -----
  daemonsets.apps   []                 [aws-node]                 [get]
  deployments.apps  []                 [vpc-resource-controller]  [get]
```

RoleBindings: `RoleBindings` se circunscriben a un espacio de nombres de Kubernetes. Todos los `RoleBindings` de Amazon EKS creados están incluidos en el espacio de nombres de `kube-system`.

El siguiente comando devuelve todos los `RoleBindings` de Kubernetes de Amazon EKS creados en su clúster.

```
kubectl get rolebindings -n kube-system | grep eks
```

Para ver la especificación de un `RoleBinding`, sustituya *`eks:k8s-metrics`* en el siguiente comando por el `RoleBinding` devuelto en el resultado del comando anterior. El siguiente ejemplo devuelve la especificación de `RoleBinding` *`eks:k8s-metrics`*.

```
kubectl describe rolebinding eks:k8s-metrics -n kube-system
```

Un ejemplo de salida sería el siguiente.

```
Name:         eks:k8s-metrics
```

```
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  Role
  Name:  eks:k8s-metrics
Subjects:
  Kind  Name             Namespace
  ----  ----             ---------
  User  eks:k8s-metrics
```

Validación de conformidad para Amazon Elastic Kubernetes Service

Para saber si un Servicio de AWS está incluido en el ámbito de programas de conformidad específicos, consulte Servicios de AWS en el ámbito del programa de conformidad y elija el programa de conformidad que le interese. Para obtener información general, consulte Programas de conformidad de AWS.

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Su responsabilidad de conformidad al utilizar Servicios de AWS se determina en función de la sensibilidad de los datos, los objetivos de cumplimiento de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- Guías de inicio rápido de seguridad y conformidad: estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Architecting for HIPAA Security and Compliance on Amazon Web Services (Arquitectura para la seguridad y el cumplimiento de la HIPAA en Amazon Web Services): en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones aptas para HIPAA.

> **Note**
>
> No todos los Servicios de AWS son aptos para HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- Recursos de conformidad de AWS: este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- Guías de cumplimiento para clientes de AWS: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad de los Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés), el Consejo de Estándares de Seguridad de la Industria de Tarjetas de Pago (PCI, por sus siglas en inglés) y la Organización Internacional de Normalización (ISO, por sus siglas en inglés)).
- Evaluación de recursos con reglas en la Guía para desarrolladores de AWS Config: el servicio AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- AWS Security Hub: este Servicio de AWS proporciona una visión completa de su estado de seguridad en AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: este Servicio de AWS detecta posibles amenazas para sus Cuentas de AWS, cargas de trabajo, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a satisfacer varios requisitos de conformidad, como PCI DSS, cumpliendo los requisitos de detección de intrusos que exigen determinados marcos de conformidad.
- AWS Audit Manager: este Servicio de AWS le ayuda a auditar continuamente el uso de AWS con el fin de simplificar la forma en que administra el riesgo y la conformidad con las normativas y los estándares del sector.

Resiliencia en Amazon EKS

La infraestructura global de AWS se divide en Regiones de AWS y zonas de disponibilidad. Las Regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Amazon EKS ejecuta y escala el plano de control de Kubernetes en varias zonas de disponibilidad de AWS para garantizar una alta disponibilidad. Amazon EKS escala de forma automática las instancias del plano de control en función de la carga, detecta y reemplaza instancias del plano de control en mal estado y revisa el plano de control de forma automática. Después de iniciar una actualización de versión, Amazon EKS actualiza el plano de control en su nombre y mantiene una alta disponibilidad durante la actualización.

Este plano de control consta de al menos dos instancias de servidor de la API y tres instancias `etcd` que se ejecutan en tres zonas de disponibilidad de una Región de AWS. Amazon EKS:

- Monitorea de forma activa la carga en las instancias del plano de control y las escala de forma automática para garantizar un alto rendimiento.
- Detecta y reemplaza de forma automática las instancias del plano de control en mal estado y las reinicia en las zonas de disponibilidad de la Región de AWS, según sea necesario.
- Aprovecha la arquitectura de las Regiones de AWS con el fin de mantener una alta disponibilidad. Por este motivo, Amazon EKS puede ofrecer un acuerdo de nivel de servicio (SLA) de disponibilidad del punto de conexión del servidor de la API.

Para obtener más información sobre las Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global de AWS.

Seguridad de la infraestructura de Amazon EKS

Como servicio administrado, Amazon Elastic Kubernetes Service está protegido por la seguridad de la red global AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte Seguridad en la nube de AWS. Para diseñar su entorno de AWS conforme a las prácticas recomendadas de seguridad de la infraestructura, consulte Protección de la infraestructura en Pilar de seguridad del Marco de AWS Well-Architected.

Puede utilizar llamadas a la API publicadas en AWS para acceder a Amazon EKS a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Cuando se crea un clúster de Amazon EKS, se especifican las subredes de la VPC que utilizará el clúster. Amazon EKS requiere subredes en al menos dos zonas de disponibilidad. Recomendamos una VPC con subredes públicas y privadas para que Kubernetes pueda crear equilibradores de carga públicos en las subredes públicas que equilibren la carga de tráfico con los Pods que se ejecutan en los nodos de las subredes privadas.

Para obtener información acerca de las consideraciones de VPC, consulte Requisitos y consideraciones de Amazon EKS VPC y subred.

Si crea la VPC y los grupos de nodos con las plantillas de AWS CloudFormation incluidas en la explicación de Introducción a Amazon EKS, los grupos de seguridad del plano de control y los nodos de los grupos de trabajo se ajustan con la configuración recomendada.

Para obtener más información acerca de las consideraciones del grupo de seguridad, consulte Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS.

Cuando se crea un clúster nuevo, Amazon EKS crea un punto de conexión para el servidor de la API de Kubernetes administrado que utiliza a fin de comunicarse con su clúster (mediante herramientas de administración de Kubernetes como, por ejemplo, `kubectl`). De forma predeterminada, este punto de conexión del servidor de la API es público en Internet y el acceso al servidor de la API está protegido mediante una combinación de AWS Identity and Access Management (IAM) y el Control de acceso basado en rol (RBAC) nativo de Kubernetes.

Puede habilitar el acceso privado al servidor de la API de Kubernetes para que toda la comunicación entre los nodos y el servidor de la API permanezcan dentro de su VPC. Puede limitar las direcciones IP que pueden acceder a su servidor de API desde Internet o desactivar por completo el acceso a Internet al servidor de API.

Para obtener más información acerca de la modificación del acceso al punto de conexión del clúster, consulte Modificar el acceso al punto de conexión del clúster.

Puede implementar políticas de red de Kubernetes con el CNI de Amazon VPC o con herramientas de terceros, como Proyecto Calico. Para obtener más información sobre el CNI de Amazon VPC para las políticas de red, consulte Configure su clúster para las políticas de red de Kubernetes. Project

Calico es un proyecto abierto de terceros. Para obtener más información, consulte la documentación de Project Calico.

Configuración y análisis de vulnerabilidades en Amazon EKS

La seguridad es una consideración fundamental para configurar y mantener clústeres y aplicaciones de Kubernetes. A continuación, se enumeran los recursos disponibles para que pueda analizar la configuración de seguridad de sus clústeres de EKS y comprobar si hay vulnerabilidades, y las integraciones con servicios de AWS que pueden realizar ese análisis por usted.

Referencia del Center for Internet Security (CIS, Centro para la seguridad de Internet) para Amazon EKS

El Punto de referencia de Kubernetes del Center for Internet Security (Centro para la seguridad de Internet, CIS) proporciona orientación para las configuraciones de seguridad de Amazon EKS. El punto de referencia:

- Es aplicable a los nodos de Amazon EC2 (administrados y autoadministrados) donde es responsable de las configuraciones de seguridad de los componentes de Kubernetes.
- Proporciona una forma estándar y aprobada por la comunidad de garantizar que ha configurado de forma segura el clúster y los nodos de Kubernetes al utilizar Amazon EKS.
- Consta de cuatro secciones: configuración de registro del plano de control, configuraciones de seguridad de nodos, políticas y servicios administrados.
- Admite todas las versiones de Kubernetes actualmente disponibles en Amazon EKS y se puede ejecutar con kube-bench, una herramienta estándar de código abierto para verificar la configuración mediante el punto de referencia del CIS en clústeres de Kubernetes.

Para obtener más información, consulte Presentación del punto de referencia de Amazon EKS del CIS.

Versiones de la plataforma de Amazon EKS

Las versiones de la plataforma de Amazon EKS representan las capacidades del plano de control del clúster, incluido las marcas de servidor de la API de Kubernetes que se encuentran habilitadas y la versión de parche de Kubernetes actual. Los nuevos clústeres están implementados con la versión más reciente de la plataforma. Para obtener más información, consulte Versiones de la plataforma de Amazon EKS.

Puede actualizar un clúster de Amazon EKS con las versiones más recientes de Kubernetes. Cuando haya nuevas versiones de Kubernetes disponibles en Amazon EKS, le recomendamos que actualice proactivamente los clústeres para que utilicen la versión más reciente disponible. Para obtener más información sobre las versiones de Kubernetes en EKS, consulte Versiones de Amazon EKS de Kubernetes.

Lista de vulnerabilidades del sistema operativo

Lista de vulnerabilidades de AL2023

Realice un seguimiento de los eventos de seguridad o privacidad de Amazon Linux 2023 en el Centro de seguridad de Amazon Linux o suscríbase a la fuente RSS asociada. Los eventos de seguridad y privacidad incluyen una descripción general del problema, los paquetes afectados e instrucciones para actualizar las instancias y corregir el problema.

Lista de vulnerabilidades Amazon Linux 2

Realice un seguimiento de los eventos de seguridad o privacidad de Amazon Linux 2 en el Centro de seguridad de Amazon Linux o suscríbase a la fuente RSS asociada. Los eventos de seguridad y privacidad incluyen una descripción general del problema, los paquetes afectados e instrucciones para actualizar las instancias y corregir el problema.

Detección de nodos con Amazon Inspector

Puede utilizar Amazon Inspector para verificar la accesibilidad de la red no deseada de sus nodos y a fin de buscar vulnerabilidades en dichas instancias de Amazon EC2.

Detección de clústeres y nodos con Amazon GuardDuty

Amazon GuardDuty es un servicio de detección de amenazas que ayuda a proteger las cuentas, los contenedores, las cargas de trabajo y los datos de su entorno de AWS. Entre otras características, GuardDuty ofrece las siguientes dos características que detectan posibles amenazas para sus clústeres de EKS: Protección de EKS y Supervisión en tiempo de ejecución.

Para obtener más información, consulte Detección de amenazas con Amazon GuardDuty.

Prácticas recomendadas de seguridad para Amazon EKS

Las prácticas recomendadas de seguridad de Amazon EKS se mantienen en Github: https://aws.github.io/aws-eks-best-practices/security/docs/

Política de seguridad del pod

El controlador de admisión de la política de seguridad del Pod de Kubernetes valida la creación del Pod y actualiza las solicitudes utilizando un conjunto de reglas. Los clústeres de Amazon EKS se envían con un política de seguridad totalmente permisiva y sin restricciones de forma predeterminada. Para obtener más información, consulte Políticas de seguridad del pod en la documentación de Kubernetes.

> **Note**
>
> La `PodSecurityPolicy` (PSP) quedó obsoleta en la versión de Kubernetes `1.21` y se eliminó en Kubernetes `1.25`. Los PSPs se sustituirán por Pod Security Admission (PSA), un controlador de admisión integrado que implementa los controles de seguridad descritos en los estándares de seguridad de pods (PSS). Tanto PSA como PSS han alcanzado el estado de características beta y están habilitadas en Amazon EKS de forma predeterminada. Para hacer frente a la eliminación de PSP en `1.25`, le recomendamos que implemente PSS en Amazon EKS. Para obtener más información, consulte Implementing Pod Security Standards in Amazon EKS (Implementación de estándares de seguridad para módulos en Amazon EKS) en el blog de AWS.

Política de seguridad predeterminada del Pod de Amazon EKS

Los clústeres de Amazon EKS con la versión `1.13` y posterior de Kubernetes tienen una política de seguridad de Pod predeterminada denominada `eks.privileged`. Esta política no presenta restricciones en cuanto al tipo de Pod que se puede aceptar en el sistema, lo que equivale a ejecutar Kubernetes con el controlador `PodSecurityPolicy` deshabilitado.

> **Note**
>
> Esta política se creó mantener la compatibilidad con los clústeres que no tenía el controlador `PodSecurityPolicy` habilitado. Puede crear políticas más restrictivas para su clúster y para los espacios de nombres individuales y cuentas de servicio y, a continuación, eliminar la política predeterminada para habilitar políticas más restrictivas.

Puede consultar la política predeterminada con el siguiente comando.

```
kubectl get psp eks.privileged
```

Un ejemplo de salida sería el siguiente.

```
NAME             PRIV   CAPS   SELINUX    RUNASUSER   FSGROUP    SUPGROUP
 READONLYROOTFS   VOLUMES
eks.privileged   true   *      RunAsAny   RunAsAny    RunAsAny   RunAsAny   false
      *
```

Para obtener más información, puede describir la política con el siguiente comando.

```
kubectl describe psp eks.privileged
```

Un ejemplo de salida sería el siguiente.

```
Name:  eks.privileged

Settings:
  Allow Privileged:                       true
  Allow Privilege Escalation:             0xc0004ce5f8
  Default Add Capabilities:               <none>
  Required Drop Capabilities:             <none>
  Allowed Capabilities:                   *
  Allowed Volume Types:                   *
  Allow Host Network:                     true
  Allow Host Ports:                       0-65535
  Allow Host PID:                         true
  Allow Host IPC:                         true
  Read Only Root Filesystem:              false
  SELinux Context Strategy: RunAsAny
    User:                                 <none>
    Role:                                 <none>
    Type:                                 <none>
    Level:                                <none>
  Run As User Strategy: RunAsAny
    Ranges:                               <none>
  FSGroup Strategy: RunAsAny
    Ranges:                               <none>
  Supplemental Groups Strategy: RunAsAny
    Ranges:                               <none>
```

Puede ver el archivo YAML completo de la política de seguridad del Pod `eks.privileged`, el rol de clúster y la vinculación del rol de clúster en Instale o restaure la política de seguridad del Pod predeterminada.

Eliminar la política de seguridad predeterminada del Pod de Amazon EKS

Si crea políticas más restrictivas para sus Pods, después de hacerlo, puede eliminar la política de seguridad del Pod `eks.privileged` de Amazon EKS predeterminada a fin de habilitar sus políticas personalizadas.

> **Important**
>
> Si utiliza la versión `1.7.0` o posterior del complemento CNI y asigna una política de seguridad de Pod personalizada a la cuenta de servicio de `aws-node` Kubernetes utilizada para los Pods de `aws-node` implementados por el Daemonset, la política debe contar con `NET_ADMIN` en su sección de `allowedCapabilities` junto con `hostNetwork: true` y `privileged: true` en el objeto `spec` de la política.

Para eliminar la política de seguridad de Pod predeterminada

1. Cree un archivo denominado *`privileged-podsecuritypolicy.yaml`* con el contenido del archivo de ejemplo en Instale o restaure la política de seguridad del Pod predeterminada.
2. Elimine el archivo YAML con el siguiente comando. Elimina la política de seguridad predeterminada del Pod, el `ClusterRole` y el `ClusterRoleBinding` asociado a él.

```
kubectl delete -f privileged-podsecuritypolicy.yaml
```

Instale o restaure la política de seguridad del Pod predeterminada

Si actualiza desde una versión anterior de Kubernetes o ha modificado o eliminado la política de seguridad del Pod `eks.privileged` de Amazon EKS, puede restaurarla con los pasos que se describen a continuación.

Para instalar o restaurar la política de seguridad del Pod predeterminada

1. Cree un archivo denominado *`privileged-podsecuritypolicy.yaml`* con el siguiente contenido.

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: eks.privileged
  annotations:
    kubernetes.io/description: 'privileged allows full unrestricted access to
      Pod features, as if the PodSecurityPolicy controller was not enabled.'
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
  labels:
    kubernetes.io/cluster-service: "true"
    eks.amazonaws.com/component: pod-security-policy
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
  readOnlyRootFilesystem: false

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: eks:podsecuritypolicy:privileged
  labels:
    kubernetes.io/cluster-service: "true"
    eks.amazonaws.com/component: pod-security-policy
rules:
```

```
- apiGroups:
  - policy
  resourceNames:
  - eks.privileged
  resources:
  - podsecuritypolicies
  verbs:
  - use

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks:podsecuritypolicy:authenticated
  annotations:
    kubernetes.io/description: 'Allow all authenticated users to create privileged
 Pods.'
  labels:
    kubernetes.io/cluster-service: "true"
    eks.amazonaws.com/component: pod-security-policy
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: eks:podsecuritypolicy:privileged
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: system:authenticated
```

2. Aplique el archivo YAML con el siguiente comando.

```
kubectl apply -f privileged-podsecuritypolicy.yaml
```

Preguntas frecuentes sobre la eliminación (PSP) de políticas de seguridad de pods

`PodSecurityPolicy` quedó en desuso en Kubernetes`1.21` y se ha eliminado en Kubernetes`1.25`. Si utiliza PodSecurityPolicy en el clúster, debe migrar a los Kubernetesestándares de seguridad de pod integrados(PSS) o a una solución de política como código antes de actualizar el clúster a

la versión **`1.25`** para evitar interrupciones en las cargas de trabajo. Seleccione cualquier pregunta frecuente para obtener más información.

¿Qué es un PSP?

PodSecurityPolicy es un controlador de admisión integrado que permite al administrador del clúster controlar los aspectos sensibles a la seguridad de la especificación del Pod. Si un Pod cumple con los requisitos de su PSP, el Pod se admite en el clúster como de costumbre. Si un Pod no cumple con los requisitos del PSP, el Pod se rechaza y no se puede ejecutar.

¿La eliminación de PSP es específica de Amazon EKS o se está eliminando en una Kubernetes anterior?

Se trata de un cambio previo en el proyecto Kubernetes y no de Amazon EKS. PSPquedó en desuso en Kubernetes `1.21` y se eliminó en Kubernetes `1.25`. La comunidad de Kubernetes identificó graves problemas de usabilidad con PSP. Estas incluían la concesión accidental de permisos más amplios de lo previsto y la dificultad de inspeccionar los que PSPs se aplican en una situación determinada. Estos problemas no podían solucionarse sin realizar cambios importantes. Esta es la razón principal por la que la comunidad de Kubernetes decidió eliminar PSP.

¿Cómo puedo comprobar si lo estoy utilizando PSPs en mis clústeres de Amazon EKS?

Para comprobar si está utilizando PSPs en su clúster, puede ejecutar el siguiente comando:

```
kubectl get psp
```

Para ver los Pods que los PSPs impactan en el clúster, ejecute el siguiente comando. Este comando genera el nombre del Pod, el espacio de nombres y los PSPs:

```
kubectl get pod -A -o jsonpath='{range.items[?(@.metadata.annotations.kubernetes
\.io/psp)]}{.metadata.name}{"\t"}{.metadata.namespace}{"\t"}
{.metadata.annotations.kubernetes\.io/psp}{"\n"}'
```

Si utilizo PSPs en mi clúster de Amazon EKS, ¿qué puedo hacer?

Antes de actualizar el clúster a `1.25`, debe migrar PSPs a una de estas alternativas:

- Kubernetes PSS.

- Soluciones de políticas como código del entorno de Kubernetes.

En respuesta a la obsolescencia de PSP y a la necesidad continua de controlar la seguridad de los Pod desde el principio, la comunidad de Kubernetes creó una solución integrada con los (PSS) y la admisión de seguridad del pod (PSA). El webhook de PSA implementa los controles que se definen en la PSS.

Puede revisar las prácticas recomendadas para migrar PSPs a la PSS integrada en la Guía de mejores prácticas de EKS. También le recomendamos que consulte nuestro blog sobre la implementación de estándares de seguridad de pods en Amazon EKS. Las referencias adicionales incluyen migrar de PodSecurityPolicy al controlador de admisión PodSecurity integrado y asignar PodSecurityPolicies a los estándares de seguridad de pods.

Las soluciones de políticas como código proporcionan barreras de protección para guiar a los usuarios del clúster y evitan los comportamientos no deseados mediante controles automatizados prescritos. Las soluciones de política como código suelen utilizar los controladores de admisión dinámica de Kubernetes para interceptar el flujo de solicitudes del servidor de API de Kubernetes mediante una llamada de webhook. Las soluciones de políticas como código mutan y validan las cargas de las solicitudes en función de las políticas escritas y almacenadas como código.

Hay varias soluciones de políticas como código de código abierto disponibles para Kubernetes. Para revisar las prácticas recomendadas para migrar PSPs a una solución de política como código, consulte la sección Política como código de la página de seguridad de pods en GitHub.

Veo una PSP llamada **`eks.privileged`** en mi clúster. ¿Qué es y qué puedo hacer al respecto?

Los clústeres de Amazon EKS con la versión `1.13` o posterior de Kubernetes tienen un PSP predeterminado denominado `eks.privileged`. Esta política se creó en `1.24` y en clústeres anteriores. No se usa en `1.25` ni en clústeres posteriores. Amazon EKS migra automáticamente esta PSP a un sistema de cumplimiento basado en PSS. No tiene que hacer nada.

¿Amazon EKS realizará cambios en las PSPs presentes en mi clúster existente cuando actualice mi clúster a la versión **`1.25`**?

No. Además de `eks.privileged`, que es un PSP creado por Amazon EKS, no se realizan cambios en otros PSPs del clúster cuando se actualiza a `1.25`.

¿Impedirá Amazon EKS una actualización de clúster a la versión **1.25** si aún no he migrado de PSP?

No. Amazon EKS no impedirá que el clúster se actualice a la versión `1.25` si aún no ha realizado la migración de PSP.

¿Qué sucede si me olvido de migrar PSPs a PSS/PSA o a una solución de política como código antes de actualizar mi clúster a la versión **1.25**? ¿Puedo realizar la migración después de actualizar mi clúster?

Cuando un clúster que contiene un PSP se actualiza a Kubernetes versión `1.25`, el servidor de API no reconoce el recurso de PSP en `1.25`. Esto podría provocar que los Pods obtengan alcances de seguridad incorrectos. Para obtener una lista completa de las implicaciones, consulte Migrar de PodSecurityPolicy al controlador de admisión de PodSecurity integrado.

¿Cómo afecta este cambio a la seguridad de los pods para las cargas de trabajo de Windows?

No esperamos ningún impacto específico en las cargas de trabajo de Windows. PodSecurityContext tiene un campo llamado `windowsOptions` en la API de `PodSpec v1` para los Pods de Windows. Esto usa PSS en Kubernetes `1.25`. Para obtener más información y las prácticas recomendadas sobre la aplicación de PSS para las cargas de trabajo de Windows, consulte la Guía de prácticas recomendadas de EKS y la documentación de Kubernetes.

Uso de secretos de AWS Secrets Manager con Kubernetes

A fin de mostrar secretos de Secrets Manager y parámetros de Parameter Store como archivos montados en Pods de Amazon EKS, puede utilizar el proveedor de secretos y configuración de AWS (ASCP) para el controlador de CSI del almacén de secretos de Kubernetes.

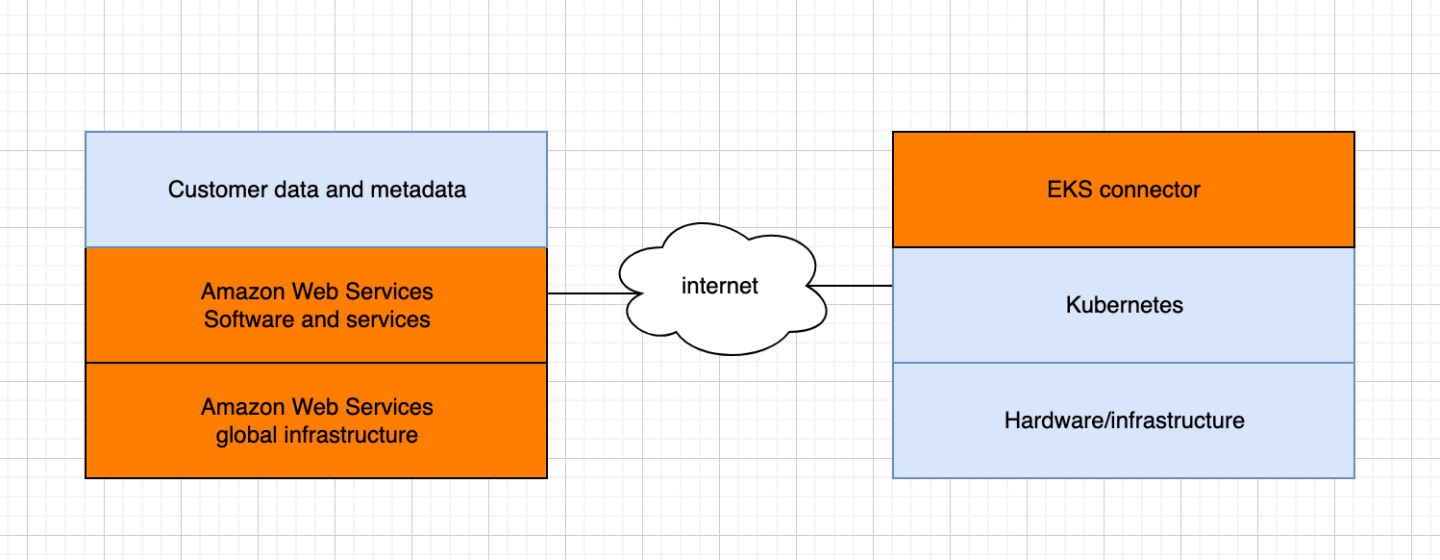
Con el ASCP, puede almacenar y administrar sus secretos en Secrets Manager y recuperarlos a través de sus cargas de trabajo que se ejecutan en Amazon EKS. Puede utilizar roles y políticas de IAM para limitar el acceso a sus secretos a Pods de Kubernetes específicos de un clúster. El ASCP recupera la identidad del Pod e intercambia la identidad por un rol de IAM. El ASCP asume el rol de IAM del Pod y, a continuación, puede recuperar secretos de Secrets Manager autorizados para ese rol.

Si utiliza la rotación automática de Secrets Manager para sus secretos, también puede utilizar la característica de conciliador de rotación del controlador de CSI del almacén de secretos a fin de asegurarse de que está recuperando el último secreto de Secrets Manager.

Para obtener más información, consulte Uso de secretos de Secrets Manager en Amazon EKS en la Guía del usuario de AWS Secrets Manager.

Consideraciones sobre Amazon EKS Connector

Amazon EKS Connector es un componente de código abierto que se ejecuta en el clúster de Kubernetes. Este clúster se puede ubicar fuera del entorno AWS. Esto crea consideraciones adicionales para las responsabilidades de seguridad. El siguiente diagrama ilustra esta configuración. Naranja representa responsabilidades de AWS y azul representa las responsabilidades del cliente:



En este tema se describen las diferencias en el modelo de responsabilidad si el clúster conectado está fuera de AWS.

Responsabilidades de AWS

- Mantenimiento, creación y entrega de Amazon EKS Connector, que es un componente de código abierto que se ejecuta en el clúster de Kubernetes de un cliente y se comunica con AWS.
- Mantenimiento de la seguridad de la comunicación del nivel de aplicaciones y del transporte entre el clúster de Kubernetes conectado y servicios de AWS.

Responsabilidades del cliente

- Seguridad específica del clúster de Kubernetes, específicamente en las siguientes líneas:
  - Los secretos de Kubernetes deben estar cifrados y protegidos correctamente.
  - Bloquee el acceso al espacio de nombres de `eks-connector`.
- Configuración de permisos de control de acceso basado en roles (RBAC) para administrar el acceso de entidades principales de IAM desde AWS. Para obtener más información, consulte Concesión de acceso a una entidad principal de IAM para ver recursos de Kubernetes en un clúster.
- Instalación y actualización de Amazon EKS Connector.
- Mantenimiento del hardware, el software y la infraestructura que admite el clúster de Kubernetes conectado.
- Seguridad de sus cuentas de AWS (por ejemplo, protegiendo sus credenciales de usuario raíz seguro).

Vea los recursos de Kubernetes

Puede ver los recursos de Kubernetes implementados en su clúster con la AWS Management Console. No puede ver los recursos de Kubernetes con la AWS CLI o `eksctl`. Para ver los recursos de Kubernetes mediante una herramienta de línea de comandos, utilice `kubectl`.

Requisito previo

Para ver la pestaña Recursos y la sección Nodos de la pestaña Computación del AWS Management Console, la entidad principal de IAM que utilice debe tener un IAM y permisos de Kubernetes específicos. Para obtener más información, consulte Permisos necesarios.

Para ver los recursos de Kubernetes con el AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. En la lista Clústeres, seleccione el clúster que contiene los recursos de Kubernetes que desea ver.
3. Seleccione la pestaña Recursos.
4. Seleccione un grupo de Tipo de recurso del que desea ver los recursos, como Cargas de trabajo. Aparece una lista de los tipos de recursos de ese grupo.
5. Seleccione un tipo de recurso, como Implementaciones, en el grupo Cargas de trabajo. Puede ver una descripción del tipo de recurso, un enlace a la documentación de Kubernetes para obtener más información sobre el tipo de recurso y una lista de los recursos de ese tipo que se implementan en el clúster. Si la lista está vacía, no hay recursos de ese tipo implementado en el clúster.
6. Seleccione un recurso para ver más información acerca de una instantánea. Pruebe los siguientes ejemplos:

   - Seleccione el grupo Cargas de trabajo, seleccione el tipo de recurso de Implementaciones y seleccione el recurso coredns. Al seleccionar un recurso, se encuentra en Vista estructurada, de forma predeterminada. Para algunos tipos de recursos, verá una sección de Pods en Vista estructurada. En esta sección se muestran los Pods administrados por la carga de trabajo. Puede seleccionar cualquier Pod de la lista para ver información acerca del Pod. No todos los tipos de recursos muestran información en Vista estructurada. Si selecciona Vista sin procesar en la esquina superior derecha de la página del recurso, verá la respuesta JSON completa de la API de Kubernetes para el recurso.

- Seleccione el grupo Clústeres y, a continuación, seleccione el tipo de recursos Nodos. Aparece una lista de todos los nodos del clúster. Los nodos pueden ser cualquier tipo de nodo de Amazon EKS. Esta es la misma lista que ve en la sección Nodos al seleccionar la pestaña Informática de su clúster. Seleccione un recurso de nodo de la lista. En Vista estructurada, también ve una sección de Pods. En esta sección le mostramos todos los Pods que se ejecutan en el nodo.

Permisos necesarios

Para ver la pestaña Recursos y la sección Nodos de la pestaña Computación del AWS Management Console, la entidad principal de IAM que utilice debe tener un IAM mínimo y permisos de Kubernetes específicos. Complete los siguientes pasos para asignar los permisos necesarios a las entidades principales de IAM.

1. Asegúrese de que los `eks:AccessKubernetesApi` y otros permisos de IAM necesarios para ver los recursos de Kubernetes estén asignados a la entidad principal de IAM que esté utilizando. Para obtener más información acerca de cómo editar los permisos para una entidad principal de IAM, consulte Control del acceso para las entidades principales de IAM en la Guía del usuario de IAM. Para obtener más información acerca de cómo editar los permisos de un rol, consulte Modificación de una política de permisos de rol (consola) en la Guía del usuario de IAM.

   En la siguiente política de ejemplo se incluyen los permisos necesarios para que una entidad principal vea los recursos de Kubernetes de todos los clústeres de su cuenta. Reemplace *111122223333* por su ID de cuenta de AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "eks:ListFargateProfiles",
                "eks:DescribeNodegroup",
                "eks:ListNodegroups",
                "eks:ListUpdates",
                "eks:AccessKubernetesApi",
                "eks:ListAddons",
                "eks:DescribeCluster",
                "eks:DescribeAddonVersions",
```

```
                "eks:ListClusters",
                "eks:ListIdentityProviderConfigs",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:GetParameter",
            "Resource": "arn:aws:ssm:*:111122223333:parameter/*"
        }
    ]
}
```

Para ver los nodos de los clústeres conectados, el rol de IAM del conector de Amazon EKS debería poder suplantar a la entidad principal en el clúster. Esto permite que Amazon EKS Connector asigne la entidad principal a un usuario de Kubernetes.

2. Cree un `rolebinding` o `clusterrolebinding` de Kubernetes vinculado a un `role` o `clusterrole` de Kubernetes que tenga los permisos necesarios para ver los recursos de Kubernetes. Para conocer más sobre roles y vinculaciones de roles de Kubernetes, consulte la Utilización de la autorización de RBAC en la documentación de Kubernetes. Puede aplicar uno de los siguientes manifiestos al clúster que crea un `role` y `rolebinding` o un `clusterrole` y `clusterrolebinding` con los permisos de Kubernetes necesarios:

   Ver recursos de Kubernetes en todos los espacios de nombres

   El nombre del grupo en el archivo es `eks-console-dashboard-full-access-group`. Aplique el manifiesto al clúster con el siguiente comando:

   ```
   kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-full-access.yaml
   ```

   Ver recursos de Kubernetes en un espacio de nombres específico

   El espacio de nombres de este archivo es `default`. El nombre del grupo en el archivo es `eks-console-dashboard-restricted-access-group`. Aplique el manifiesto al clúster con el siguiente comando:

```
kubectl apply -f https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-restricted-access.yaml
```

Si necesita cambiar el nombre del grupo de Kubernetes, el espacio de nombres, los permisos o cualquier otra configuración del archivo, descargue el archivo y edítelo antes de aplicarlo al clúster:

1. Descargue el archivo con uno de los siguientes comandos:

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-full-access.yaml
   ```

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/docs/eks-console-restricted-access.yaml
   ```

2. Edite el archivo según sea necesario.
3. Aplique el manifiesto al clúster con uno de los siguientes comandos:

   ```
   kubectl apply -f eks-console-full-access.yaml
   ```

   ```
   kubectl apply -f eks-console-restricted-access.yaml
   ```

3. Asigne la entidad principal de IAM al usuario de Kubernetes o al grupo en el `aws-auth ConfigMap`. Puede utilizar una herramienta como `eksctl` para actualizar el `ConfigMap` o puede actualizarlo manualmente editándolo.

> **Important**
>
> Recomendamos utilizar `eksctl`, u otra herramienta, para editar el `ConfigMap`. Para obtener información acerca de otras herramientas que puede utilizar, consulte Utilice herramientas para realizar cambios en el `aws-authConfigMap` en las guías de prácticas recomendadas de Amazon EKS. Un formato incorrecto de `aws-auth ConfigMap` puede provocar que pierda el acceso a su clúster.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

1. Vea las asignaciones actuales en la `ConfigMap`. Reemplace *`my-cluster`* por el nombre del clúster. Reemplace *`region-code`* por la Región de AWS en la que se encuentra el clúster.

   ```
   eksctl get iamidentitymapping --cluster my-cluster --region=region-code
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   ARN
                     USERNAME                                GROUPS
               ACCOUNT
   arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
   NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
    system:bootstrappers,system:nodes
   ```

2. Agregue una asignación para un rol. En este ejemplo se supone que ha adjuntado los permisos de IAM en el primer paso a un rol denominado *`my-console-viewer-role`*. Reemplace *`111122223333`* por su ID de cuenta.

   ```
   eksctl create iamidentitymapping \
       --cluster my-cluster \
       --region=region-code \
       --arn arn:aws:iam::111122223333:role/my-console-viewer-role \
       --group eks-console-dashboard-full-access-group \
       --no-duplicate-arns
   ```

> **Important**
>
> El ARN de rol no puede incluir una ruta como `role/my-team/developers/my-role`. El formato del ARN debe ser `arn:aws:iam::111122223333:role/my-role`. En este ejemplo, se necesita eliminar `my-team/developers/`.

Un ejemplo de salida sería el siguiente.

```
[...]
2022-05-09 14:51:20 [#]  adding identity "arn:aws:iam::111122223333:role/my-console-viewer-role" to auth ConfigMap
```

3. Agregue una asignación para un usuario. Según las prácticas recomendadas de IAM, se recomienda conceder permisos a los roles en lugar de a los usuarios. En este ejemplo se supone que ha adjuntado los permisos de IAM en el primer paso a un usuario denominado *my-user*. Reemplace *111122223333* por su ID de cuenta.

```
eksctl create iamidentitymapping \
    --cluster my-cluster \
    --region=region-code \
    --arn arn:aws:iam::111122223333:user/my-user \
    --group eks-console-dashboard-restricted-access-group \
    --no-duplicate-arns
```

Un ejemplo de salida sería el siguiente.

```
[...]
2022-05-09 14:53:48 [#]  adding identity "arn:aws:iam::111122223333:user/my-user" to auth ConfigMap
```

4. Vea las asignaciones en el `ConfigMap` de nuevo.

```
eksctl get iamidentitymapping --cluster my-cluster --region=region-code
```

Un ejemplo de salida sería el siguiente.

```
ARN
                    USERNAME                                      GROUPS
                      ACCOUNT
arn:aws:iam::111122223333:role/eksctl-my-cluster-my-nodegroup-
NodeInstanceRole-1XLS7754U3ZPA    system:node:{{EC2PrivateDNSName}}
 system:bootstrappers,system:nodes
arn:aws:iam::111122223333:role/my-console-viewer-role
                                                                  eks-console-
dashboard-full-access-group
arn:aws:iam::111122223333:user/my-user
                                                                  eks-console-
dashboard-restricted-access-group
```

Edit ConfigMap manually

Para obtener más información sobre cómo agregar usuarios al `ConfigMap` de `aws-auth`, consulte Agregar las entidades principales de IAM al clúster de Amazon EKS.

1. Abra el `ConfigMap` de `aws-auth` para editar.

```
kubectl edit -n kube-system configmap/aws-auth
```

2. Agregue las asignaciones a la `aws-auth` `ConfigMap`, pero no reemplace ninguna de las asignaciones existentes. En el siguiente ejemplo se agregan asignaciones entre entidades principales de IAM con permisos agregados en el primer paso y los grupos de Kubernetes creados en el paso anterior:

   - El rol *my-console-viewer-role* y la `eks-console-dashboard-full-access-group`.
   - El usuario *my-user* y el `eks-console-dashboard-restricted-access-group`.

   En estos ejemplos se supone que ha adjuntado los permisos de IAM en el primer paso a un rol denominado *my-console-viewer-role* y un usuario llamado *my-user*. Reemplace *111122223333* por su ID de cuenta de AWS.

```
apiVersion: v1
data:
mapRoles: |
  - groups:
    - eks-console-dashboard-full-access-group
```

```
    rolearn: arn:aws:iam::111122223333:role/my-console-viewer-role
    username: my-console-viewer-role
mapUsers: |
  - groups:
    - eks-console-dashboard-restricted-access-group
    userarn: arn:aws:iam::111122223333:user/my-user
    username: my-user
```

> **Important**
>
> El ARN de rol no puede incluir una ruta como `role/my-team/developers/my-console-viewer-role`. El formato del ARN debe ser `arn:aws:iam::111122223333:role/my-console-viewer-role`. En este ejemplo, se debe eliminar `my-team/developers/`.

3. Guarde el archivo y salga del editor de texto.

Observabilidad en Amazon EKS

Puede observar sus datos en Amazon EKS utilizando muchas herramientas de monitoreo o registro disponibles. Los datos de registro de Amazon EKS se pueden transmitir a Servicios de AWS o a herramientas de socios para análisis de datos. Hay muchos servicios disponibles en el AWS Management Console que proporcionan datos para solucionar problemas de Amazon EKS. También puede utilizar una solución AWS de código abierto compatible para monitorear la infraestructura de Amazon EKS.

Puede ver el estado y los detalles del clúster seleccionando el nombre del clúster después de seleccionar Clústeres en el panel de navegación izquierdo de la consola de Amazon EKS. Para ver detalles acerca de los recursos de Kubernetes existentes implementados en el clúster, consulte Vea los recursos de Kubernetes.

El monitoreo es una parte importante a la hora de mantener la fiabilidad, la disponibilidad y el rendimiento de Amazon EKS y de las soluciones de AWS. Le recomendamos que recopile datos de monitorización de todas las partes de su solución de AWS. De esta forma, puede depurar con mayor facilidad un error que se produce en distintas partes del error que se produce en distintos puntos. Antes de comenzar a supervisar Amazon EKS, asegúrese de que su plan de monitorización responde a las siguientes preguntas.

- ¿Cuáles son los objetivos? ¿Necesita notificaciones en tiempo real si los clústeres escalan drásticamente?
- ¿Qué recursos hay que observar?
- ¿Con qué frecuencia necesita observar estos recursos? ¿Quiere su empresa responder rápidamente a los riesgos?
- ¿Qué herramientas pretende utilizar? Si ya ejecuta AWS Fargate como parte de su lanzamiento, puede usar el enrutador de registros integrado.
- ¿Quién pretende realizar las tareas de monitorización?
- ¿A quién quiere que se envíen notificaciones cuando algo sale mal?

Registro y monitoreo en Amazon EKS

Amazon EKS proporciona herramientas integradas para registrar y monitorear. El registro del plano de control registra todas las llamadas de API a los clústeres, la información de auditoría que captura

qué usuarios realizaron qué acciones en los clústeres, así como la información basada en roles. Para obtener más información, consulte Registro y monitorización en Amazon EKS en la Guía prescriptiva de AWS.

El registro del plano de control de Amazon EKS proporciona registros de auditoría y diagnóstico directamente desde el plano de control de Amazon EKS a CloudWatch Logs en su cuenta. Estos registros hacen que le resulte más fácil asegura y ejecutar los clústeres. Puede seleccionar los tipos de registro exactos que necesita. Los registros se envían como secuencias de registro a un grupo para cada clúster de Amazon EKS en CloudWatch. Para obtener más información, consulte Registro de plano de control de Amazon EKS.

> **Note**
>
> Al verificar los registros de autenticador de Amazon EKS en Amazon CloudWatch, se muestran las entradas que contienen texto similar al siguiente texto de ejemplo.
>
> ```
> level=info msg="mapping IAM role" groups="[]"
>  role="arn:aws:iam::111122223333:role/XXXXXXXXXXXXXXXXXX-
> NodeManagerRole-XXXXXXXX" username="eks:node-manager"
> ```
>
> Se esperan entradas que contengan este texto. El `username` es una función del servicio interna de Amazon EKS que realiza operaciones específicas para grupos de nodos administrados y Fargate.
> Para un registro de bajo nivel y personalizable, Registros de Kubernetes está disponible.

Amazon EKS se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un servicio de AWS en Amazon EKS. CloudTrail captura todas las llamadas a la API para Amazon EKS como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Amazon EKS y las llamadas desde el código a las operaciones de la API de Amazon EKS. Para obtener más información, consulte Registro de llamadas a la API de Amazon EKS con AWS CloudTrail.

El servidor de la API de Kubernetes expone una serie de métricas que son útiles a efectos del monitoreo y el análisis. Para obtener más información, consulte Métricas de Prometheus.

Para configurar Fluent Bit de los registros Amazon CloudWatch personalizados, consulte Configuración de Fluent Bit en la Guía del usuario de Amazon CloudWatch.

Herramientas de registro y monitoreo en Amazon EKS

Amazon Web Services proporciona varias herramientas que puede utilizar para monitorear Amazon EKS. Puede configurar algunas herramientas para configurar la monitorización automática, pero algunas requieren llamadas manuales. Le recomendamos que automatice las tareas de monitorización de la misma manera que lo permitan su entorno y el conjunto de herramientas existente.

Herramientas de registro

Áreas	Herramienta	Descripción	Configuración
Aplicaciones	Información de contenedores de Amazon CloudWatch	Recopila, agrega y resume métricas y registros de las aplicaciones y microservicios en contenedores.	Procedimiento de configuración
Plano de control	AWS CloudTrail	Registra las llamadas a la API de un usuario, rol o servicio.	Procedimiento de configuración
Múltiples áreas para instancias de AWS Fargate	Enrutador de registro de AWS Fargate	Para instancias de AWS Fargate, transmite registros a servicios de AWS o herramientas de socios. Usa AWS para Fluent Bit. Los registros se pueden	Procedimiento de configuración

Áreas	Herramienta	Descripción	Configuración
		transmitir a otros Servicios de AWS o herramientas de socios.	

Herramientas de monitorización

Áreas	Herramienta	Descripción	Configuración
Aplicaciones	Información de contenedores de CloudWatch	CloudWatch Container Insights recopila, agrega y resume métricas y registros de las aplicaciones y microservicios en contenedores.	Procedimiento de configuración
Aplicaciones	AWS Distro para OpenTelemetry (ADOT)	Recopila y envía métricas correlacionadas, datos de seguimiento y metadatos a servicios de monitoreo de AWS o socios. Se puede configurar a través de Información de	Procedimiento de configuración

Áreas	Herramienta	Descripción	Configuración
		contenedores de CloudWatch.	
Aplicaciones	Amazon DevOps Guru	Detecta la disponibilidad y el rendimien to operacional a nivel de nodo.	Procedimiento de configuración
Aplicaciones	AWS X-Ray	Recibe datos de seguimiento de su aplicació n. Estos datos de seguimiento incluyen solicitud es entrantes y salientes y metadatos sobre las solicitudes. Para Amazon EKS, la implement ación requiere el complemento OpenTelemetry.	Procedimiento de configuración

Áreas	Herramienta	Descripción	Configuración
Aplicaciones	Amazon CloudWatch Observability Operator	El Amazon CloudWatch Observability Operator recopila métricas, registros y datos de rastreo. Los envía a Amazon CloudWatch y AWS X-Ray.	Procedimiento de configuración
Plano de control	Prometheus	Las tasas de ingesta, almacenamiento de archivos y análisis de datos de CloudWatch Logs se aplican a los registros del plano de control habilitados.	Procedimiento de configuración

Métricas de Prometheus

Prometheus es una base de datos de serie temporal y de monitoreo que recopila puntos de conexión. Ofrece la posibilidad de consultar, agregar y almacenar los datos recopilados. También se puede utilizar para la generación de alertas y su agregación. En este tema se explica cómo configurar Prometheus como una opción gestionada o de código abierto. La monitorización de las métricas del plano de control de Amazon EKS es un caso de uso común.

Amazon Managed Service for Prometheus es un servicio de supervisión y alertas compatible con Prometheus que facilita el monitoreo de las aplicaciones y la infraestructura en contenedores

a escala. Es un servicio totalmente administrado que escala automáticamente la ingesta, el almacenamiento, las consultas y las alertas de sus métricas. También se integra con servicios de seguridad de AWS para permitir un acceso rápido y seguro a sus datos. Puede utilizar el lenguaje de consulta ProMQL de código abierto para consultar sus métricas y crear alertas sobre ellas.

Para obtener más información sobre cómo utilizar las métricas de Prometheus después de activarlas, consulte la Guía del usuario de Amazon Managed Service for Prometheus.

Active las métricas de Prometheus al crear un clúster

> **Important**
>
> Los recursos de Amazon Managed Service para Prometheus están fuera del ciclo de vida del clúster y deben mantenerse por fuera del clúster. Al eliminar el clúster, asegúrese de eliminar, también, cualquier raspador para reducir los costes aplicables. Para más información, consulte Búsqueda y eliminación de rapsadores en la Guía de usuario de Amazon Managed Service para Prometheus.

Al crear un clúster nuevo, puede activar la opción de enviar métricas a Prometheus. En la AWS Management Console, esta opción se encuentra en el paso Configurar la observabilidad, que consiste en crear un clúster nuevo. Para obtener más información, consulte Creación de un clúster de Amazon EKS.

Prometheus descubre y recopila las métricas de su clúster mediante un modelo basado en la extracción denominado raspado. Los raspadores están configurados para recopilar datos de la infraestructura del clúster y de las aplicaciones en contenedores.

Al activar la opción de enviar métricas de Prometheus, Amazon Managed Service for Prometheus proporciona un rastreador sin agentes totalmente gestionado. Use las siguientes opciones de configuración avanzada para personalizar el raspador predeterminado según sea necesario.

Alias de raspador

(Opcional) Escriba un alias único para el raspador.

Destino

Elija un espacio de trabajo de Amazon Managed Service para Prometheus. Un espacio de trabajo es un espacio lógico dedicado al almacenamiento y la consulta de las métricas de Prometheus.

Con este espacio de trabajo, podrá ver las métricas de Prometheus de las cuentas que tienen acceso a él. La opción Crear un nuevo espacio de trabajo indica a Amazon EKS que cree un espacio de trabajo en su nombre con el alias de espacio de trabajo que usted proporcione. Con la opción Seleccionar un espacio de trabajo existente, puede seleccionar un espacio de trabajo existente de una lista desplegable. Para obtener más información sobre los espacios de trabajo, consulte Gestión de espacios de trabajo en la Guía del usuario de Amazon Managed Service for Prometheus.

Acceso a los servicios

En esta sección se resumen los permisos que se conceden al enviar métricas de Prometheus:

- Permiso para que Amazon Managed Service for Prometheus describa el clúster de Amazon EKS
- Permiso para la escritura remota en el espacio de trabajo de Amazon Managed Prometheus

Si el `AmazonManagedScraperRole` ya existe, el raspador lo usa. Elija el enlace de `AmazonManagedScraperRole` para ver los detalles del permiso. Si el `AmazonManagedScraperRole` aún no existe, seleccione el enlace Ver detalles del permiso para ver los permisos específicos que está concediendo mediante el envío de las métricas de Prometheus.

Subredes

Vea las subredes que heredará el raspador. Si necesita cambiarlas, vuelva al paso de creación del clúster de Especificar red.

Grupos de seguridad

Vea los grupos de seguridad que heredará el raspador. Si necesita cambiarlas, vuelva al paso de creación del clúster de Especificar red.

Configuración del raspador

Modifique la configuración del raspador en formato YAML según sea necesario. Para ello, utilice el formulario o cargue un archivo YAML de reemplazo. Para obtener más información, consulte Configuración del raspador en la Guía del usuario de Amazon Managed Service for Prometheus.

Amazon Managed Service for Prometheus hace referencia al raspador sin agente que se crea junto con el clúster como recopilador gestionado de AWS. Para obtener más información sobre los recopiladores gestionados por AWS, consulte los recopiladores gestionados por AWS en la Guía del usuario de Amazon Managed Service for Prometheus.

> **Important**
>
> Debe configurar sus `aws-auth ConfigMap` para conceder al rastreador permisos dentro del clúster. Para obtener más información, consulte Configuración del clúster de Amazon EKS en la Guía del usuario de Amazon Managed Service for Prometheus.

Visualización de los detalles del raspador de Prometheus

Tras crear un clúster con la opción de métricas de Prometheus activada, podrá ver los detalles del raspador de Prometheus. Cuando vea su clúster en la AWS Management Console, seleccione la pestaña Observabilidad. En una tabla se muestra una lista de los raspadores del clúster, que incluye información como el ID, el alias, el estado y la fecha de creación del raspador.

Para ver más detalles sobre el raspador, elija un enlace de ID del raspador. Por ejemplo, puede ver la configuración del raspador, el nombre de recurso de Amazon (ARN), la URL de escritura remota y la información de la red. Puedes usar el identificador del raspador como entrada en Amazon Managed Service para operaciones de la API de Prometheus, como `DescribeScraper` y `DeleteScraper`. También puede utilizar la API para crear más raspadores.

Para obtener más información sobre el uso de la API de Prometheus, consulte la Referencia de API de Amazon Managed Service for Prometheus.

Implementación de Prometheus mediante Helm

Como alternativa, puede realizar la implementación de Prometheus en su clúster con la versión 3 de Helm. Si ya ha instalado Helm, puede comprobar su versión con el comando `helm version`. Helme s un administrador de paquetes para clústeres Kubernetes. Para obtener más información sobre Helm y sobre cómo instalarlo, consulte Utilizar Helm con Amazon EKS.

Después de configurar Helm para su clúster de Amazon EKS, puede utilizarlo para implementar Prometheus con los pasos que se describen a continuación.

Para implementar Prometheus usando Helm

1. Cree un espacio de nombres de Prometheus.

```
kubectl create namespace prometheus
```

2. Agregue el repositorio de gráficos de `prometheus-community`.

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

3. Implemente Prometheus.

```
helm upgrade -i prometheus prometheus-community/prometheus \
    --namespace prometheus \
    --set alertmanager.persistence.storageClass="gp2" \
    --set server.persistentVolume.storageClass="gp2"
```

> **Note**
>
> Si recibe el error `Error: failed to download "stable/prometheus" (hint: running `helm repo update` may help)` al ejecutar este comando, ejecute `helm repo update prometheus-community` y, a continuación, vuelva a ejecutar el comando del Paso 2.
>
> Si recibe el error `Error: rendered manifests contain a resource that already exists`, ejecute `helm uninstall` *`your-release-name`* `-n` *`namespace`* y, a continuación, vuelva a ejecutar el comando del Paso 3.

4. Compruebe que todos los Pods en el espacio de nombres de `prometheus` se encuentran en estado `READY`.

```
kubectl get pods -n prometheus
```

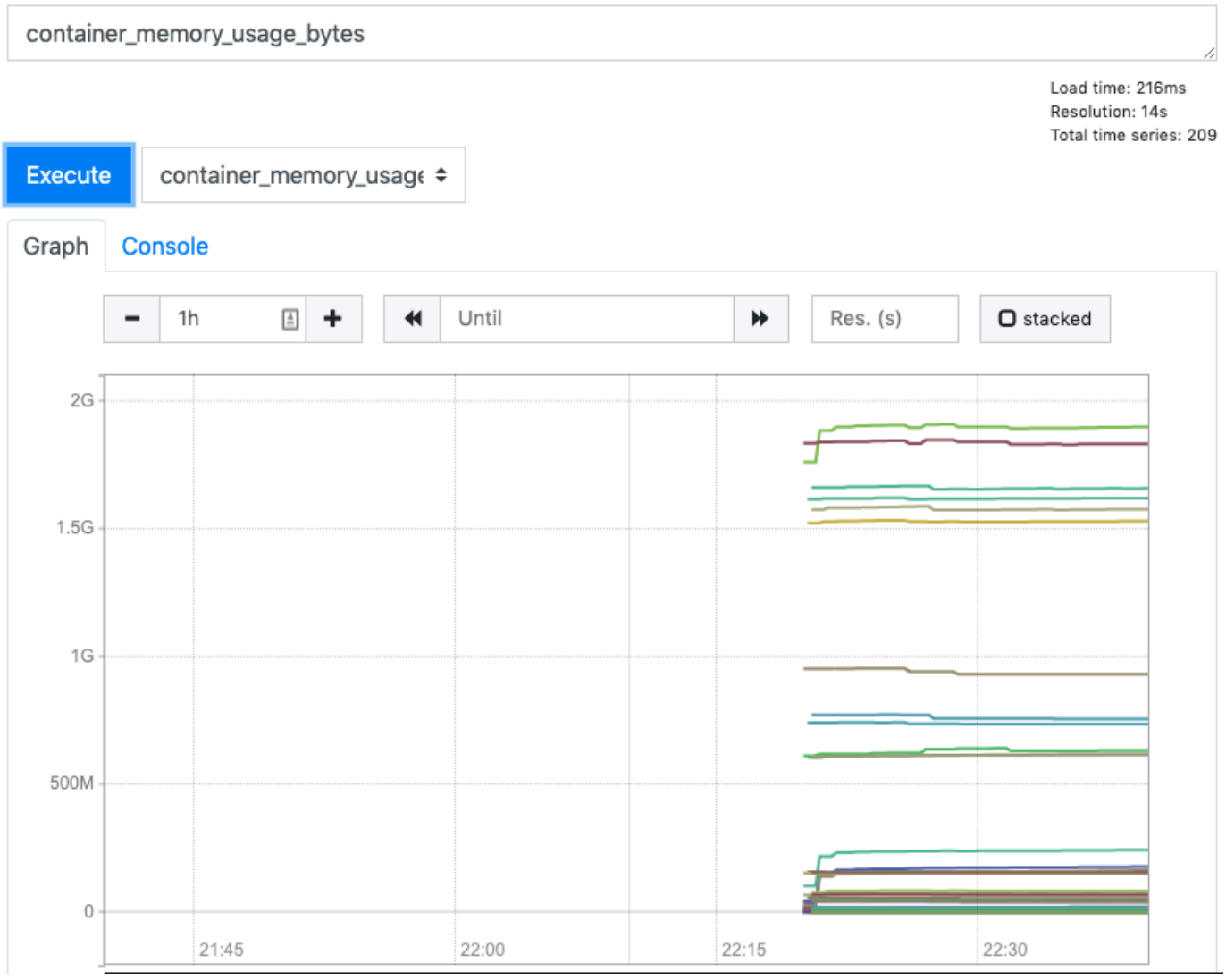
Un ejemplo de salida sería el siguiente.

```
NAME                                             READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-59b4c8c744-r7bgp         1/2     Running   0          48s
prometheus-kube-state-metrics-7cfd87cf99-jkz2f   1/1     Running   0          48s
prometheus-node-exporter-jcjqz                   1/1     Running   0          48s
prometheus-node-exporter-jxv2h                   1/1     Running   0          48s
prometheus-node-exporter-vbdks                   1/1     Running   0          48s
prometheus-pushgateway-76c444b68c-82tnw          1/1     Running   0          48s
prometheus-server-775957f748-mmht9               1/2     Running   0          48s
```

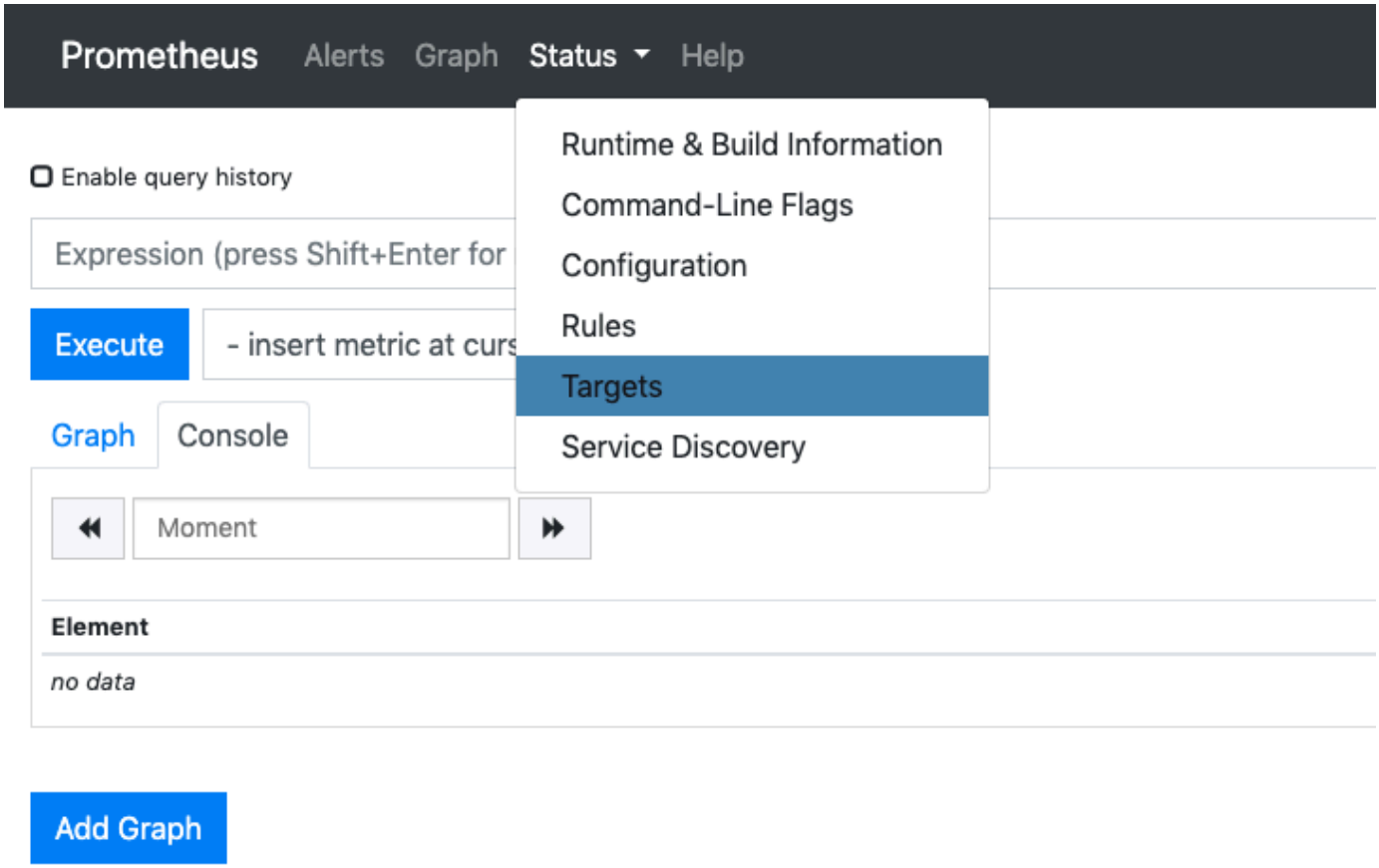
5. Utilice `kubectl` para el enrutamiento del puerto de la consola de Prometheus a su equipo local.

```
kubectl --namespace=prometheus port-forward deploy/prometheus-server 9090
```

6. Apunte un navegador web a `http://localhost:9090` para ver la consola de Prometheus.
7. Elija una métrica del menú - insert metric at cursor (- insertar métrica en el cursor) y elija Execute (Ejecutar). Elija la pestaña Graph (Gráfico) para mostrar la métrica con el paso del tiempo. La siguiente imagen muestra `container_memory_usage_bytes` a lo largo del tiempo.



8. En la barra de navegación superior, elija Status (Estado) y Targets (Destinos).



Se muestran todos los puntos de conexión de Kubernetes que están conectados a Prometheus mediante detección de servicios.

Visualización de las métricas sin procesar del plano de control

Como alternativa a la implementación de Prometheus, el servidor de API de Kubernetes expone una serie de métricas que se representan en un formato Prometheus. Estas métricas son útiles para el monitoreo y el análisis. Se exponen internamente a través de un punto de conexión de métricas que hace referencia a las API HTTP `/metrics`. Al igual que otros puntos de conexión, este punto de conexión se expone en el plano de control de Amazon EKS. Este punto de conexión es útil principalmente para analizar una métrica específica. Para analizar las métricas a lo largo del tiempo, recomendamos implementar Prometheus.

Para ver el resultado de métricas sin procesar, utilice `kubectl` con la marca `--raw`. Este comando le permite transferir cualquier ruta HTTP y devuelve la respuesta sin procesar.

```
kubectl get --raw /metrics
```

Un ejemplo de salida sería el siguiente.

```
[...]
# HELP rest_client_requests_total Number of HTTP requests, partitioned by status code,
 method, and host.
# TYPE rest_client_requests_total counter
rest_client_requests_total{code="200",host="127.0.0.1:21362",method="POST"} 4994
rest_client_requests_total{code="200",host="127.0.0.1:443",method="DELETE"} 1
rest_client_requests_total{code="200",host="127.0.0.1:443",method="GET"} 1.326086e+06
rest_client_requests_total{code="200",host="127.0.0.1:443",method="PUT"} 862173
rest_client_requests_total{code="404",host="127.0.0.1:443",method="GET"} 2
rest_client_requests_total{code="409",host="127.0.0.1:443",method="POST"} 3
rest_client_requests_total{code="409",host="127.0.0.1:443",method="PUT"} 8
# HELP ssh_tunnel_open_count Counter of ssh tunnel total open attempts
# TYPE ssh_tunnel_open_count counter
ssh_tunnel_open_count 0
# HELP ssh_tunnel_open_fail_count Counter of ssh tunnel failed open attempts
# TYPE ssh_tunnel_open_fail_count counter
ssh_tunnel_open_fail_count 0
```

Este resultado sin procesar devuelve literalmente lo que el servidor de API expone. Las diferentes métricas se muestran por línea, y cada línea incluye un nombre de métrica, etiquetas y un valor.

```
metric_name{"tag"="value"[,...]}
            value
```

Compatibilidad del componente de Amazon EKS para Amazon CloudWatch

Amazon CloudWatch Observability recopila registros, métricas y datos de rastreo en tiempo real. Los envía a Amazon CloudWatch y AWS X-Ray. Puede instalar este complemento para habilitar tanto CloudWatch Application Signals como Container Insights de CloudWatch con una observabilidad mejorada para Amazon EKS. Esto le ayuda a monitorear el estado y el rendimiento de su infraestructura y aplicaciones en contenedores. El Amazon CloudWatch Observability Operator está diseñado para instalar y configurar los componentes necesarios.

Amazon EKS es compatible con Amazon CloudWatch Observability Operator como un complemento de Amazon EKS. El complemento Container Insights también admite Linux en nodos de trabajo de Windows en el clúster. Para activar Container Insights en Windows, la versión del complemento

de Amazon EKS debe ser `1.5.0` o superior. Actualmente, CloudWatch Application Signals no es compatible con Amazon EKS en Windows.

En los temas siguientes se describe cómo comenzar a utilizar Amazon CloudWatch Observability Operator para el clúster de Amazon EKS.

- Para obtener instrucciones sobre la instalación de este complemento, consulte Install the CloudWatch agent by using the CloudWatch Observability Amazon EKS add-ons en la Guía del usuario de Amazon CloudWatch.
- Para obtener más información sobre CloudWatch Application Signals, consulte Application Signals.
- Para obtener más información sobre Container Insights, consulte Container Insights en la Guía del usuario de Amazon CloudWatch.

Registro de plano de control de Amazon EKS

El registro del plano de control de Amazon EKS proporciona registros de auditoría y diagnóstico directamente desde el plano de control de Amazon EKS a CloudWatch Logs en su cuenta. Estos registros hacen que le resulte más fácil asegura y ejecutar los clústeres. Puede seleccionar los tipos de registro exactos que necesita. Los registros se envían como secuencias de registro a un grupo para cada clúster de Amazon EKS en CloudWatch. Para obtener más información, consulte Registros de Amazon CloudWatch.

Para comenzar a utilizar el registro del plano de control de Amazon EKS, elija los tipos de registro que desea habilitar para los clústeres nuevos o existentes de Amazon EKS. Puede habilitar o desactivar cada tipo de registro en función del clúster por medio de la AWS Management Console, la AWS CLI (versión `1.16.139` o posterior) o la API de Amazon EKS. Cuando se encuentran habilitados, los registros se envían automáticamente desde el clúster de Amazon EKS a CloudWatch Logs en la misma cuenta.

Cuando se utiliza el registro del plano de control de Amazon EKS, se facturan los precios de Amazon EKS estándar para cada clúster que ejecuta. Se cobra la ingesta de datos de CloudWatch Logs y los costos de almacenamiento estándar para cualquier registro enviado a CloudWatch Logs desde sus clústeres. También se cobran los recursos de AWS, como las instancias de Amazon EC2 o los volúmenes de Amazon EBS que aprovisiona como parte de su clúster.

Están disponibles los siguientes tipos de registro de plano de control de clúster. Cada tipo de registro se corresponde con un componente del plano de control de Kubernetes. Para obtener

más información acerca de estos componentes, consulte los componentes de Kubernetes en la documentación de Kubernetes.

Servidor de la API (**`api`**)

El servidor de la API del clúster es el componente del plano de control que expone la API de Kubernetes. Si habilita los registros del servidor de la API al lanzar el clúster o poco después, estos registros incluyen los indicadores del servidor de la API que se usaron para iniciar el servidor de la API. Para obtener más información, consulte `kube-apiserver` y la política de auditoría en la documentación de Kubernetes.

Auditoría (**`audit`**)

Los registros de auditoría de Kubernetes ofrecen un registro de los usuarios individuales, administradores o componentes del sistema que han afectado el clúster. Para obtener más información, consulte la sección de auditorías en la documentación de Kubernetes.

Autenticador (**`authenticator`**)

Los registros del autenticador son exclusivos de Amazon EKS. Estos registros representan el componente del plano de control que Amazon EKS utiliza para la autenticación de control de acceso basado en rol (RBAC) de Kubernetes mediante credenciales de IAM. Para obtener más información, consulte Administración de clústeres.

Administrador de controladores (**`controllerManager`**)

El administrador de controladores administra los bucles de control principal que se envían con Kubernetes. Para obtener más información, consulte kube-controller-manager en la documentación de Kubernetes.

Programador (**`scheduler`**)

El componente programador administra cuándo y dónde ejecutar Pods en su clúster. Para obtener más información, consulte kube-scheduler en la documentación de Kubernetes.

Habilitar y deshabilitar registros de plano de control

De forma predeterminada, los registros del plano de control del clúster no se envían a CloudWatch Logs. Debe habilitar cada tipo de registro de manera individual a fin de enviar registros para el clúster. Las tasas de ingesta, almacenamiento de archivos y análisis de datos de CloudWatch Logs se aplican a los registros del plano de control habilitados. Para obtener más información, consulte los precios de CloudWatch.

Para actualizar la configuración de registro del plano de control, Amazon EKS requiere hasta cinco direcciones IP disponibles en cada subred. Al habilitar un tipo de registro, los registros se envían con un nivel de detalle de registro de 2.

AWS Management Console

Para habilitar o deshabilitar los registros de plano de control con la AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Elija el nombre del clúster para mostrar la información del clúster.
3. Seleccione la pestaña Observabilidad.
4. En la sección Registro del plano de control, elija Administrar registro.
5. Para cada tipo de registro individual, elija si el tipo de registro debe estar habilitado o deshabilitado. De forma predeterminada, cada tipo de registro está desactivado.
6. Elija Save changes (Guardar cambios) para terminar.

AWS CLI

Para habilitar o deshabilitar los registros de plano de control con la AWS CLI

1. Consulte su versión de AWS CLI con el siguiente comando.

```
aws --version
```

   Si su versión de AWS CLI es anterior a `1.16.139`, primero debe actualizar a la última versión. Para instalar o actualizar la AWS CLI, consulte Instalación de AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface.
2. Actualice la configuración de exportación del registro de plano de control del clúster con el siguiente comando de la AWS CLI. Reemplace *`my-cluster`* con el nombre del clúster y especifique los valores de acceso de punto de conexión deseados.

> **Note**
>
> El siguiente comando envía todos los tipos de registros disponibles a CloudWatch Logs.

```
aws eks update-cluster-config \
    --region region-code \
    --name my-cluster \
    --logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Un ejemplo de salida sería el siguiente.

```
{
    "update": {
        "id": "883405c8-65c6-4758-8cee-2a7c1340a6d9",
        "status": "InProgress",
        "type": "LoggingUpdate",
        "params": [
            {
                "type": "ClusterLogging",
                "value": "{\"clusterLogging\":[{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}]}"
            }
        ],
        "createdAt": 1553271814.684,
        "errors": []
    }
}
```

3. Monitoree el estado de la actualización de la configuración del registro con el siguiente comando, utilizando el nombre del clúster y el ID de actualización devueltos por el comando anterior. Su actualización se habrá completado cuando el estado mostrado sea `Successful`.

```
aws eks describe-update \
    --region region-code\
    --name my-cluster \
    --update-id 883405c8-65c6-4758-8cee-2a7c1340a6d9
```

Un ejemplo de salida sería el siguiente.

```
{
    "update": {
```

```
        "id": "883405c8-65c6-4758-8cee-2a7c1340a6d9",
        "status": "Successful",
        "type": "LoggingUpdate",
        "params": [
            {
                "type": "ClusterLogging",
                "value": "{\"clusterLogging\":[{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}]}"
            }
        ],
        "createdAt": 1553271814.684,
        "errors": []
    }
}
```

Visualización de registros de plano de control de clúster

Una vez que haya habilitado cualquiera de los tipos de registro del plano de control para el clúster de Amazon EKS, puede verlos en la consola de CloudWatch.

Para obtener más información sobre la visualización, el análisis y la administración de registros en CloudWatch, consulte la Guía del usuario de Amazon CloudWatch Logs.

Para ver los registros del plano de control del clúster en la consola de CloudWatch

1. Abra la consola de CloudWatch. Este enlace abre la consola y muestra los grupos de registro disponibles actualmente y los filtra con el prefijo `/aws/eks`.
2. Elija el clúster para el que desee ver registros. El formato del nombre del grupo de registros es `/aws/eks/my-cluster/cluster`.
3. Elija la secuencia de registro que desea ver. En la siguiente lista se describe el nombre de secuencia de registro para cada tipo de registro.

> **Note**
>
> A medida que aumentan los datos de la secuencia de registro, se rotan los nombres de la secuencia de registro. Cuando hay mucho flujo de registro para un determinado tipo de registro, puede ver el último flujo de registro si busca el nombre del flujo de registro con la Last Event Time (Hora del último evento) más reciente.

- Registros de componentes del servidor de API de Kubernetes (**api**): `kube-apiserver-1234567890abcdef01234567890abcde`
- Auditoría (**audit**): `kube-apiserver-audit-1234567890abcdef01234567890abcde`
- Autenticador (**authenticator**): `authenticator-1234567890abcdef01234567890abcde`
- Administrador de controladores (**controllerManager**): `kube-controller-manager-1234567890abcdef01234567890abcde`
- Programador (**scheduler**): `kube-scheduler-1234567890abcdef01234567890abcde`

4. Examine los eventos del flujo de registro.

   Por ejemplo, debería ver los indicadores iniciales del servidor de la API del clúster al ver la parte superior de `kube-apiserver-1234567890abcdef01234567890abcde`.

   > **Note**
   >
   > Si no ve los registros del servidor API al principio de la secuencia de registro, es probable que el archivo de registro del servidor API se haya rotado en el servidor antes de habilitar el registro del servidor API en el servidor. Los archivos de registro que se rotan antes de habilitar el registro del servidor de la API no se pueden exportar a CloudWatch.
   > Sin embargo, puede crear un nuevo clúster con la misma versión de Kubernetes y habilitar el registro del servidor API cuando cree el clúster. Los clústeres con la misma versión de plataforma tienen los mismos indicadores habilitados, por lo que sus indicadores deben coincidir con los indicadores del nuevo clúster. Cuando termine de ver los indicadores del nuevo clúster en CloudWatch, puede eliminar el nuevo clúster.

Registro de llamadas a la API de Amazon EKS con AWS CloudTrail

Amazon EKS se integra con AWS CloudTrail. CloudTrail es un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un servicio de AWS en Amazon EKS. CloudTrail captura todas las llamadas a la API para Amazon EKS como eventos. Esto incluye llamadas

realizadas desde la consola de Amazon EKS y las llamadas desde el código a las operaciones de la API de Amazon EKS.

Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3. Incluye eventos para Amazon EKS. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar ciertos detalles sobre una solicitud. Por ejemplo, puede determinar cuándo se realizó la solicitud a Amazon EKS, la dirección IP desde la que se realizó la solicitud y quién la realizó.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Temas

Información de Amazon EKS en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce alguna actividad en Amazon EKS, dicha actividad se registra en un evento de CloudTrail junto con los eventos de los demás servicios de AWS en Event history (Historial de eventos). Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de Amazon EKS, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. El seguimiento registra los eventos de todas las Regiones de AWS en la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los recursos siguientes.

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail

- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

Todas las acciones de Amazon EKS se registran en CloudTrail y están documentadas en Amazon EKS API Reference (Referencia de la API de Amazon EKS). Por ejemplo, las llamadas a las secciones `CreateCluster`, `ListClusters` y `DeleteCluster` generan entradas en los archivos de registro de CloudTrail.

Cada entrada de registro o de evento contiene información sobre el tipo de identidad de IAM que ha realizado la solicitud y las credenciales que se han utilizado. Si se utilizaron credenciales temporales, la entrada muestra cómo se obtuvieron las credenciales.

Para obtener más información, consulte el elemento userIdentity de CloudTrail.

Descripción de las entradas de archivos de registro de Amazon EKS

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos log de CloudTrail pueden contener una o varias entradas de log. Un evento representa una solicitud específica realizada desde una fuente y contiene información sobre la acción solicitada. Incluye información como la fecha y la hora de la acción y los parámetros de solicitud que se utilizaron. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateCluster`.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/username",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "username"
  },
  "eventTime": "2018-05-28T19:16:43Z",
  "eventSource": "eks.amazonaws.com",
```

```
  "eventName": "CreateCluster",
  "awsRegion": "region-code",
  "sourceIPAddress": "205.251.233.178",
  "userAgent": "PostmanRuntime/6.4.0",
  "requestParameters": {
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-a670c2df",
        "subnet-4f8c5004"
      ]
    },
    "roleArn": "arn:aws:iam::111122223333:role/AWSServiceRoleForAmazonEKS-
CAC1G1VH3ZKZ",
    "clusterName": "test"
  },
  "responseElements": {
    "cluster": {
      "clusterName": "test",
      "status": "CREATING",
      "createdAt": 1527535003.208,
      "certificateAuthority": {},
      "arn": "arn:aws:eks:region-code:111122223333:cluster/test",
      "roleArn": "arn:aws:iam::111122223333:role/AWSServiceRoleForAmazonEKS-
CAC1G1VH3ZKZ",
      "version": "1.10",
      "resourcesVpcConfig": {
        "securityGroupIds": [],
        "vpcId": "vpc-21277358",
        "subnetIds": [
          "subnet-a670c2df",
          "subnet-4f8c5004"
        ]
      }
    }
  },
  "requestID": "a7a0735d-62ab-11e8-9f79-81ce5b2b7d37",
  "eventID": "eab22523-174a-499c-9dd6-91e7be3ff8e3",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Entradas de registro para roles vinculados al servicio de Amazon EKS

Los roles vinculados al servicio de Amazon EKS realizan llamadas a la API para los recursos de AWS. Aparecen las entradas de registro de CloudTrail con `username: AWSServiceRoleForAmazonEKS` y `username: AWSServiceRoleForAmazonEKSNodegroup` para las llamadas realizadas por los roles vinculados al servicio de Amazon EKS. Para obtener más información acerca de Amazon EKS y los roles vinculados a servicios, consulte Utilizar roles vinculados a servicios para Amazon EKS.

En el siguiente ejemplo se incluye una entrada de registro de CloudTrail que muestra una acción `DeleteInstanceProfile` realizada por el rol vinculado al servicio de `AWSServiceRoleForAmazonEKSNodegroup`, que se indica en `sessionContext`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA3WHGPEZ7SJ2CW55C5:EKS",
        "arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForAmazonEKSNodegroup/EKS",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA3WHGPEZ7SJ2CW55C5",
                "arn": "arn:aws:iam::111122223333:role/aws-service-role/eks-
nodegroup.amazonaws.com/AWSServiceRoleForAmazonEKSNodegroup",
                "accountId": "111122223333",
                "userName": "AWSServiceRoleForAmazonEKSNodegroup"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-02-26T00:56:33Z"
            }
        },
        "invokedBy": "eks-nodegroup.amazonaws.com"
    },
    "eventTime": "2020-02-26T00:56:34Z",
    "eventSource": "iam.amazonaws.com",
    "eventName": "DeleteInstanceProfile",
```

```
    "awsRegion": "region-code",
    "sourceIPAddress": "eks-nodegroup.amazonaws.com",
    "userAgent": "eks-nodegroup.amazonaws.com",
    "requestParameters": {
        "instanceProfileName": "eks-11111111-2222-3333-4444-abcdef123456"
    },
    "responseElements": null,
    "requestID": "11111111-2222-3333-4444-abcdef123456",
    "eventID": "11111111-2222-3333-4444-abcdef123456",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

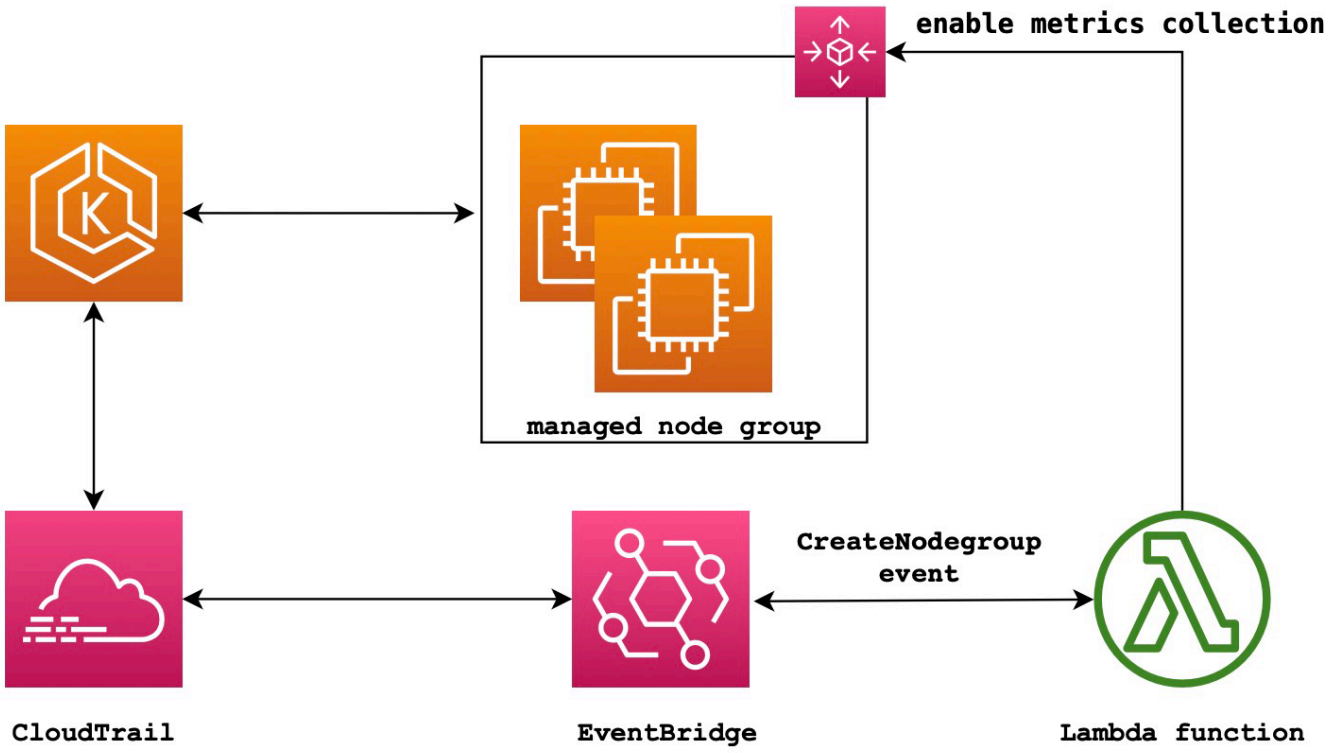
Habilitación de la recopilación de métricas de grupo de escalado automático

En este tema, se explica cómo habilitar la recopilación de métricas de grupo de escalado automático mediante AWS Lambda y AWS CloudTrail. Amazon EKS no habilita automáticamente la recopilación de métricas de grupo para los grupos de escalado automático creados para los nodos administrados.

Puede utilizar las métricas de grupo de escalado automático para realizar un seguimiento de los cambios en un grupo de escalado automático y configurar alarmas en los valores de umbral. Las métricas de grupo de escalado automático están disponibles en la consola de Auto Scaling o en la consola de Amazon CloudWatch. Una vez habilitadas, el grupo de escalado automático envía datos de muestra a Amazon CloudWatch cada minuto. No se aplica ningún cargo por habilitar estas métricas.

Al habilitar la recopilación de métricas de grupo de escalado automático, podrá supervisar el escalado de los grupos de nodos administrados. Las métricas de grupo de escalado automático indican el tamaño mínimo, máximo y deseado de un grupo de escalado automático. Puede crear una alarma si el número de nodos de un grupo de nodos está por debajo del tamaño mínimo, lo que indicaría que el grupo de nodos no funciona bien. El seguimiento del tamaño de un grupo de nodos también es útil para ajustar el recuento máximo de modo que el plano de datos no se quede sin capacidad.

Al crear un grupo de nodos administrado, AWS CloudTrail envía un evento `CreateNodegroup` a Amazon EventBridge. Al crear una regla de Amazon EventBridge que coincida con el evento `CreateNodegroup`, se activa una función de Lambda para habilitar la recopilación de métricas de grupo para el grupo de escalado automático asociado al grupo de nodos administrado.



Para habilitar la recopilación de métricas de grupo de escalado automático

1. Cree un rol de IAM para Lambda.

```
LAMBDA_ROLE=$(aws iam create-role \
  --role-name lambda-asg-enable-metrics \
  --assume-role-policy-document '{"Version": "2012-10-17","Statement":
 [{ "Effect": "Allow", "Principal": {"Service": "lambda.amazonaws.com"}, "Action":
 "sts:AssumeRole"}]}' \
  --output text \
  --query 'Role.Arn')
echo $LAMBDA_ROLE
```

2. Cree una política que permita describir los grupos de nodos de Amazon EKS y habilitar la recopilación de métricas de grupo de escalado automático.

```
cat > /tmp/lambda-policy.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
```

```
      {
        "Effect": "Allow",
        "Action": [
            "eks:DescribeNodegroup",
            "autoscaling:EnableMetricsCollection"
            ],
        "Resource": [
            "*"
        ]
      }
    ]
}
EOF
LAMBDA_POLICY_ARN=$(aws iam create-policy \
  --policy-name lambda-asg-enable-metrics-policy \
  --policy-document file:///tmp/lambda-policy.json \
  --output text \
  --query 'Policy.Arn')
echo $LAMBDA_POLICY_ARN
```

3. Adjunte la política de rol de IAM para Lambda.

```
aws iam attach-role-policy \
  --policy-arn $LAMBDA_POLICY_ARN \
  --role-name lambda-asg-enable-metrics
```

4. Agregue la política administrada `AWSLambdaBasicExecutionRole`, que tiene los permisos que la función necesita para escribir registros en los registros de CloudWatch.

```
aws iam attach-role-policy \
  --role-name lambda-asg-enable-metrics \
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
```

5. Cree el código de Lambda.

```
cat > /tmp/lambda-handler.py <<EOF
import json
import boto3
import time
import logging

eks = boto3.client('eks')
autoscaling = boto3.client('autoscaling')
```

```
logger = logging.getLogger()
logger.setLevel(logging.INFO)

def lambda_handler(event, context):
    ASG_METRICS_COLLLECTION_TAG_NAME = "ASG_METRICS_COLLLECTION_ENABLED"
    initial_retry_delay = 10
    attempts = 0

    #print(event)

    if not event["detail"]["eventName"] == "CreateNodegroup":
        print("invalid event.")
        return -1

    clusterName = event["detail"]["requestParameters"]["name"]
    nodegroupName = event["detail"]["requestParameters"]["nodegroupName"]
    try:
        metricsCollectionEnabled = event["detail"]["requestParameters"]["tags"]
[ASG_METRICS_COLLLECTION_TAG_NAME]
    except KeyError:
        print(ASG_METRICS_COLLLECTION_TAG_NAME, "tag not found.")
        return

    # Check if metrics collection is enabled in tags
    if metricsCollectionEnabled.lower() != "true":
        print("Metrics collection is not enabled in nodegroup tags.")
        return

    # Get the name of the associated autoscaling group
    print("Getting the autoscaling group name for nodegroup=", nodegroupName, ",
 cluster=", clusterName )
    for i in range(0,10):
        try:
            autoScalingGroup =
 eks.describe_nodegroup(clusterName=clusterName,nodegroupName=nodegroupName)
["nodegroup"]["resources"]["autoScalingGroups"][0]["name"]
        except:
            attempts += 1
            print("Failed to obtain the associated autoscaling group for
 nodegroup", nodegroupName, "Retrying in", initial_retry_delay*attempts,
 "seconds.")
            time.sleep(initial_retry_delay*attempts)
        else:
```

```
            break

    print("Enabling metrics collection on autoscaling group ", autoScalingGroup)

    # Enable metrics collection in the autoscaling group
    try:
        enableMetricsCollection =
 autoscaling.enable_metrics_collection(AutoScalingGroupName=autoScalingGroup,Granularity="1
    except:
        print("Unable to enable metrics collection on nodegroup=",nodegroup)
    print("Enabled metrics collection on nodegroup", nodegroupName)
EOF
```

6. Cree un paquete de implementación.

```
cd /tmp
zip function.zip lambda-handler.py
```

7. Creación de una función de Lambda.

```
LAMBDA_ARN=$(aws lambda create-function --function-name asg-enable-metrics-
collection \
  --zip-file fileb://function.zip --handler lambda-handler.lambda_handler \
  --runtime python3.9 \
  --timeout 600 \
  --role $LAMBDA_ROLE \
  --output text \
  --query 'FunctionArn')
echo $LAMBDA_ARN
```

8. Cree una regla de EventBridge.

```
RULE_ARN=$(aws events put-rule --name CreateNodegroupRuleToLambda \
  --event-pattern "{\"source\":[\"aws.eks\"],\"detail-type\":[\"AWS API Call via
 CloudTrail\"],\"detail\":{\"eventName\":[\"CreateNodegroup\"],\"eventSource\":
[\"eks.amazonaws.com\"]}}" \
  --output text \
  --query 'RuleArn')
echo $RULE_ARN
```

9. Agregue la función de Lambda como objetivo.

```
aws events put-targets --rule CreateNodegroupRuleToLambda \
```

```
  --targets "Id"="1","Arn"="$LAMBDA_ARN"
```

10. Agregue una política que permita a EventBridge invocar la función de Lambda.

```
aws lambda add-permission \
  --function-name asg-enable-metrics-collection \
  --statement-id CreateNodegroupRuleToLambda \
  --action 'lambda:InvokeFunction' \
  --principal events.amazonaws.com \
  --source-arn $RULE_ARN
```

La función de Lambda permite recopilar métricas de grupo de escalado automático para cualquier grupo de nodos administrado que etiquete con `ASG_METRICS_COLLLECTION_ENABLED` establecido como `TRUE`. Para confirmar que la recopilación de métricas de grupo de escalado automático está habilitada, vaya al grupo de escalado automático en la consola de Amazon EC2. En la pestaña Monitoring (Supervisión), verá que la casilla Enable (Habilitar) está activada.

Compatibilidad del componente de Amazon EKS para Operador ADOT

Amazon EKS admite el uso de AWS Management Console, AWS CLI y la API de Amazon EKS para instalar y administrar el operador AWS Distro for OpenTelemetry (ADOT). Esto posibilita que las aplicaciones se ejecutan en Amazon EKS para enviar datos de métricas y seguimiento a varias opciones de servicio de monitorización, como Amazon CloudWatch, Prometheus y X-Ray.

Para obtener más información, consulte Getting Started with AWS Distro for OpenTelemetry using EKS Add-Ons en la documentación de AWS Distro para OpenTelemetry.

Más servicios de AWS integrados con Amazon EKS

Además de los servicios mencionados en otras secciones, Amazon EKS trabaja con más servicios de AWS para ofrecer soluciones adicionales. En este tema, se identifican algunos de los demás servicios que utiliza Amazon EKS para agregar funcionalidad o los servicios que Amazon EKS utiliza para realizar tareas.

Temas

Crear recursos de Amazon EKS con AWS CloudFormation

Amazon EKS está integrado con AWS CloudFormation, un servicio que lo ayuda a modelar y configurar los recursos de AWS para que pueda dedicar menos tiempo a crear y administrar sus recursos e infraestructura. Puede crear una plantilla que describa todos los recursos de AWS que desea, por ejemplo un clúster de Amazon EKS, y AWS CloudFormation se encarga de aprovisionar y configurar esos recursos.

Cuando utiliza AWS CloudFormation, puede volver a utilizar la plantilla para configurar los recursos de Amazon EKS de forma coherente y repetida. Solo tiene que describir los recursos una vez y, luego, aprovisionar los mismos recursos una y otra vez en varias cuentas y regiones de AWS.

Amazon EKS y plantillas de AWS CloudFormation

Para aprovisionar y configurar los recursos de Amazon EKS y los servicios relacionados, debe entender las plantillas de AWS CloudFormation. Las plantillas son archivos de texto con formato de tipo JSON o YAML. Estas plantillas describen los recursos que desea aprovisionar en sus pilas de

AWS CloudFormation. Si no está familiarizado con JSON o YAML, puede utilizar Designer de AWS CloudFormation para comenzar a utilizar las plantillas de AWS CloudFormation. Para obtener más información, consulte ¿Qué es Designer de AWS CloudFormation? en la Guía del usuario de AWS CloudFormation.

Amazon EKS admite la creación de clústeres y grupos de nodos en AWS CloudFormation. Para obtener más información, incluidos ejemplos de plantillas JSON y YAML para los recursos de Amazon EKS, consulte la referencia del tipo de recurso de Amazon EKS en la Guía del usuario de AWS CloudFormation.

Obtener más información sobre AWS CloudFormation

Para obtener más información acerca de AWS CloudFormation, consulte los siguientes recursos:

- AWS CloudFormation
- Guía del usuario de AWS CloudFormation
- Guía del usuario de la interfaz de la línea de comandos de AWS CloudFormation

Amazon EKS y AWS Local Zones

Una zona local de AWS es una extensión de una Región de AWS cercana geográficamente a los usuarios. Las Local Zones tienen sus propias conexiones a Internet y admiten AWS Direct Connect. Los recursos creados en una zona local pueden prestar servicio a los usuarios locales con comunicaciones de baja latencia. Para obtener más información, consulte Local Zones (Zonas locales).

Amazon EKS admite ciertos recursos en Local Zones. Esto incluye nodos autoadministrados de Amazon EC2, los volúmenes de Amazon EBS y los equilibradores de carga de aplicaciones (ALB). Recomendamos que tenga en cuenta lo siguiente al utilizar Local Zones como parte del clúster de Amazon EKS.

Nodos

No se pueden crear grupos de nodos administrados en o nodos Fargate en Local Zones con Amazon EKS. Sin embargo, puede crear nodos autoadministrados de Amazon EC2 en Local Zones mediante la API de Amazon EC2, AWS CloudFormation o bien `eksctl`. Para obtener más información, consulte Nodos autoadministrados.

Arquitectura de redes

- El plano de control de Kubernetes administrado por Amazon EKS siempre se ejecuta en la Región de AWS. El plano de control de Kubernetes administrado por Amazon EKS no se puede ejecutar en la zona local. Dado que las Local Zones aparecen como una subred dentro de la VPC, Kubernetes ve los recursos de la zona local como parte de esa subred.
- El clúster de Kubernetes de Amazon EKS se comunica con las instancias de Amazon EC2 que ejecuta en la Región de AWS o zona local mediante interfaces de redes elásticas administradas por Amazon EKS. Para obtener más información sobre la arquitectura de redes de Amazon EKS, consulte Redes de Amazon EKS.
- A diferencia de las subredes regionales, Amazon EKS no puede colocar interfaces de redes en las subredes de zona local. Esto significa que no debe especificar las subredes de zona local al crear el clúster.

Deep Learning Containers

Los contenedores de aprendizaje profundo de AWS son un conjunto de imágenes de Docker para entrenar y trabajar con modelos en TensorFlow en Amazon EKS y Amazon Elastic Container Service (Amazon ECS). Los contenedores de aprendizaje profundo proporcionan entornos optimizados con bibliotecas de TensorFlow, NVIDIA CUDA (para instancias de GPU) y MKL de Intel (para instancias de CPU) y están disponibles en Amazon ECR.

Para empezar a utilizar los contenedores de aprendizaje profundo de AWS en Amazon EKS, consulte Configuración de Amazon EKS en la Guía para desarrolladores de contenedores de aprendizaje profundo de AWS.

Amazon VPC Lattice

Amazon VPC Lattice es un servicio de redes de aplicaciones administrado totalmente e integrado directamente en la infraestructura de red de AWS que puede utilizar para conectar, proteger y supervisar sus servicios en varias cuentas y nubes privadas virtuales (VPC). Con Amazon EKS, puede aprovechar Amazon VPC Lattice mediante el uso del controlador de la API de AWS de puerta de enlace, una implementación de la API de puerta de enlace de Kubernetes. Con Amazon VPC Lattice, puede configurar la conectividad entre clústeres con semántica Kubernetes estándar de forma sencilla y coherente. Para empezar a utilizar Amazon VPC Lattice con Amazon EKS, consulte la Guía del usuario del controlador de la API de puerta de enlace de AWS.

AWS Resilience Hub

AWS Resilience Hub evalúa la resiliencia de un clúster de Amazon EKS mediante el análisis de su infraestructura. AWS Resilience Hub utiliza la configuración de control de acceso basado en roles (RBAC) de Kubernetes para evaluar las cargas de trabajo de Kubernetes implementadas en el clúster. Para obtener más información, consulte Habilitación del acceso de AWS Resilience Hub a un clúster de Amazon EKS en la Guía del usuario de AWS Resilience Hub.

Detección de amenazas con Amazon GuardDuty

Amazon GuardDuty es un servicio de detección de amenazas que ayuda a proteger las cuentas, los contenedores, las cargas de trabajo y los datos de su entorno de AWS. Mediante modelos de machine learning (ML) y capacidades de detección de anomalías y amenazas, GuardDuty supervisa continuamente los diferentes orígenes de registro y la actividad en tiempo de ejecución para identificar y priorizar los posibles riesgos de seguridad y actividades maliciosas en su entorno.

Entre otras características, GuardDuty ofrece las siguientes dos características que detectan posibles amenazas para sus clústeres de EKS: Protección de EKS y Supervisión en tiempo de ejecución.

Protección de EKS

> Esta característica proporciona una cobertura de detección de amenazas para ayudarlo a proteger los clústeres de Amazon EKS mediante la supervisión de los registros de auditoría de Kubernetes asociados. Los registros de auditoría de Kubernetes capturan las acciones secuenciales del clúster, incluidas las actividades de los usuarios, las aplicaciones que utilizan la API de Kubernetes y el plano de control. Por ejemplo, GuardDuty puede identificar que las API llamadas para manipular los recursos de un clúster de Kubernetes fueron invocadas por un usuario no autenticado.
>
> Al activar la Protección de EKS, GuardDuty solo podrá acceder a los registros de auditoría de Amazon EKS para detectar amenazas de forma continua. Si GuardDuty identifica una amenaza potencial para el clúster, genera un resultado del registro de auditoría de Kubernetes asociado de un tipo específico. Para obtener más información sobre los tipos de resultados disponibles en los registros de auditoría de Kubernetes, consulte los tipos de resultados de registros de auditoría de Kubernetes en la Guía del usuario de Amazon GuardDuty.
>
> Para obtener más información, consulte Protección de EKS en la Guía del usuario de Amazon GuardDuty.

Supervisión en tiempo de ejecución

Esta característica supervisa y analiza los eventos de archivos, redes y sistemas operativos para ayudarlo a detectar posibles amenazas en cargas de trabajo específicas de AWS en su entorno.

Cuando habilita la Supervisión del tiempo de ejecución e instala el agente GuardDuty en sus clústeres de Amazon EKS, GuardDuty comienza a supervisar los eventos de tiempo de ejecución asociados a este clúster. Si GuardDuty identifica una amenaza potencial para el clúster, genera un resultado de la supervisión del tiempo de ejecución asociado. Por ejemplo, una amenaza puede empezar por comprometer un único contenedor que ejecuta una aplicación web vulnerable. Es posible que esta aplicación web tenga permisos de acceso a los contenedores y las cargas de trabajo subyacentes. En este escenario, las credenciales mal configuradas podrían dar lugar a un acceso más amplio a la cuenta y a los datos almacenados en ella.

Para configurar Runtime Monitoring, instale el agente de GuardDuty en el clúster como complemento de Amazon EKS. Para obtener más información del complemento, consulte Complementos de Amazon EKS disponibles en Amazon EKS.

Para obtener más información, consulte Runtime Monitoring en la Guía del usuario de Amazon GuardDuty.

Uso de Amazon Security Lake con Amazon EKS

Amazon Security Lake es un servicio de lago de datos de seguridad totalmente administrado que le permite centralizar los datos de seguridad de varios orígenes, incluido Amazon EKS. Al integrar Amazon EKS con Security Lake, puede obtener información más detallada sobre las actividades que se llevan a cabo en sus recursos de Kubernetes y mejorar la seguridad de sus clústeres de Amazon EKS.

> **Note**
>
> Para obtener más información sobre el uso de Security Lake con Amazon EKS y la configuración de los orígenes de datos, consulte la documentación de Amazon Security Lake.

Ventajas de usar Security Lake con Amazon EKS

Datos de seguridad centralizados: Security Lake recopila y centraliza automáticamente los datos de seguridad de sus clústeres de Amazon EKS, junto con datos de otros servicios de AWS, proveedores

de SaaS, orígenes en las instalaciones y orígenes de terceros. Esto proporciona una visión completa de su postura de seguridad en toda su organización.

Formato de datos estandarizado: Security Lake convierte los datos recopilados al formato Open Cybersecurity Schema Framework (OCSF), que es un esquema estándar de código abierto. Esta normalización facilita el análisis y la integración con otras herramientas y servicios de seguridad.

Detección de amenazas mejorada: al analizar los datos de seguridad centralizados, incluidos los registros del plano de control de Amazon EKS, puede detectar actividades potencialmente sospechosas en sus clústeres de Amazon EKS de manera más eficaz. Esto ayuda a identificar y responder rápidamente a los incidentes de seguridad.

Administración de datos simplificada: Security Lake administra el ciclo de vida de sus datos de seguridad con configuraciones de retención y replicación personalizables. Esto simplifica las tareas de administración de datos y garantiza que retenga los datos necesarios para fines de cumplimiento y auditoría.

Habilitación de Security Lake en Amazon EKS

Para comenzar a usar Security Lake con Amazon EKS, siga los pasos que se indican a continuación:

1. Active el registro del plano de control de Amazon EKS en sus clústeres de EKS. Consulte Habilitar y deshabilitar registros de plano de control para obtener instrucciones detalladas.
2. Agregue los registros de auditoría de Amazon EKS como origen en Security Lake. A continuación, Security Lake comenzará a recopilar información detallada sobre las actividades hechas en los recursos de Kubernetes que se ejecutan en sus clústeres de EKS.
3. Configure los ajustes de retención y replicación de sus datos de seguridad en Security Lake en función de sus requisitos.
4. Utilice los datos OCSF normalizados almacenados en Security Lake para responder a incidentes, llevar a cabo análisis de seguridad e integrarlos con otros servicios de AWS o herramientas de terceros. Por ejemplo, puede generar información sobre seguridad a partir de los datos de Amazon Security Lake mediante Amazon OpenSearch Ingestion.

Análisis de los registros de EKS en Security Lake

Security Lake normaliza los eventos de registro de EKS al formato OCSF, lo que facilita el análisis y la correlación de los datos con otros eventos de seguridad. Puede utilizar diversas herramientas

y servicios, como Amazon Athena, Amazon QuickSight o herramientas de análisis de seguridad de terceros, para consultar y visualizar los datos normalizados.

Para obtener más información sobre la asignación de OCSF para los eventos de registro de EKS, consulte la referencia de asignación en el repositorio de GitHub de OCSF.

Amazon Detective

Amazon Detective ayuda a analizar, investigar e identificar rápidamente la causa raíz de resultados de seguridad o actividades sospechosas. Detective recopila automáticamente los datos de registro de sus recursos de AWS. A continuación, utiliza el machine learning, el análisis estadístico y la teoría de grafos para generar visualizaciones que lo ayuden a realizar investigaciones sobre la seguridad con mayor rapidez y de forma más eficaz. Las agregaciones de datos, los resúmenes y los contextos prediseñados de Detective ayudan a analizar y determinar rápidamente la naturaleza y el alcance de los posibles problemas de seguridad. Para obtener más información, consulte la Guía del usuario de Amazon Detective.

Detective organiza los datos de Kubernetes y AWS en resultados tales como:

- Detalles del clúster de Amazon EKS, incluyendo la identidad de IAM que creó el clúster y el rol de servicio del clúster. Puede investigar la actividad de la API de AWS y Kubernetes de estas identidades de IAM con Detective.
- Detalles del contenedor, como la imagen y el contexto de seguridad. También puede revisar los detalles de los Pods finalizados.
- Actividad de la API de Kubernetes, incluyendo tendencias generales de actividad de la API y detalles sobre llamadas a API específicas. Por ejemplo, puede mostrar el número de llamadas a la API de Kubernetes procesadas correctamente y fallidas que se han emitido durante un intervalo de tiempo seleccionado. Además, la sección sobre llamadas a la API observadas recientemente puede resultar útil para identificar actividades sospechosas.

Los registros de auditoría de Amazon EKS son un paquete de orígenes de datos opcionales que se puede agregar a su gráfico de comportamiento de Detective. Puede ver los paquetes de orígenes opcionales disponibles y su estado en su cuenta. Para obtener más información, consulte Amazon EKS audit logs for Detective en la Guía del usuario de Amazon Detective.

Uso de Amazon Detective con Amazon EKS

Para revisar los resultados de un clúster de Amazon EKS

Para poder revisar los resultados, Detective debe estar habilitado durante al menos 48 horas en la misma Región de AWS donde se encuentre el clúster. Para obtener más información, consulte Setting up Amazon Detective en la Guía del usuario de Amazon Detective.

1. Abra la consola de Detective en https://console.aws.amazon.com/detective/.
2. En el panel de navegación izquierdo, seleccione Buscar.
3. Seleccione Elegir tipo y, a continuación, Clúster de EKS.
4. Escriba el nombre o ARN del clúster y, a continuación, seleccione Buscar.
5. En los resultados de la búsqueda, seleccione el nombre del clúster cuya actividad desea ver. Para obtener más información sobre lo que puede ver, consulte Overall Kubernetes API activity involving an Amazon EKS cluster en la Guía del usuario de Amazon Detective.

Solución de problemas de Amazon EKS

En este capítulo se tratan algunos errores habituales que pueden aparecer al utilizar Amazon EKS y cómo solucionarlos. Si necesita solucionar problemas en áreas específicas de Amazon EKS, consulte los temas aparte Solución de problemas de IAM, Solución de problemas en Amazon EKS Connector y Troubleshooting for ADOT using EKS Add-Ons.

Para obtener más información sobre la solución de problemas, consulte el contenido de Centro de conocimientos sobre Amazon Elastic Kubernetes Service en AWS re:Post.

Capacidad insuficiente

Si aparece el siguiente error al intentar crear un clúster de Amazon EKS, significa que una de las zonas de disponibilidad que ha especificado no tiene capacidad suficiente para admitir un clúster.

```
Cannot create cluster 'example-cluster' because region-1d, the targeted Availability Zone, does not currently have sufficient capacity to support the cluster. Retry and choose from these Availability Zones: region-1a, region-1b, region-1c
```

Intente crear de nuevo el clúster con subredes de la VPC de clúster alojadas en las zonas de disponibilidad indicadas en el mensaje de error.

Hay zonas de disponibilidad en las que un clúster no puede residir. Compare las zonas de disponibilidad en las que se encuentran sus subredes con la lista de zonas de disponibilidad de los Requisitos y consideraciones de la subred.

Los nodos no pueden unirse al clúster

Hay algunos motivos que suelen impedir que los nodos se unan al clúster:

- Si los nodos son nodos administrados, Amazon EKS añade entradas al `ConfigMap` de `aws-auth` al crear el grupo de nodos. Si la entrada se eliminó o modificó, tendrá que volver a agregarla. Para obtener más información, ingrese **`eksctl create iamidentitymapping --help`** en su terminal. Puede comprobar las entradas actuales de `aws-auth` `ConfigMap` reemplazando *`my-cluster`* en el siguiente comando por el nombre de su clúster y luego ejecutando el comando modificado: **`eksctl get iamidentitymapping --cluster`** ***`my-cluster`***. El ARN del rol que especifique no puede incluir una ruta que no sea `/`. Por ejemplo, si el nombre de su rol es

`development/apps/my-role`, deberá cambiarlo a `my-role` cuando especifique el ARN del rol. Asegúrese de especificar el ARN del rol de IAM correspondiente al nodo (y no el ARN del perfil de instancia).

Si los nodos son autoadministrados y no ha creado entradas de acceso para el ARN del rol de IAM del nodo, ejecute los mismos comandos indicados para los nodos administrados. Si ha creado una entrada de acceso para el ARN del rol de IAM de su nodo, es posible que no esté configurado correctamente en la entrada de acceso. Asegúrese de especificar el ARN del rol de IAM del nodo (y no el ARN del perfil de instancia) como el ARN de la entidad principal en su entrada o entrada de acceso del `ConfigMap` de `aws-auth`. Para obtener más información acerca de las entradas de acceso, consulte Administración de entradas de acceso.

- El valor de ClusterName (Nombre de clúster) en la plantilla de AWS CloudFormation del nodo no coincide exactamente con el nombre del clúster al que desea que se unan los nodos. Si se especifica un valor incorrecto en este campo, la configuración del archivo `/var/lib/kubelet/kubeconfig` del nodo no será correcta y los nodos no podrán unirse al clúster.
- El nodo no está etiquetado como propiedad del clúster. Los nodos deben tener la siguiente etiqueta aplicada a ellos, donde *`my-cluster`* se reemplaza por el nombre del clúster.

Clave	Valor
`kubernetes.io/cluster/` *`my-cluster`*	`owned`

- Es posible que los nodos no puedan tener acceso al clúster mediante una dirección IP pública. Asegúrese de que los nodos implementados en subredes públicas tengan asignada una dirección IP pública. Si no es así, puede asociar una dirección IP elástica a un nodo después de que se lance. Para obtener más información, consulte Asociación de una dirección IP elástica a una instancia de ejecución o una interfaz de red. Si la subred pública no está configurada para asignar automáticamente direcciones IP públicas a instancias implementadas en ella, recomendamos habilitar esa configuración. Para obtener más información, consulte Modificación del atributo de direcciones `IPv4` públicas de su subred. Si se implementa el nodo en una subred privada, la subred debe tener una ruta a una puerta de enlace NAT que tenga asignada una dirección IP pública.
- El punto de conexión de AWS STS para la Región de AWS en la que está implementando los nodos no está habilitado para su cuenta. Para habilitar la región, consulte Activación y desactivación de AWS STS en una Región de AWS.

- El nodo no tiene una entrada DNS privada, lo que provoca que el registro de `kubelet` contenga un error de `node "" not found`. Asegúrese de que la VPC donde se crea el nodo tenga valores establecidos para `domain-name` y `domain-name-servers` como `Options` en un `DHCP options set`. Los valores predeterminados son `domain-name:<region>.compute.internal` y `domain-name-servers:AmazonProvidedDNS`. Para más información, consulte Conjuntos de opciones de DHCP en la Guía del usuario de Amazon VPC.
- Si los nodos del grupo de nodos gestionados no se conectan al clúster dentro de 15 minutos, se emitirá un problema de estado con el nombre «NodeCreationFailure» y el estado de la consola se establecerá en `Create failed`. En el caso de las AMI de Windows con tiempos de inicio lentos, este problema se puede resolver con el inicio rápido.

Para identificar y solucionar problemas de causas comunes que impiden que los nodos de trabajo se unan a un clúster, puede usar el manual de procedimientos `AWSSupport-TroubleshootEKSWorkerNode`. Para obtener más información, consulte `AWSSupport-TroubleshootEKSWorkerNode` en la Referencia del manual de procedimientos de automatización de AWS Systems Manager.

Acceso denegado o no autorizado (**`kubectl`**)

Si aparece uno de los siguientes errores al ejecutar los comandos de `kubectl`, eso significa que su `kubectl` no está configurado correctamente para Amazon EKS o las credenciales para la entidad principal de IAM (rol u usuario) que utiliza no están asignadas a un nombre de usuario de Kubernetes con suficientes permisos para los objetos de Kubernetes en su clúster de Amazon EKS.

- `could not get token: AccessDenied: Access denied`
- `error: You must be logged in to the server (Unauthorized)`
- `error: the server doesn't have a resource type "svc"`

Esto podría deberse a una de las siguientes razones:

- El clúster se creó con credenciales para una entidad principal de IAM y `kubectl` utiliza credenciales para otra entidad principal de IAM. Para resolver este problema, actualice el archivo `kube config` para usar las credenciales que crearon el clúster. Para obtener más información, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

- Si el clúster cumple con los requisitos mínimos de plataforma de la sección de requisitos previos de Administración de entradas de acceso, no existe una entrada de acceso con su entidad principal de IAM. Si existe, no tiene definidos los nombres de grupo de Kubernetes necesarios o no tiene asociada la política de acceso adecuada. Para obtener más información, consulte Administración de entradas de acceso.
- Si su clúster no cumple con los requisitos mínimos de plataforma en Administración de entradas de acceso, no existe una entrada con su entidad principal de IAM en el `ConfigMap` de `aws-auth`. Si existe, no está asignada a nombres de grupo de Kubernetes vinculados a un `Role` o `ClusterRole` de Kubernetes con los permisos necesarios. Para obtener más información sobre los objetos de autorización basada en roles (RBAC) de Kubernetes, consulte Uso de la autorización de RBAC en la documentación de Kubernetes. Puede comprobar las entradas actuales de `aws-auth` `ConfigMap` reemplazando *`my-cluster`* en el siguiente comando por el nombre de su clúster y luego ejecutando el comando modificado: **`eksctl get iamidentitymapping --cluster`** ***`my-cluster`***. Si no hay una entrada para el ARN de su entidad principal de IAM en el `ConfigMap`, introduzca **`eksctl create iamidentitymapping --help`** en su terminal para aprender a crear una.

Si instala y configura la AWS CLI, puede configurar las credenciales de IAM que utiliza. Para obtener más información, consulte Configuración de la AWS CLI en la Guía del usuario de AWS Command Line Interface. También puede configurar `kubectl` para utilizar un rol de IAM si asume un rol de IAM para acceder a los objetos de Kubernetes en su clúster. Para obtener más información, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

`hostname doesn't match`

La versión de Python de su sistema debe ser `2.7.9` o posterior. En caso contrario, se producirán errores `hostname doesn't match` en las llamadas de la AWS CLI a Amazon EKS. Para obtener más información, consulte ¿Qué significan los errores “el nombre del host no coincide”? en Preguntas frecuentes sobre las solicitudes de Python.

`getsockopt: no route to host`

Docker se ejecuta en el rango de CIDR `172.17.0.0/16` en los clústeres de Amazon EKS. Le recomendamos que las subredes de la VPC de su clúster no se superpongan. De lo contrario, recibirá el siguiente error:

```
Error: : error upgrading connection: error dialing backend: dial tcp
 172.17.<nn>.<nn>:10250: getsockopt: no route to host
```

Instances failed to join the Kubernetes cluster

Si aparece el error `Instances failed to join the Kubernetes cluster` en la AWS Management Console, asegúrese de que esté habilitado el acceso privado al punto de conexión del clúster o de que haya configurado correctamente los bloques de CIDR para el acceso público al punto de conexión. Para obtener más información, consulte Control de acceso al punto de conexión del clúster de Amazon EKS.

Códigos de error del grupo de nodos administrado

Si el grupo de nodos administrado encuentra un problema de estado del hardware, Amazon EKS devuelve un código de error para ayudarlo a diagnosticar el problema. Estas comprobaciones de estado no detectan problemas de software porque se basan en comprobaciones de estado de Amazon EC2. En la siguiente lista se describen los códigos de error.

AccessDenied

Amazon EKS, o uno o varios de sus nodos administrados, no pueden autenticarse ni autorizarse con su servidor de la API del clúster de Kubernetes. Para obtener más información sobre la resolución de una causa común, consulte Resolución de una causa común de errores `AccessDenied` para grupos de nodos administrados.. Las AMI privadas de Windows también pueden provocar este código de error junto con el mensaje de error `Not authorized for images`. Para obtener más información, consulte `Not authorized for images`.

AmiIdNotFound

No hemos podido encontrar el ID de la AMI asociado a su plantilla de lanzamiento. Asegúrese de que la AMI existe y de que se comparte con su cuenta.

AutoScalingGroupNotFound

No se pudo encontrar el grupo de escalado automático asociado al grupo de nodos administrados. Es posible que pueda volver a crear un grupo de Auto Scaling con la misma configuración para recuperarlo.

ClusterUnreachable

Amazon EKS, o uno o varios de los nodos administrados, no pueden comunicarse con el servidor de la API del clúster de Kubernetes. Esto puede suceder si hay interrupciones en la red o si los servidores de la API están agotando el tiempo de espera de las solicitudes de procesamiento.

Ec2SecurityGroupNotFound

No se pudo encontrar el grupo de seguridad del clúster para el clúster. Debe volver a crear el clúster.

Ec2SecurityGroupDeletionFailure

No hemos podido eliminar el grupo de seguridad de acceso remoto para el grupo de nodos administrados. Elimine cualquier dependencia del grupo de seguridad.

Ec2LaunchTemplateNotFound

No hemos podido encontrar la plantilla de lanzamiento de Amazon EC2 para el grupo de nodos administrados. Debe volver a crear el grupo de nodos para recuperarlo.

Ec2LaunchTemplateVersionMismatch

La versión de la plantilla de lanzamiento de Amazon EC2 para el grupo de nodos administrados no coincide con la versión creada por Amazon EKS. Es posible que pueda volver a la versión que creó Amazon EKS para recuperarla.

IamInstanceProfileNotFound

No hemos podido encontrar el perfil de instancia de IAM para su grupo de nodos administrados. Es posible que pueda volver a crear un perfil de instancia con la misma configuración para recuperarlo.

IamNodeRoleNotFound

No hemos podido encontrar el rol de IAM para su grupo de nodos gestionados. Es posible que pueda volver a crear un rol de IAM con la misma configuración para recuperarlo.

AsgInstanceLaunchFailures

Su grupo de escalado automático está experimentando errores al intentar lanzar instancias.

NodeCreationFailure

Las instancias lanzadas no pueden registrarse en el clúster de Amazon EKS. Las causas comunes de este error son permisos del rol de IAM del nodo insuficientes o falta de acceso a Internet saliente para los nodos. Los nodos deben cumplir cualquiera de los siguientes requisitos:

- Capaz de acceder a internet mediante una dirección IP pública. El grupo de seguridad asociado a la subred en la que se encuentra el nodo debe permitir la comunicación. Para obtener más información, consulte Requisitos y consideraciones de la subred y Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS.
- Los nodos y la VPC deben cumplir los requisitos de Requisitos del clúster privado.

InstanceLimitExceeded

Su cuenta de AWS no puede lanzar más instancias del tipo de instancia especificado. Es posible que pueda solicitar un aumento del límite de instancias de Amazon EC2 para recuperarlas.

InsufficientFreeAddresses

Una o varias de las subredes asociadas al grupo de nodos administrados no tienen suficientes direcciones IP disponibles para nuevos nodos.

InternalFailure

Estos errores suelen ser causados por un problema del lado del servidor de Amazon EKS.

Resolución de una causa común de errores **`AccessDenied`** para grupos de nodos administrados.

La causa más común de los errores de `AccessDenied` al realizar operaciones en grupos de nodos administrados es la falta de `eks:node-manager`, `ClusterRole` o `ClusterRoleBinding`. Amazon EKS configura estos recursos en el clúster como parte de la incorporación con grupos de nodos administrados, y estos son necesarios para administrar los grupos de nodos.

`ClusterRole` puede cambiar con el tiempo, pero debe parecerse al siguiente ejemplo:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: eks:node-manager
rules:
- apiGroups:
  - ''
  resources:
  - pods
  verbs:
  - get
```

```
  - list
  - watch
  - delete
- apiGroups:
  - ''
  resources:
  - nodes
  verbs:
  - get
  - list
  - watch
  - patch
- apiGroups:
  - ''
  resources:
  - pods/eviction
  verbs:
  - create
```

`ClusterRoleBinding` puede cambiar con el tiempo, pero debe parecerse al siguiente ejemplo:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: eks:node-manager
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: eks:node-manager
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: eks:node-manager
```

Verifique que el `ClusterRole` de `eks:node-manager` existe.

```
kubectl describe clusterrole eks:node-manager
```

Si está presente, compare la salida con el ejemplo de `ClusterRole` anterior.

Verifique que el `ClusterRoleBinding` de `eks:node-manager` existe.

```
kubectl describe clusterrolebinding eks:node-manager
```

Si está presente, compare la salida con el ejemplo de `ClusterRoleBinding` anterior.

Si ha identificado un `ClusterRole` o `ClusterRoleBinding` ausente o defectuoso como la causa de un error de `AcessDenied` al solicitar operaciones de grupo de nodos administrados, puede restaurarlos. Guarde los siguientes contenidos en un archivo llamado *`eks-node-manager-role.yaml`*.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: eks:node-manager
rules:
- apiGroups:
  - ''
  resources:
  - pods
  verbs:
  - get
  - list
  - watch
  - delete
- apiGroups:
  - ''
  resources:
  - nodes
  verbs:
  - get
  - list
  - watch
  - patch
- apiGroups:
  - ''
  resources:
  - pods/eviction
  verbs:
  - create
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
```

```
  name: eks:node-manager
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: eks:node-manager
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: eks:node-manager
```

Aplique el archivo.

```
kubectl apply -f eks-node-manager-role.yaml
```

Vuelva a intentar la operación del grupo de nodos para ver si se resolvió el problema.

`Not authorized for images`

Una posible causa de un mensaje de error `Not authorized for images` es el uso de una AMI privada de Windows de Amazon EKS para lanzar grupos de nodosWindows administrados. Tras lanzar nuevas AMI de Windows, AWS convierte en privadas las AMI de más de cuatro meses de antigüedad, por lo que ya no se puede acceder a ellas. Si el grupo de nodos que administra utiliza una AMI de Windows privada, considere actualizar el grupo de nodos de Windows. Si bien no podemos garantizar que podamos proporcionar acceso a las AMI que se han convertido en privadas, puede solicitar el acceso enviando un ticket a AWS Support. Para obtener más información sobre pares de claves, consulte Revisiones, actualizaciones e ID de AMI en la Guía del usuario de Amazon EC2.

El nodo está en estado **`NotReady`**

Si el nodo entra en estado `NotReady`, es probable que esto indique que el nodo está en mal estado y no está disponible para programar nuevos Pods. Esto puede ocurrir por varios motivos, como que el nodo carezca de recursos suficientes para la CPU, la memoria o el espacio disponible en el disco.

En el caso de las AMI de Windows optimizadas para Amazon EKS, no hay reservas para los recursos de cómputos especificados de forma predeterminada en la configuración de `kubelet`. Para evitar problemas de recursos, puede reservar recursos de cómputos para los procesos del sistema proporcionando valores de configuración de `kube-reserved` o `system-reserved` al `kubelet`. Para ello, utilice el parámetro de línea de comandos `-KubeletExtraArgs` del script de

arranque. Para obtener más información, consulte Reservar recursos de cómputos para los daemons del sistema en la documentación de Kubernetes y los Parámetros de configuración del script de arranque en esta guía del usuario.

Herramienta de recopilación de registros de CNI

El Amazon VPC CNI plugin for de Kubernetes tiene su propio script de resolución de problemas que está disponible en los nodos en `/opt/cni/bin/aws-cni-support.sh`. Puede utilizar el script a fin de recopilar registros de diagnóstico para casos de soporte y solución de problemas general.

Utilice el siguiente comando para ejecutar el script en su nodo:

```
sudo bash /opt/cni/bin/aws-cni-support.sh
```

> **Note**
>
> Si el script no está presente en esa ubicación, el contenedor de CNI no podrá ejecutarse. Puede descargar y ejecutar manualmente el script con el siguiente comando:
>
> ```
> curl -O https://raw.githubusercontent.com/awslabs/amazon-eks-ami/master/log-collector-script/linux/eks-log-collector.sh
> sudo bash eks-log-collector.sh
> ```

El script recopila la siguiente información de diagnóstico: La versión de CNI que ha implementado puede ser anterior a la versión del script.

```
      This is version 0.6.1. New versions can be found at https://github.com/awslabs/
amazon-eks-ami

Trying to collect common operating system logs...
Trying to collect kernel logs...
Trying to collect mount points and volume information...
Trying to collect SELinux status...
Trying to collect iptables information...
Trying to collect installed packages...
Trying to collect active system services...
Trying to collect Docker daemon information...
Trying to collect kubelet information...
Trying to collect L-IPAMD information...
```

```
Trying to collect sysctls information...
Trying to collect networking information...
Trying to collect CNI configuration information...
Trying to collect running Docker containers and gather container data...
Trying to collect Docker daemon logs...
Trying to archive gathered information...

 Done... your bundled logs are located in /var/
log/eks_i-0717c9d54b6cfaa19_2020-03-24_0103-UTC_0.6.1.tar.gz
```

La información de diagnóstico se recopila y se almacena en:

```
/var/log/eks_i-0717c9d54b6cfaa19_2020-03-24_0103-UTC_0.6.1.tar.gz
```

La red de tiempo de ejecución del contenedor no está lista

Puede recibir un error `Container runtime network not ready` y errores de autorización similares a los siguientes:

```
4191 kubelet.go:2130] Container runtime network not ready: NetworkReady=false
 reason:NetworkPluginNotReady message:docker: network plugin is not ready: cni config
 uninitialized
4191 reflector.go:205] k8s.io/kubernetes/pkg/kubelet/kubelet.go:452: Failed to list
 *v1.Service: Unauthorized
4191 kubelet_node_status.go:106] Unable to register node
 "ip-10-40-175-122.ec2.internal" with API server: Unauthorized
4191 reflector.go:205] k8s.io/kubernetes/pkg/kubelet/kubelet.go:452: Failed to list
 *v1.Service: Unauthorized
```

Esto puede deberse a una de las siguientes razones:

1. O bien no tiene un `ConfigMap` de `aws-auth` en su clúster, o este no incluye entradas para el rol de IAM con el que configuró sus nodos.

   Esta entrada de `ConfigMap` es necesaria si sus nodos cumplen uno de los siguientes criterios:

   - Nodos administrados en un clúster con cualquier versión de Kubernetes o la plataforma.
   - Nodos autoadministrados en un clúster anterior a una de las versiones de la plataforma que se indican en la sección de requisitos previos del tema Administración de entradas de acceso.

Para resolver este problema, consulte las entradas existentes en su `ConfigMap` reemplazando *`my-cluster`* en el siguiente comando con el nombre de su clúster y, luego, ejecute el comando modificado: **`eksctl get iamidentitymapping --cluster my-cluster`**. Si recibe un mensaje de error del comando, puede que esto se deba a que su clúster no tiene un `aws-auth` `ConfigMap`. El siguiente comando añade una entrada al `ConfigMap`. Si `ConfigMap` no existe, el comando también lo crea. Reemplace *`111122223333`* por el ID de la Cuenta de AWS para el rol de IAM y *`myAmazonEKSNodeRole`* por el nombre del rol de su nodo.

```
eksctl create iamidentitymapping --cluster my-cluster \
    --arn arn:aws:iam::111122223333:role/myAmazonEKSNodeRole --group
 system:bootstrappers,system:nodes \
    --username system:node:{{EC2PrivateDNSName}}
```

El ARN del rol que especifique no puede incluir una ruta que no sea `/`. Por ejemplo, si el nombre de su rol es `development/apps/my-role`, tendrá que cambiarlo por `my-role` cuando especifique el ARN del rol. Asegúrese de especificar el ARN del rol de IAM correspondiente al nodo (y no el ARN del perfil de instancia).

2. Los nodos autoadministrados se encuentran en un clúster con una versión de la plataforma que es la versión mínima indicada en los requisitos previos del tema Administración de entradas de acceso, pero no aparece ninguna entrada en `aws-auth` `ConfigMap` (consulte el punto anterior) para el rol de IAM del nodo o no existe una entrada de acceso para ese rol. Para resolver el problema, consulte las entradas de acceso existentes reemplazando *`my-cluster`* en el siguiente comando por el nombre de su clúster y luego ejecutando el comando modificado: **`aws eks list-access-entries --cluster-name my-cluster`**. El siguiente comando agrega una entrada de acceso para el rol de IAM del nodo. Reemplace *`111122223333`* por el ID de la Cuenta de AWS para el rol de IAM y *`myAmazonEKSNodeRole`* por el nombre del rol de su nodo. Si tiene un nodo de Windows, reemplace *`EC2_Linux`* por **`EC2_Windows`**. Asegúrese de especificar el ARN del rol de IAM correspondiente al nodo (y no el ARN del perfil de instancia).

```
aws eks create-access-entry --cluster-name my-cluster --principal-arn
 arn:aws:iam::111122223333:role/myAmazonEKSNodeRole --type EC2_Linux
```

Tiempo de espera de protocolo de enlace TLS

Cuando un nodo no puede establecer una conexión con el punto de conexión del servidor de la API público, podría ver un error similar al siguiente.

```
server.go:233] failed to run Kubelet: could not init cloud provider "aws": error
 finding instance i-1111f2222f333e44c: "error listing AWS instances: \"RequestError:
 send request failed\\ncaused by: Post  net/http: TLS handshake timeout\""
```

El proceso `kubelet` reaparecerá continuamente y probará el punto de conexión del servidor de API. El error también puede producirse temporalmente durante cualquier procedimiento que realice una actualización sucesiva del clúster en el plano de control, como un cambio de configuración o una actualización de versión.

Para resolver el problema, verifique la tabla de enrutamiento y los grupos de seguridad para asegurarse de que el tráfico de los nodos puede llegar al punto de conexión público.

InvalidClientTokenId

Si está utilizando roles de IAM para cuentas de servicio de un Pod o `DaemonSet` implementado en un clúster del Región de AWS de China y no ha establecido la variable de entorno `AWS_DEFAULT_REGION` en la especificación, el Pod o el `DaemonSet` pueden recibir el siguiente error:

```
An error occurred (InvalidClientTokenId) when calling the GetCallerIdentity operation:
 The security token included in the request is invalid
```

Para resolver el problema, debe agregar la variable de entorno `AWS_DEFAULT_REGION` a la especificación del Pod o `DaemonSet`, tal y como se muestra en el siguiente ejemplo de especificación de un Pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: envar-demo
  labels:
    purpose: demonstrate-envars
spec:
  containers:
```

```
  - name: envar-demo-container
    image: gcr.io/google-samples/node-hello:1.0
    env:
    - name: AWS_DEFAULT_REGION
      value: "region-code"
```

Vencimiento del certificado webhook de admisión de la VPC

Si el certificado utilizado para firmar el webhook de admisión de la VPC caduca, el estado de las nuevas implementaciones de Pod de Windows permanece en `ContainerCreating`.

Para resolver el problema si tiene compatibilidad con Windows heredado en el plano de datos, consulte Renovación del certificado de webhook de admisión de VPC. Si la versión del clúster y la plataforma son posteriores a una versión indicada en los Requisitos previos de la compatibilidad con Windows, recomendamos que elimine la compatibilidad con Windows heredado del plano de datos y lo habilite para el plano de control. Una vez que lo haga, no tiene que administrar el certificado de webhook. Para obtener más información, consulte Activación de la compatibilidad con Windows para su clúster de Amazon EKS.

Los grupos de nodos deben coincidir con la versión de Kubernetes antes de actualizar el plano de control

Antes de actualizar un plano de control con una nueva versión de Kubernetes, la versión secundaria de los nodos administrados y de Fargate en su clúster debe ser la misma que la de la versión actual de su plano de control. La API `update-cluster-version` de Amazon EKS rechazará las solicitudes hasta que usted actualice todos los nodos administrados de Amazon EKS a la versión del clúster actual. Amazon EKS proporciona las API para actualizar los nodos administrados. Para obtener información sobre la actualización de la versión de Kubernetes del grupo de nodos administrados, consulte Actualización de un grupo de nodos administrados. Para actualizar la versión de un nodo de Fargate, elimine el pod que representa el nodo y vuelva a implementar el pod después de actualizar el plano de control. Para obtener más información, consulte Actualización de una versión de Kubernetes de clúster de Amazon EKS.

Al lanzar muchos nodos, hay errores de **`Too Many Requests`**

Si lanza muchos nodos al mismo tiempo, es posible que vea un mensaje de error en Datos de usuario de Amazon EC2registros de ejecución que dice `Too Many Requests`. Esto puede ocurrir

porque el plano de control está sobrecargado con llamadas `describeCluster`. La sobrecarga da como resultado una limitación, es decir, que los nodos no ejecutan el script de arranque ni se unen por completo al clúster.

Asegúrese de que los argumentos `--apiserver-endpoint`, `--b64-cluster-ca` y `--dns-cluster-ip` se estén pasando al script de arranque del nodo. Al incluir estos argumentos, no es necesario que el script de arranque haga una llamada `describeCluster`, lo que ayuda a evitar que el plano de control se sobrecargue. Para obtener más información, consulte Proporcione datos de usuario a fin de pasar argumentos al archivo `bootstrap.sh` incluido con una AMI optimizada Linux/Bottlerocket para Amazon EKS.

Respuesta de error no autorizada HTTP 401 en solicitudes del servidor API de Kubernetes

Verá estos errores si el token de la cuenta de servicio de Pod caducó en un clúster.

El servidor de API de Kubernetes de los clústeres de Amazon EKS rechaza solicitudes con tokens de más de 90 días. En versiones anteriores de Kubernetes, los tokens no tenían caducidad. Esto significa que los clientes que confían en estos tokens deben actualizarlos en una hora. Para evitar que el servidor API de Kubernetes rechace su solicitud debido a un token no válido, la versión del SDK de cliente de Kubernetes utilizada por la carga de trabajo debe ser la misma o posterior a las siguientes versiones:

- Versión de Go `0.15.7` y posteriores
- Versión de Python `12.0.0` y posteriores
- Versión de Java `9.0.0` y posterior
- Versión de JavaScript `0.10.3` y posterior
- Rama de Ruby `master`
- Versión de Haskell `0.3.0.0`
- Versión C# `7.0.5` y posterior

Puede identificar todos los Pods existentes de tu clúster que utilizan tokens obsoletos. Para obtener más información, consulte Cuentas de servicio de Kubernetes.

La versión de la plataforma Amazon EKS está más de dos versiones por detrás de la versión de plataforma actual

Esto puede ocurrir cuando Amazon EKS no puede actualizar automáticamente la versión de la plataforma del clúster. Aunque hay muchas causas para esto, se presentan algunas de las causas más comunes. Si alguno de estos problemas se aplica a su clúster, es posible que siga funcionando. Amazon EKS no actualizará la versión de la plataforma.

Problema

El rol de IAM de clúster se eliminó: este rol se especificó cuando se creó el clúster. Puede ver qué rol se ha especificado con el siguiente comando. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-cluster --name my-cluster --query cluster.roleArn --output text | cut
 -d / -f 2
```

Un ejemplo de salida sería el siguiente.

```
eksClusterRole
```

Solución

Cree un nuevo rol de IAM de clúster con el mismo nombre.

Problema

Se eliminó una subred especificada durante la creación del clúster: las subredes que se usarían con el clúster se especificaron durante la creación del clúster. Puede ver qué subredes se especificaron con el siguiente comando. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-cluster --name my-cluster --query cluster.resourcesVpcConfig.subnetIds
```

Un ejemplo de salida sería el siguiente.

```
[
"subnet-EXAMPLE1",
"subnet-EXAMPLE2"
]
```

Solución

Confirme si los ID de subred existen en su cuenta.

```
vpc_id=$(aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.vpcId --output text)
aws ec2 describe-subnets --filters "Name=vpc-id,Values=$vpc_id" --query
 "Subnets[*].SubnetId"
```

Un ejemplo de salida sería el siguiente.

```
[
"subnet-EXAMPLE3",
"subnet-EXAMPLE4"
]
```

Si los ID de subredes devueltos en la salida no coinciden con los ID de subredes que se especificaron cuando se creó el clúster y desea que Amazon EKS actualice el clúster, debe cambiar las subredes utilizadas por el clúster. Esto se debe a que, si especificó más de dos subredes cuando creó el clúster, Amazon EKS seleccionará aleatoriamente las subredes que especificó para crear nuevas interfaces de red elástica en ellas. Estas interfaces de red permiten que el plano de control se comunique con los nodos. Amazon EKS no actualizará el clúster si la subred que selecciona no existe. No tiene control sobre las subredes que especificó en la creación del clúster en las que Amazon EKS elige crear una nueva interfaz de red.

Cuando inicia una actualización de la versión Kubernetes del clúster, la actualización puede fallar por el mismo motivo.

Problema

Se eliminó un grupo de seguridad especificado durante la creación del clúster: si especificó grupos de seguridad durante la creación del clúster, puede ver sus ID con el siguiente comando. Reemplace *my-cluster* por el nombre de su clúster.

```
aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.securityGroupIds
```

Un ejemplo de salida sería el siguiente.

```
[
```

```
    "sg-EXAMPLE1"
]
```

Si se devuelve `[]`, no se especificó ningún grupo de seguridad cuando se creó el clúster y el problema no es que falte un grupo de seguridad. Si se devuelven grupos de seguridad, confirme que los grupos de seguridad existen en su cuenta.

Solución

Confirme si estos grupos de seguridad existen en su cuenta.

```
vpc_id=$(aws eks describe-cluster --name my-cluster --query
 cluster.resourcesVpcConfig.vpcId --output text)
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=$vpc_id" --query
 "SecurityGroups[*].GroupId"
```

Un ejemplo de salida sería el siguiente.

```
[
"sg-EXAMPLE2"
]
```

Si los ID de grupos de seguridad devueltos en la salida no coinciden con los ID de grupos de seguridad que se especificaron cuando se creó el clúster y desea que Amazon EKS actualice el clúster, debe cambiar los grupos de seguridad utilizados por el clúster. Amazon EKS no actualizará un clúster si los ID de grupos de seguridad especificados en la creación del clúster no existen.

Cuando inicia una actualización de la versión Kubernetes del clúster, la actualización puede fallar por el mismo motivo.

Otros motivos por los que Amazon EKS no actualiza la versión de plataforma de su clúster

- No tiene al menos seis (aunque recomendamos 16) direcciones IP disponibles en cada una de las subredes que especificó al crear el clúster. Si no tiene suficientes direcciones IP disponibles en la subred, debe liberar direcciones IP en la subred o bien cambiar las subredes utilizadas por el clúster para que emplee subredes con suficientes direcciones IP disponibles.
- Ha habilitado cifrado de secretos cuando se creó el clúster y el AWS KMS se ha eliminado la clave especificada. Si desea que Amazon EKS actualice el clúster, deberá crear un clúster nuevo

Preguntas frecuentes sobre el estado de los clústeres y los códigos de error con rutas de resolución

Amazon EKS detecta problemas con los clústeres de EKS y la infraestructura del clúster y los almacena en el estado del clúster. Puede detectar, solucionar y abordar los problemas del clúster más rápido con la ayuda de la información sobre el estado del clúster. Esto le permite crear entornos de aplicación más seguros y actualizados. Además, es posible que no pueda actualizar a versiones más recientes de Kubernetes o que Amazon EKS no pueda instalar actualizaciones de seguridad en un clúster degradado debido a problemas con la infraestructura necesaria o la configuración del clúster. Amazon EKS puede tardar 3 horas en detectar problemas o detectar que se resolvió un problema.

El estado de un clúster de Amazon EKS es una responsabilidad compartida entre Amazon EKS y sus usuarios. Usted es responsable de la infraestructura previa de los roles de IAM y las subredes de Amazon VPC, así como de cualquier otra infraestructura necesaria que deba proporcionarse con antelación. Amazon EKS detecta cambios en la configuración de esta infraestructura y del clúster.

Para acceder al estado de su clúster en la consola de Amazon EKS, busque una sección llamada Problemas del estado en la pestaña Descripción general de la página de detalles del clúster de Amazon EKS. Estos datos también estarán disponibles al llamar a la acción `DescribeCluster` en la API de EKS, por ejemplo, desde AWS Command Line Interface.

¿Por qué debo utilizar esta característica?

> Obtendrá una mayor visibilidad del estado de su clúster de Amazon EKS, diagnosticará y solucionará rápidamente cualquier problema, sin necesidad de perder tiempo depurando o abriendo casos de asistencia de AWS. Por ejemplo: si accidentalmente eliminó una subred del clúster de Amazon EKS, Amazon EKS no podrá crear interfaces entre cuentas ni comandos de la AWS CLI de Kubernetes, tales como ejecuciones en `kubectl` o registros en `kubectl`. Estos fallarán con el error: `Error from server: error dialing backend: remote error: tls: internal error.` Ahora verá un problema de estado de Amazon EKS que dice: `subnet-da60e280 was deleted: could not create network interface`.

¿Cómo se relaciona o funciona esta característica con otros servicios de AWS?

> Los roles de IAM y las subredes de Amazon VPC son dos ejemplos de infraestructura previa donde el estado del clúster detecta problemas. Esta característica devolverá información detallada si esos recursos no están configurados correctamente.

¿Se cobran cargos por un clúster con problemas de estado?

Sí, todos los clústeres de Amazon EKS se facturan al precio estándar de Amazon EKS. La característica estado del clúster está disponible sin costo adicional.

¿Funciona esta característica con los clústeres de Amazon EKS en AWS Outposts?

Sí, se detectan problemas con los clústeres de EKS en la Nube de AWS, incluidos los clústeres extendidos en AWS Outposts y los clústeres locales en AWS Outposts. El estado del clúster no detecta problemas con Amazon EKS Anywhere o Amazon EKS Distro (EKS-D).

¿Puedo recibir una notificación cuando se detecten nuevos problemas?

No, debe revisar la Consola de Amazon EKS o llamar a la API `DescribeCluster` de EKS.

¿La consola me avisa en caso de problemas de estado?

Sí, cualquier clúster con problemas de estado incluirá un cartel en la parte superior de la consola.

Las primeras dos columnas son las que se necesitan para los valores de respuesta de la API. El tercer campo del objeto Problemas en el estado del clúster es resourceIds, cuya devolución depende del tipo de problema.

Código	Mensaje	ResourceIds	¿Clúster recuperable?	
SUBNET_NOT_FOUND	No logramos encontrar una o más subredes actualmente asociadas a su clúster. Llame a la API update-cluster-config de Amazon EKS para actualizar las subredes.	ID de subred	Sí	
SECURITY_GROUP_NOT_FOUND	No logramos encontrar uno o más grupos de seguridad actualmente asociados a su clúster. Llame a la API update-cluster-config de Amazon EKS para actualizar los grupos de seguridad	ID de grupos de seguridad	Sí	

Código	Mensaje	ResourceIds	¿Clúster recuperable?	
IP_NOT_AVAILABLE	Una o varias de las subredes asociadas a su clúster no tienen suficientes direcciones IP disponibles para que Amazon EKS realice operaciones de administración del clúster. Libere direcciones en las subredes o asocie diferentes subredes a su clúster mediante la API update-cluster-config de Amazon EKS.	ID de subred	Sí	
VPC_NOT_FOUND	No logramos encontrar la VPC asociada a su clúster. Debe eliminar y volver a crear su clúster.	ID de la VPC	No	
ASSUME_ROLE_ACCESS_DENIED	Su clúster no utiliza el rol vinculado al servicio de Amazon EKS. No logramos asumir el rol asociado a su clúster para realizar las operaciones de administración necesarias de Amazon EKS. Compruebe si el rol existe y tiene la política de confianza requerida.	El rol de IAM del clúster	Sí	

Código	Mensaje	ResourceIds	¿Clúster recuperable?	
PERMISSION_ACCESS_DENIED	Su clúster no utiliza el rol vinculado al servicio de Amazon EKS. El rol asociado a su clúster no otorga permisos suficientes para que Amazon EKS lleve a cabo las operaciones de administración requeridas. Compruebe las políticas asociadas al rol del clúster y si se aplica alguna política de denegación independiente.	El rol de IAM del clúster	Sí	
ASSUME_ROLE_ACCESS_DENIED_USING_SLR	No logramos asumir el rol vinculado al servicio de administración del clúster de Amazon EKS. Compruebe si el rol existe y tiene la política de confianza requerida.	El rol vinculado al servicio de Amazon EKS	Sí	
PERMISSION_ACCESS_DENIED_USING_SLR	El rol vinculado al servicio de administración del clúster de Amazon EKS no concede permisos suficientes para que Amazon EKS lleve a cabo las operaciones de administración requeridas. Compruebe las políticas asociadas al rol del clúster y si se aplica alguna política de denegación independiente.	El rol vinculado al servicio de Amazon EKS	Sí	

Código	Mensaje	ResourceIds	¿Clúster recuperable?	
OPT_IN_REQUIRED	Su cuenta no tiene una suscripción al servicio Amazon EC2. Actualice las suscripciones de su cuenta en la página de configuración de su cuenta.	N/A	Sí	
STS_REGIONAL_ENDPOINT_DISABLED	El punto de conexión regional de STS está deshabilitado. Habilite el punto de conexión para que Amazon EKS lleve a cabo las operaciones de administración del clúster necesarias.	N/A	Sí	
KMS_KEY_DISABLED	La clave de AWS KMS asociada a su clúster está deshabilitada. Vuelva a habilitar la clave para recuperar su clúster.	Con la KMS Key Arn	Sí	
KMS_KEY_NOT_FOUND	No logramos encontrar la clave de AWS KMS asociada a su clúster. Debe eliminar y volver a crear el clúster.	Con la KMS Key ARN	No	
KMS_GRANT_REVOKED	Se revocan las concesiones de la clave de AWS KMS asociada al clúster. Debe eliminar y volver a crear el clúster.	Con la KMS Key Arn	No	

Amazon EKS Connector

Puede utilizar Amazon EKS Connector a fin de registrar y conectar cualquier clúster de Kubernetes conforme a AWS y visualizarlo en la consola de Amazon EKS. Una vez conectado, puede ver el estado, la configuración y las cargas de trabajo del clúster en la consola de Amazon EKS. Puede utilizar esta característica para ver los clústeres conectados en la consola de Amazon EKS, pero no puede administrarlos. Amazon EKS Connector requiere un agente que sea un proyecto de código abierto en GitHub. Para obtener contenido técnico adicional, incluidas las preguntas frecuentes y las soluciones de problemas, consulte Solución de problemas en Amazon EKS Connector.

Amazon EKS Connector puede conectar los siguientes tipos de clústeres de Kubernetes a Amazon EKS.

- Clústeres de Kubernetes en las instalaciones
- Clústeres autoadministrados que se ejecutan en Amazon EC2
- Clústeres administrados de otros proveedores de nube

Consideraciones sobre Amazon EKS Connector

Antes de utilizar Amazon EKS Connector, debe comprender lo siguiente:

- Debe tener privilegios administrativos en el clúster de Kubernetes para conectar el clúster a Amazon EKS.
- El clúster de Kubernetes debe tener nodos de trabajo Linux de 64 bits (x86) presentes antes de conectarse. No se admiten los nodos de trabajo de ARM.
- Debe tener nodos de trabajo en el clúster de Kubernetes que tengan acceso saliente a los puntos de conexión `ssm.` y `ssmmessages.` de Systems Manager. Para obtener más información, consulte Puntos de enlace de Systems Manager en la Referencia general de AWS.
- De forma predeterminada, puede conectar hasta 10 clústeres en una región. Puede solicitar un aumento a través de la consola de cuotas de servicio. Para obtener más información, consulte Solicitud de un aumento de cuota.
- Solo se admiten las API `RegisterCluster`, `ListClusters`, `DescribeCluster` y `DeregisterCluster` de Amazon EKS para clústeres de Kubernetes externos.
- Debe tener los siguientes permisos para registrar un clúster:

  - eks:RegisterCluster
  - ssm:CreateActivation
  - ssm:DeleteActivation
  - iam:PassRole
- Debe tener los siguientes permisos para anular el registro de un clúster:
  - eks:DeregisterCluster
  - ssm:DeleteActivation
  - ssm:DeregisterManagedInstance

Roles de IAM necesarios para Amazon EKS Connector

El uso de Amazon EKS Connector requiere los dos roles de IAM siguientes:

- El rol vinculado al servicio de Amazon EKS Connector se crea al registrar un clúster por primera vez.
- Debe crear el rol de IAM de agente de Amazon EKS Connector. Para obtener más información, consulte Rol de IAM conector de Amazon EKS.

A fin de habilitar el permiso de visualización de clúster y carga de trabajo para entidades principales de IAM, aplique el `eks-connector` y los roles de clúster de Amazon EKS Connector en el clúster. Siga los pasos de Concesión de acceso a una entidad principal de IAM para ver recursos de Kubernetes en un clúster.

Conexión de un clúster externo

Puede conectar un clúster externo de Kubernetes en Amazon EKS mediante varios métodos en el siguiente proceso. Este proceso consta de dos pasos: registrar el clúster en Amazon EKS e instalar el agente `eks-connector` del clúster.

> **Important**
>
> Debe completar el segundo paso en un plazo de 3 días a partir de haber completado el primer paso, antes de que caduque el registro.

Métodos del conector

No todos los métodos para instalar el agente se pueden usar después de cada uno de los métodos para registrar el clúster. En la siguiente tabla se enumeran cada uno de los métodos de registro y los métodos que se pueden utilizar para instalar el agente.

Paso	Métodos		
Registrar el clúster	AWS Management Console	AWS Command Line Interface	`eksctl`
Instalación del agente de	Manifiestos de Helm y YAML	Manifiestos de Helm y YAML	Manifiestos de YAML

Requisitos previos

- Asegúrese de que se haya creado el rol de agente de Amazon EKS Connector. Siga los pasos de Creación del rol de agente conector de Amazon EKS.
- Debe tener los siguientes permisos para registrar un clúster:
  - `eks:RegisterCluster`
  - `ssm:CreateActivation`
  - `ssm:DeleteActivation`
  - `iam:PassRole`

Paso 1: registro del clúster

AWS CLI

Requisitos previos

- La AWS CLI debe estar instalada. Para instalarla o actualizarla, consulte Instalación de la AWS CLI.

Para registrar el clúster con la AWS CLI

- Para la configuración de Connector, especifique su rol de IAM de agente de Amazon EKS Connector. Para obtener más información, consulte Roles de IAM necesarios para Amazon EKS Connector.

```
aws eks register-cluster \
     --name my-first-registered-cluster \
     --connector-config roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider="OTHER" \
     --region aws-region
```

Un ejemplo de salida sería el siguiente.

```
{
    "cluster": {
        "name": "my-first-registered-cluster",
        "arn": "arn:aws:eks:region:111122223333:cluster/my-first-registered-
cluster",
        "createdAt": 1627669203.531,
        "ConnectorConfig": {
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": 1627672543.0,
            "provider": "OTHER",
            "roleArn": "arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole"
        },
        "status": "CREATING"
    }
}
```

Utilizará los valores de `aws-region`, `activationId` y `activationCode` en un paso posterior.

AWS Management Console

Para registrar el clúster de Kubernetes en la consola.

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.

2. Elija Add cluster (Agregar clúster) y seleccione Register (Registrar) para abrir la página de configuración.
3. En la sección Configure cluster (Configurar clúster), rellene los siguientes campos:
   - Nombre: un nombre único para el clúster.
   - Provider (Proveedor): elija esta opción para mostrar la lista desplegable de proveedores de clústeres de Kubernetes. Si no conoce al proveedor específico, seleccione Otro.
   - EKS Connector role (Rol de EKS Connector): seleccione el rol que se va a utilizar para conectar el clúster.
4. Seleccione Register cluster (Registrar el clúster).
5. Se abre la página de información general del clúster. Si quiere usar el gráfico de Helm, copie el comando `helm install` y continúe con el siguiente paso. Si desea usar el manifiesto de YAML, elija Descargar archivo YAML para descargar el archivo de manifiesto en la unidad local.

   > **Important**
   >
   > - Esta es su única oportunidad para copiar el comando `helm install` o descargar este archivo. No navegue fuera de esta página, ya que no se podrá acceder al enlace y deberá anular el registro del clúster e iniciar los pasos desde el principio.
   > - El comando o el archivo de manifiesto solo se puede utilizar una vez para el clúster registrado. Si elimina recursos del clúster de Kubernetes, debe volver a registrar el clúster y obtener un nuevo archivo de manifiesto.

   Continúe en el paso siguiente para aplicar el archivo de manifiesto al clúster de Kubernetes.

eksctl

Requisitos previos

- Se debe instalar la versión `0.68` de `eksctl` o posterior. Para instalarla o actualizarla, consulte Introducción a Amazon EKS: `eksctl`.

Para registrar el clúster con **eksctl**

1. Registre el clúster al proporcionar un nombre, un proveedor y una región.

```
eksctl register cluster --name my-cluster --provider my-provider --
region region-code
```

Ejemplo de salida:

```
2021-08-19 13:47:26 [#]  creating IAM role "eksctl-20210819194112186040"
2021-08-19 13:47:26 [#]  registered cluster "<name>" successfully
2021-08-19 13:47:26 [#]  wrote file eks-connector.yaml to <current directory>
2021-08-19 13:47:26 [#]  wrote file eks-connector-clusterrole.yaml to <current
 directory>
2021-08-19 13:47:26 [#]  wrote file eks-connector-console-dashboard-full-access-
group.yaml to <current directory>
2021-08-19 13:47:26 [!]  note: "eks-connector-clusterrole.yaml" and "eks-
connector-console-dashboard-full-access-group.yaml" give full EKS Console access
 to IAM identity "<aws-arn>", edit if required; read https://eksctl.io/usage/
eks-connector for more info
2021-08-19 13:47:26 [#]  run `kubectl apply -f eks-connector.yaml,eks-connector-
clusterrole.yaml,eks-connector-console-dashboard-full-access-group.yaml` before
 expiry> to connect the cluster
```

De este modo, se crean archivos en el equipo local. Estos archivos deben aplicarse al clúster externo en un plazo de 3 días o el registro vence.

2. En una terminal que pueda acceder al clúster, aplique el archivo `eks-connector-binding.yaml`:

```
kubectl apply -f eks-connector-binding.yaml
```

Paso 2: Instalar el agente de **eks-connector**

Helm chart

1. Si utilizó la AWS CLI en el paso anterior, sustituya el `ACTIVATION_CODE` y `ACTIVATION_ID` en el siguiente comando por los valores de `activationId` y `activationCode`,

respectivamente. Sustituya el `aws-region` por el Región de AWS que utilizó en el paso anterior. Ejecute el siguiente comando para instalar el agente de `eks-connector` en el clúster del registro:

```
$ helm install eks-connector \
  --namespace eks-connector \
  oci://public.ecr.aws/eks-connector/eks-connector-chart \
  --set eks.activationCode=ACTIVATION_CODE \
  --set eks.activationId=ACTIVATION_ID \
  --set eks.agentRegion=aws-region
```

Si ha utilizado la AWS Management Console en el paso anterior, utilice el comando que copió del paso anterior y que contiene estos valores rellenados.

2. Compruebe el buen estado de la implementación de `eks-connector` instalado y espere a que el estado del clúster registrado en Amazon EKS sea `ACTIVE`.

YAML manifest

Complete la conexión aplicando el archivo de manifiesto de Amazon EKS Connector al clúster de Kubernetes. Para ello, debe utilizar los métodos descritos anteriormente. Si el manifiesto no se aplica en un plazo de tres días, el registro de Amazon EKS Connector vence. Si la conexión del clúster vence, debe anularse el registro del clúster antes de volverlo a conectar.

1. Descargue el archivo YAML de Amazon EKS Connector.

```
curl -O https://amazon-eks.s3.us-west-2.amazonaws.com/eks-connector/manifests/
eks-connector/latest/eks-connector.yaml
```

2. Edite el archivo YAML de Amazon EKS Connector para reemplazar todas las referencias de `%AWS_REGION%`, `%EKS_ACTIVATION_ID%` y `%EKS_ACTIVATION_CODE%` por `aws-region`, `activationId` y `activationCode` de la salida del paso anterior.

   El siguiente comando de ejemplo puede sustituir estos valores.

```
sed -i "s~%AWS_REGION%~$aws-region~g; s~%EKS_ACTIVATION_ID
%~$EKS_ACTIVATION_ID~g; s~%EKS_ACTIVATION_CODE%~$(echo -n $EKS_ACTIVATION_CODE |
 base64)~g" eks-connector.yaml
```

> **Important**
>
> Asegúrese de que el código de activación esté en el formato base64.

3. En una terminal que pueda acceder al clúster, puede aplicar el archivo de manifiesto actualizado al ejecutar el siguiente comando:

```
kubectl apply -f eks-connector.yaml
```

4. Una vez que los archivos YAML de manifiesto y de enlace de roles de Amazon EKS Connector se hayan aplicado al clúster de Kubernetes, confirme que el clúster esté conectado.

```
aws eks describe-cluster \
     --name "my-first-registered-cluster" \
     --region AWS_REGION
```

   El resultado debe incluir `status=ACTIVE`.

5. (Opcional) Agregue etiquetas a su clúster. Para obtener más información, consulte Etiquetado de los recursos de Amazon EKS.

Siguientes pasos

Si tiene algún problema con estos pasos, consulte Solución de problemas en Amazon EKS Connector.

Para conceder acceso a otras entidades principales de IAM a la consola de Amazon EKS para ver los recursos de Kubernetes en un clúster conectado, consulte Concesión de acceso a una entidad principal de IAM para ver recursos de Kubernetes en un clúster.

Concesión de acceso a una entidad principal de IAM para ver recursos de Kubernetes en un clúster

Otorgue acceso a la consola de Amazon EKS a las entidades principales de IAM para ver la información sobre los recursos de Kubernetes que se ejecutan en el clúster conectado.

Requisitos previos

La entidad principal de IAM que utiliza para acceder a la AWS Management Console debe cumplir con los siguientes requisitos:

- Debe tener el permiso de IAM `eks:AccessKubernetesApi`.
- La cuenta del servicio de Amazon EKS Connector puede suplantar la entidad principal de IAM en el clúster. Esto permite que Amazon EKS Connector asigne la entidad principal de IAM a un usuario de Kubernetes.

Para crear y aplicar el rol de clúster de Amazon EKS Connector

1. Descargue la plantilla de rol de clúster `eks-connector`.

   ```
   curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/eks-connector/manifests/eks-connector-console-roles/eks-connector-clusterrole.yaml
   ```

2. Edite el archivo YAML de la plantilla de roles del clúster. Sustituya las referencias de `%IAM_ARN %` por el nombre de recurso de Amazon (ARN) de la entidad principal de IAM.
3. Aplique el archivo YAML de rol de clúster de Amazon EKS Connector al clúster de Kubernetes.

   ```
   kubectl apply -f eks-connector-clusterrole.yaml
   ```

Para que una entidad principal de IAM visualice los recursos de Kubernetes en la consola de Amazon EKS, la entidad de principal debe estar asociada a un `role` o `clusterrole` de Kubernetes con los permisos necesarios para leer estos recursos. Para obtener más información, consulte Uso de la autorización de RBAC en la documentación de Kubernetes.

Para configurar una entidad principal de IAM para que acceda al clúster conectado

1. Puede descargar uno de estos archivos de manifiesto de ejemplo para crear un `clusterrole` y `clusterrolebinding` o un `role` y `rolebinding`, respectivamente:

   Ver recursos de Kubernetes en todos los espacios de nombres

   El rol del clúster de `eks-connector-console-dashboard-full-access-clusterrole` da acceso a todos los espacios de nombres y recursos que se pueden visualizar en la consola. Puede cambiar el nombre de `role`, `clusterrole` y su enlace

correspondiente antes de aplicarlos al clúster. Utilice el siguiente comando para descargar un archivo de muestra.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/eks-connector/manifests/eks-connector-console-roles/eks-connector-console-dashboard-full-access-group.yaml
```

Ver recursos de Kubernetes en un espacio de nombres específico

El espacio de nombres de este archivo es `default`, por lo que, si desea especificar un espacio de nombres diferente, edite el archivo antes de aplicarlo al clúster. Utilice el siguiente comando para descargar un archivo de muestra.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/eks-connector/manifests/eks-connector-console-roles/eks-connector-console-dashboard-restricted-access-group.yaml
```

2. Edite el archivo YAML de acceso completo o acceso restringido para reemplazar las referencias de `%IAM_ARN%` por el nombre de recurso de Amazon (ARN) de la entidad principal de IAM.
3. Aplique los archivos YAML de acceso completo o acceso restringido a su clúster de Kubernetes. Reemplace el valor del archivo YAML por el suyo propio.

```
kubectl apply -f eks-connector-console-dashboard-full-access-group.yaml
```

Para ver los recursos de Kubernetes en el clúster conectado, consulte Vea los recursos de Kubernetes. Datos de algunos tipos de recursos de la pestaña Resources(Recursos) no está disponible para clústeres conectados.

Anulación del registro de un clúster

Si ya ha terminado de usar un clúster conectado, puede anular su registro. Una vez anulado el registro, el clúster ya no estará visible en la consola de Amazon EKS.

Debe tener los siguientes permisos para llamar a la API deregisterCluster:

- `eks:DeregisterCluster`
- `ssm:DeleteActivation`
- `ssm:DeregisterManagedInstance`

Este proceso consta de dos pasos: anular el registro del clúster en Amazon EKS y desinstalar el agente eks-connector del clúster.

Para anular el registro del clúster de Kubernetes

AWS CLI

Requisitos previos

- La AWS CLI debe estar instalada. Para instalarla o actualizarla, consulte Instalación de la AWS CLI.
- Asegúrese de que se haya creado el rol de agente de Amazon EKS Connector.

Anule el registro del clúster conectado.

```
aws eks deregister-cluster \
    --name my-cluster \
    --region region-code
```

AWS Management Console

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
2. Seleccione Clusters (Clústeres).
3. En la página Clusters (Clústeres), seleccione el clúster conectado y seleccione Deregister (Anular registro).
4. Confirme que desea anular el registro del clúster.

eksctl

Requisitos previos

- Se debe instalar la versión `0.68` de `eksctl` o posterior. Para instalarla o actualizarla, consulte Introducción a Amazon EKS: `eksctl`.
- Asegúrese de que se haya creado el rol de agente de Amazon EKS Connector.

Para anular el registro del clúster con **eksctl**

- Para la configuración de Connector, especifique su rol de IAM de agente de Amazon EKS Connector. Para obtener más información, consulte Roles de IAM necesarios para Amazon EKS Connector.

```
eksctl deregister cluster --name my-cluster
```

Para limpiar los recursos del clúster de Kubernetes

Helm

- Ejecute el siguiente comando para desinstalar el agente.

```
helm -n eks-connector uninstall eks-connector
```

YAML manifest

1. Elimine el archivo YAML de Amazon EKS Connector del clúster de Kubernetes.

```
kubectl delete -f eks-connector.yaml
```

2. Si ha creado un `clusterrole` o `clusterrolebindings` para que entidades principales de IAM adicionales accedan al clúster, asegúrese de eliminarlas del clúster de Kubernetes.

Solución de problemas en Amazon EKS Connector

En este tema, se tratan algunos de los errores comunes que pueden producirse mientras se usa Amazon EKS Connector y se incluyen las instrucciones sobre cómo resolverlos e implementar soluciones alternativas.

Solución de problemas básicos

En esta sección, se describen los pasos para diagnosticar el problema si no está claro.

Verificación del estado de Amazon EKS Connector

Compruebe el estado de Amazon EKS Connector.

```
kubectl get pods -n eks-connector
```

Inspección de los registros de Amazon EKS Connector

El Pod de Amazon EKS Connector consta de tres contenedores. Para recuperar los registros completos de todos estos contenedores y poder inspeccionarlos, ejecute los siguientes comandos:

- `connector-init`

```
kubectl logs eks-connector-0 --container connector-init -n eks-connector
kubectl logs eks-connector-1 --container connector-init -n eks-connector
```

- `connector-proxy`

```
kubectl logs eks-connector-0 --container connector-proxy -n eks-connector
kubectl logs eks-connector-1 --container connector-proxy -n eks-connector
```

- `connector-agent`

```
kubectl exec eks-connector-0 --container connector-agent -n eks-connector -- cat /
var/log/amazon/ssm/amazon-ssm-agent.log
kubectl exec eks-connector-1 --container connector-agent -n eks-connector -- cat /
var/log/amazon/ssm/amazon-ssm-agent.log
```

Obtención del nombre real del clúster

Los clústeres de Amazon EKS se identifican de forma inequívoca a través del `clusterName` en una cuenta y Región de AWS de AWS. Si tiene varios clústeres conectados en Amazon EKS, puede confirmar qué clúster de Amazon EKS se ha registrado en el clúster de Kubernetes actual. Para ello, ingrese lo siguiente para saber cuál es el `clusterName` del clúster actual.

```
kubectl exec eks-connector-0 --container connector-agent -n eks-connector \
  -- cat /var/log/amazon/ssm/amazon-ssm-agent.log | grep -m1 -oE "eks_c:[a-zA-Z0-9_-]+"
 | sed -E "s/^.*eks_c:([a-zA-Z0-9_-]+)_[a-zA-Z0-9]+.*$/\1/"
kubectl exec eks-connector-1 --container connector-agent -n eks-connector \
  -- cat /var/log/amazon/ssm/amazon-ssm-agent.log | grep -m1 -oE "eks_c:[a-zA-Z0-9_-]+"
 | sed -E "s/^.*eks_c:([a-zA-Z0-9_-]+)_[a-zA-Z0-9]+.*$/\1/"
```

Comandos diversos

Los siguientes comandos son útiles para recuperar la información que necesita para solucionar problemas.

- Utilice el siguiente comando para recopilar imágenes que utilizan los Pods en Amazon EKS Connector.

```
kubectl get pods -n eks-connector -o jsonpath="{.items[*].spec.containers[*].image}"
 | tr -s '[[:space:]]' '\n'
```

- Utilice el siguiente comando para determinar los nombres de los nodos en los que se ejecuta Amazon EKS Connector.

```
kubectl get pods -n eks-connector -o jsonpath="{.items[*].spec.nodeName}" | tr -s
 '[[:space:]]' '\n'
```

- Ejecute el siguiente comando para obtener las versiones de cliente y servidor de Kubernetes.

```
kubectl version
```

- Ejecute el siguiente comando para obtener información acerca de los nodos.

```
kubectl get nodes -o wide --show-labels
```

Problema de Helm: 403 Prohibido

Si ha recibido el siguiente error al ejecutar los comandos de instalación de Helm:

```
Error: INSTALLATION FAILED: unexpected status from HEAD request to https://
public.ecr.aws/v2/eks-connector/eks-connector-chart/manifests/0.0.6: 403 Forbidden
```

Puede ejecutar la siguiente línea para solucionarlo:

```
docker logout public.ecr.aws
```

Error de la consola: El clúster está atascado en el estado Pending

Si, después de registrar el clúster, este se queda atascado en el estado `Pending` en la consola de Amazon EKS, tal vez se deba a que Amazon EKS Connector todavía no conectó correctamente el clúster a AWS. Para un clúster registrado, el estado `Pending` significa que la conexión todavía no se ha establecido correctamente. Para resolver este problema, asegúrese de haber aplicado el manifiesto al clúster de Kubernetes de destino. Si lo aplicó al clúster, pero este sigue en estado `Pending`, puede que StatefulSet `eks-connector` esté en mal estado. Para solucionar este problema, consulte Los Pods de Amazon EKS Connector se están bloqueando en bucle en este tema.

Error de consola: **`User “system:serviceaccount:eks-connector:eks-connector” can't impersonate resource “users” in API group “”`** en el ámbito del clúster

Amazon EKS Connector utiliza la suplantación de usuarios de Kubernetes para actuar en nombre de las entidades principales de IAM desde la AWS Management Console. A cada entidad principal de IAM que accede a la API de Kubernetes desde la cuenta de servicio de AWS `eks-connector` se le debe conceder permiso para suplantar al usuario de Kubernetes correspondiente con un ARN de IAM como nombre de usuario de Kubernetes. En los siguientes ejemplos, el ARN de IAM se asigna a un usuario de Kubernetes.

- El usuario de IAM *`john`* de la cuenta AWS de *`111122223333`* está asignado a un usuario de Kubernetes. Según las prácticas recomendadas de IAM, se recomienda conceder permisos a los roles en lugar de a los usuarios.

  ```
  arn:aws:iam::111122223333:user/john
  ```

- El rol de IAM *`admin`* de la cuenta AWS de *`111122223333`* está asignado a un usuario de Kubernetes:

  ```
  arn:aws:iam::111122223333:role/admin
  ```

  El resultado es el ARN de un rol de IAM, en lugar del ARN de la sesión de AWS STS.

Para obtener instrucciones sobre cómo configurar `ClusterRole` y `ClusterRoleBinding` a fin de conceder a la cuenta de servicio de `eks-connector` el privilegio para suplantar al usuario asignado,

consulte Concesión de acceso a una entidad principal de IAM para ver recursos de Kubernetes en un clúster. Asegúrese de que, en la plantilla, `%IAM_ARN%` se sustituya con el ARN de IAM de la entidad principal de IAM de la AWS Management Console.

Error de consola: **`[...] is forbidden: User [...] cannot list resource "[...] in API group"`** en el ámbito del clúster

Considere el siguiente problema. Amazon EKS Connector ha suplantado correctamente a la entidad principal de IAM solicitante de la AWS Management Console en el clúster de Kubernetes de destino. Sin embargo, la entidad principal suplantada no tiene permiso de RBAC en las operaciones de la API de Kubernetes.

Para resolver este problema, existen dos métodos para conceder permisos a usuarios adicionales. Si anteriormente instaló eks-connector mediante el gráfico de Helm, puede conceder acceso a los usuarios fácilmente ejecutando el siguiente comando. Sustituya `userARN1` y `userARN2` con una lista de los ARN de los roles de IAM para permitir el acceso a la visualización de los recursos de Kubernetes:

```
helm upgrade eks-connector oci://public.ecr.aws/eks-connector/eks-connector-chart \
    --reuse-values \
    --set 'authentication.allowedUserARNs={userARN1,userARN2}'
```

O bien, como administrador del clúster, otorgue el nivel adecuado de privilegios de RBAC a los usuarios individuales de Kubernetes. Para obtener más información y ejemplos, consulte Concesión de acceso a una entidad principal de IAM para ver recursos de Kubernetes en un clúster.

Error de la consola: Amazon EKS no puede comunicarse con el servidor de la API del clúster de Kubernetes. El clúster debe estar en estado ACTIVE (ACTIVO) para que la conexión se establezca de forma correcta. Intente establecer la conexión en unos minutos.

Si el servicio Amazon EKS no puede comunicarse con Amazon EKS Connector en el clúster de destino, tal vez se deba a alguno de los siguientes motivos:

- Amazon EKS Connector no funciona bien en el clúster de destino.
- La conectividad es deficiente o la conexión se ha interrumpido entre el clúster de destino y la Región de AWS.

Para resolver este problema, consulte los registros de Amazon EKS Connector. Si no aparece un error para Amazon EKS Connector, vuelva a intentar establecer la conexión después de unos minutos. Si experimenta regularmente una alta latencia o una conectividad intermitente para el clúster de destino, considere volver a registrar el clúster en una Región de AWS que se encuentre en una ubicación más cercana.

Los Pods de Amazon EKS Connector se están bloqueando en bucle

Hay muchas razones que pueden provocar que un Pod de Amazon EKS Connector entre en el estado `CrashLoopBackOff`. Es probable que este problema afecte al contenedor `connector-init`. Compruebe el estado del Pod conector de Amazon EKS.

```
kubectl get pods -n eks-connector
```

Un ejemplo de salida sería el siguiente.

```
NAME              READY   STATUS                  RESTARTS   AGE
eks-connector-0   0/2     Init:CrashLoopBackOff   1          7s
```

Si el resultado es similar al resultado anterior, consulte Inspección de los registros de Amazon EKS Connector para solucionar el problema.

`Failed to initiate eks-connector: InvalidActivation`

Cuando inicia Amazon EKS Connector por primera vez, este registra un `activationId` y un `activationCode` en Amazon Web Services. El registro puede fallar, lo que podría provocar que el contenedor `connector-init` se bloquee con un error similar al siguiente.

```
F1116 20:30:47.261469       1 init.go:43] failed to initiate eks-connector:
 InvalidActivation:
```

Para solucionar este problema, tenga en cuenta las siguientes causas y correcciones recomendadas:

- Es posible que el registro haya fallado porque el `activationId` y el `activationCode` no están en el archivo de manifiesto. En este caso, asegúrese de que se devolvieron los valores correctos de la operación `RegisterCluster` de la API y de que el `activationCode` se encuentra en el archivo de manifiesto. El `activationCode` se agrega a los secretos de Kubernetes, por lo que debe estar codificado en `base64`. Para obtener más información, consulte Paso 1: registro del clúster.

- Es posible que el registro haya fallado porque la activación venció. Esto se debe a que, por razones de seguridad, debe activar Amazon EKS Connector en un plazo de tres días después del registro del clúster. Para resolver este problema, asegúrese de haber aplicado el manifiesto de Amazon EKS Connector al clúster de Kubernetes de destino antes de la fecha y la hora de vencimiento. Para confirmar la fecha de vencimiento de la activación, llame a la operación de la API `DescribeCluster`.

  ```
  aws eks describe-cluster --name my-cluster
  ```

  En la siguiente respuesta de ejemplo, la fecha y la hora de vencimiento se registran como `2021-11-12T22:28:51.101000-08:00`.

  ```
  {
      "cluster": {
          "name": "my-cluster",
          "arn": "arn:aws:eks:region:111122223333:cluster/my-cluster",
          "createdAt": "2021-11-09T22:28:51.449000-08:00",
          "status": "FAILED",
          "tags": {
          },
          "connectorConfig": {
              "activationId": "00000000-0000-0000-0000-000000000000",
              "activationExpiry": "2021-11-12T22:28:51.101000-08:00",
              "provider": "OTHER",
              "roleArn": "arn:aws:iam::111122223333:role/my-connector-role"
          }
      }
  }
  ```

  Si `activationExpiry` ya transcurrió, anule el registro del clúster y vuelva a registrarlo. Hacer esto genera una nueva activación.

El nodo del clúster no tiene conectividad de salida

Para que funcione correctamente, Amazon EKS Connector requiere conectividad de salida a varios puntos de conexión de AWS. No se puede conectar un clúster privado sin conectividad de salida a una Región de AWS de destino. Para resolver este problema, debe agregar la conectividad

de salida necesaria. Para obtener información sobre los requisitos de los conectores, consulte Consideraciones sobre Amazon EKS Connector.

Los Pods conectores de Amazon EKS están en estado `ImagePullBackOff`

Si ejecuta el comando `get pods` y los Pods se encuentran en el estado `ImagePullBackOff`, no pueden funcionar correctamente. Si los Pods de Amazon EKS Connector se encuentran en estado `ImagePullBackOff`, no pueden funcionar correctamente. Compruebe el estado de los Pods de Amazon EKS Connector.

```
kubectl get pods -n eks-connector
```

Un ejemplo de salida sería el siguiente.

```
NAME              READY   STATUS                  RESTARTS   AGE
eks-connector-0   0/2     Init:ImagePullBackOff   0          4s
```

El archivo de manifiesto predeterminado de Amazon EKS Connector hace referencia a imágenes de Galería pública de Amazon ECR. Es posible que el clúster de Kubernetes de destino no pueda extraer imágenes de Galería pública de Amazon ECR. Resuelva el problema de extracción de imágenes de Galería pública de Amazon ECR o considere la posibilidad de reflejar las imágenes en el registro de contenedores privado de su elección.

Preguntas frecuentes

P: ¿Cómo funciona la tecnología subyacente detrás de Amazon EKS Connector?

R: Amazon EKS Connector se basa en el agente de AWS Systems Manager (Systems Manager). Amazon EKS Connector se ejecuta como `StatefulSet` en el clúster de Kubernetes. Establece una conexión y actúa como proxy para la comunicación entre el servidor de la API de su clúster y Amazon Web Services. Lo hace para mostrar los datos del clúster en la consola de Amazon EKS hasta que desconecte el clúster de AWS. El agente de Systems Manager es un proyecto de código abierto. Para obtener más información acerca de este proyecto, consulte la página de proyecto de GitHub.

P: Tengo un clúster de Kubernetes en las instalaciones que quiero conectar. ¿Debo abrir los puertos del firewall para conectarlo?

R: No, no es necesario abrir ningún puerto de firewall. El clúster de Kubernetes solo requiere conexión de salida con las Regiones de AWS. Los servicios de AWS nunca acceden a los recursos en la red en las instalaciones. Amazon EKS Connector se ejecuta en el clúster e inicia la conexión con AWS. Cuando se completa el registro del clúster, AWS solo emite comandos para Amazon EKS Connector después de que usted inicie una acción desde la consola de Amazon EKS que requiere información del servidor de la API de Kubernetes en el clúster.

P: ¿Qué datos envía Amazon EKS Connector desde mi clúster a AWS?

R: Amazon EKS Connector envía la información técnica necesaria para que su clúster se registre en AWS. También envía metadatos del clúster y la carga de trabajo para las características de la consola de Amazon EKS que solicitan los clientes. Amazon EKS Connector solo recopila o envía estos datos si usted inicia una acción desde la consola o la API de Amazon EKS que requiere que los datos se envíen a AWS. Aparte del número de versión de Kubernetes, AWS no almacena ningún dato de forma predeterminada. Los almacena solo si usted lo autoriza.

P: ¿Puedo conectar un clúster fuera de una Región de AWS?

R: Sí, puede conectar un clúster desde cualquier ubicación con Amazon EKS. Además, su servicio Amazon EKS se puede ubicar en cualquier Región de AWS comercial pública de AWS. Esto funciona con una conexión de red válida entre el clúster y la Región de AWS de destino. Le recomendamos elegir la Región de AWS más cercana a la ubicación del clúster para optimizar el rendimiento de la interfaz del usuario. Por ejemplo, si tiene un clúster en ejecución en Tokio, conecte el clúster a la Región de AWS ubicada en Tokio (es decir, la Región de AWS `ap-northeast-1`) para obtener una latencia baja. Puede conectar un clúster desde cualquier ubicación con Amazon EKS en cualquier Regiones de AWS comercial pública, excepto una Regiones de AWS de China o GovCloud.

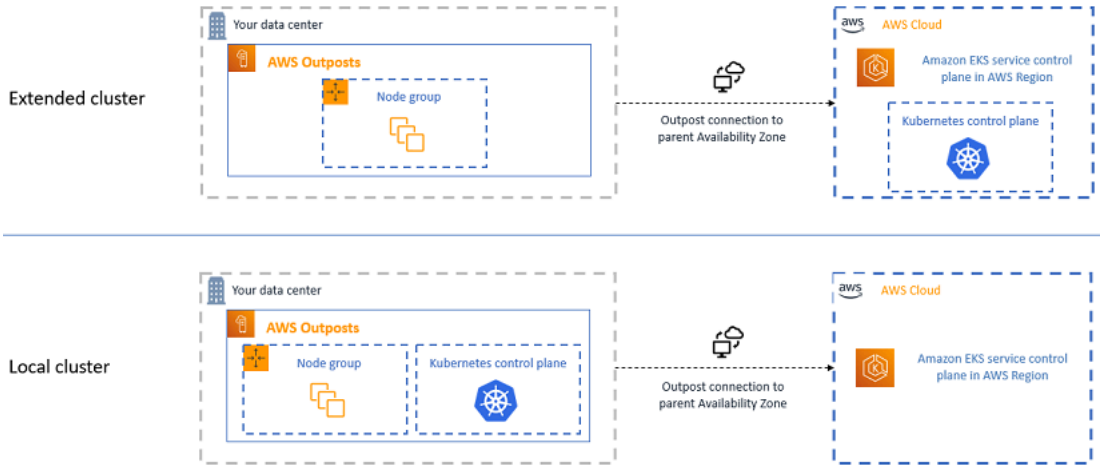
Amazon EKS en AWS Outposts

Puede usar Amazon EKS para ejecutar aplicaciones de Kubernetes en las instalaciones con AWS Outposts. Puede implementar Amazon EKS en Outposts de las siguientes formas:

- Clústeres extendidos: ejecute el plano de control de Kubernetes en una Región de AWS y nodos en el Outpost.
- Clústeres locales: ejecute el plano de control de Kubernetes y nodos en el Outpost.

Para ambas opciones de implementación, el plano de control de Kubernetes está completamente administrado por AWS. Puede usar las mismas API, herramientas y consola de Amazon EKS que usa en la nube para crear y ejecutar Amazon EKS en Outposts.

En el siguiente diagrama, se muestran estas opciones de implementación.



Cuándo usar cada opción de implementación

Tanto los clústeres locales como los extendidos son opciones de implementación de uso general y se pueden usar para una variedad de aplicaciones.

Con los clústeres locales, puede ejecutar todo el clúster de Amazon EKS de forma local en Outposts. Esta opción puede mitigar el riesgo de tiempo de inactividad de las aplicaciones que puede resultar de la desconexión temporal de la red a la nube. Estas desconexiones de red pueden deberse a cortes de fibra o eventos climáticos. Dado que todo el clúster de Amazon EKS se ejecuta de forma local en Outposts, las aplicaciones permanecen disponibles. De esta manera, puede realizar operaciones de clúster durante las desconexiones de red a la nube. Para obtener más información,

consulte Preparación para las desconexiones de red. Si le preocupa la calidad de la conexión de red de sus Outposts a la Región de AWS principal y requiere una alta disponibilidad durante desconexiones de red, use la opción de implementación de clústeres locales.

Los clústeres extendidos le permiten conservar la capacidad de su Outpost porque el plano de control de Kubernetes se ejecuta en la Región de AWS principal. Esta puede ser la opción más adecuada si puede invertir en una conectividad de red confiable y redundante desde su Outpost hacia la Región de AWS. La calidad de la conexión de red es fundamental para esta opción. La forma en que Kubernetes gestiona las desconexiones de red entre el plano de control de Kubernetes y los nodos puede provocar un tiempo de inactividad de la aplicación. Para obtener más información sobre el comportamiento de Kubernetes, consulte Programación, prevención y expulsión en la documentación de Kubernetes.

Comparación de las opciones de implementación

En la siguiente tabla, se comparan las diferencias entre las dos opciones.

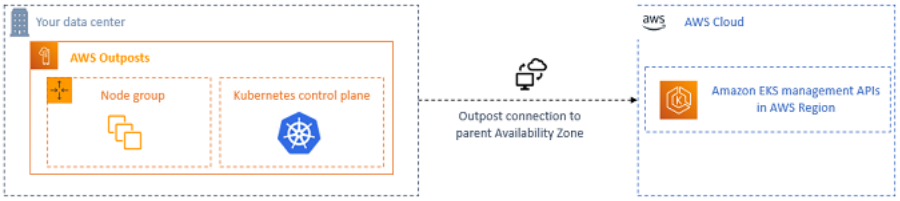
Funcionalidad	Clúster extendido	Clúster local
Ubicación del plano de control de Kubernetes	Región de AWS	Outpost
Cuenta del plano de control de Kubernetes	Cuenta de AWS	Su cuenta
Disponibilidad regional	Consulte Puntos de conexión de servicio	Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Norte de California), Oeste de EE. UU. (Oregón), Asia-Pacífico (Seúl), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney), Asia-Pacífico (Tokio), Canadá (centro), Europa (Fráncfort), Europa (Irlanda), Europa (Londres), Medio Oriente (Baréin) y América del Sur (São Paulo).

Funcionalidad	Clúster extendido	Clúster local
Versiones secundarias de Kubernetes	Versiones compatibles de Amazon EKS.	Versiones compatibles de Amazon EKS.
Versiones de la plataforma	Consulte Versiones de la plataforma de Amazon EKS	Consulte Versiones de la plataforma de clústeres locales de Amazon EKS
Factores de forma de Outpost	Bastidores de Outpost	Bastidores de Outpost
Interfaces de usuario	AWS Management Console, AWS CLI, API de Amazon EKS, `eksctl`, AWS CloudFormation y Terraform	AWS Management Console, AWS CLI, API de Amazon EKS, `eksctl`, AWS CloudFormation y Terraform
Políticas administradas	AmazonEKSClusterPolicy y AmazonEKSServiceRolePolicy	AmazonEKSLocalOutpostClusterPolicy y AmazonEKSLocalOutpostServiceRolePolicy
VPC y subredes del clúster	Consulte Requisitos y consideraciones de Amazon EKS VPC y subred	Consulte Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS
Acceso al punto de conexión del clúster	Público o privado o ambos	Solo privada
Autenticación del servidor de API de Kubernetes	AWS Identity and Access Management (IAM) y OIDC	IAM y certificados `x.509`
Tipos de nodos	Solo autoadministrado	Solo autoadministrado
Tipos de computación de nodos	Amazon EC2 bajo demanda	Amazon EC2 bajo demanda
Tipos de almacenamiento de nodos	gp2 de Amazon EBS y SSD de NVMe local	gp2 de Amazon EBS y SSD de NVMe local

Funcionalidad	Clúster extendido	Clúster local
AMI optimizadas para Amazon EKS	Amazon Linux, Windows y Bottlerocket	Solo Amazon Linux
Versiones de IP	Sólo `IPv4`	Sólo `IPv4`
Complementos	Complementos de Amazon EKS o complementos autoadministrados	Solo complementos autoadministrados
Interfaz de red de contenedores predeterminada	Amazon VPC CNI plugin for Kubernetes	Amazon VPC CNI plugin for Kubernetes
Registros del plano de control de Kubernetes	Registros de Amazon CloudWatch	Registros de Amazon CloudWatch
Equilibrio de carga	Usar el AWS Load Balancer Controller para aprovisionar únicamente equilibradores de carga de aplicación (no equilibradores de carga de red)	Usar el AWS Load Balancer Controller para aprovisionar únicamente equilibradores de carga de aplicación (no equilibradores de carga de red)
Cifrado de sobre para secretos	Consulte Habilitación del cifrado de secretos en un clúster existente	No compatible
Roles de IAM para cuentas de servicio	Consulte Roles de IAM para cuentas de servicio	No compatible
Resolución de problemas	Consulte Solución de problemas de Amazon EKS	Consulte Solución de problemas de clústeres locales para Amazon EKS en AWS Outposts

Clústeres locales para Amazon EKS en AWS Outposts

Puede usar clústeres locales para ejecutar todo su clúster de Amazon EKS de forma local en AWS Outposts. Esto ayuda a mitigar el riesgo de tiempo de inactividad de las aplicaciones que puede resultar de la desconexión temporal de la red a la nube. Estas desconexiones pueden deberse a cortes de fibra o eventos climáticos. Dado que todo el clúster de Kubernetes se ejecuta de forma local en Outposts, las aplicaciones permanecen disponibles. De esta manera, puede realizar operaciones de clúster durante las desconexiones de red a la nube. Para obtener más información, consulte Preparación para las desconexiones de red. En el siguiente diagrama, se muestra una implementación de clúster local.



Los clústeres locales suelen estar disponibles para su uso con los bastidores de Outposts.

Regiones de AWS admitidas

Puede crear clústeres locales en las siguientes Regiones de AWS: Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Norte de California), Oeste de EE. UU. (Oregón), Asia-Pacífico (Seúl), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney), Asia-Pacífico (Tokio), Canadá (centro), Europa (Fráncfort), Europa (Irlanda), Europa (Londres), Medio Oriente (Baréin) y América del Sur (São Paulo). Para obtener información detallada sobre las características admitidas, consulte Comparación de las opciones de implementación.

Temas

- Creación de un clúster local en un Outpost
- Versiones de la plataforma de clústeres locales de Amazon EKS
- Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS
- Preparación para las desconexiones de red
- Consideraciones de capacidad
- Solución de problemas de clústeres locales para Amazon EKS en AWS Outposts

Creación de un clúster local en un Outpost

En este tema, se proporciona información general sobre lo que debe tener en cuenta al ejecutar un clúster local en un Outpost. El tema también contiene instrucciones sobre cómo implementar un clúster local en un Outpost.

Consideraciones

> **Important**
>
> - Estas consideraciones no se replican en la documentación relacionada de Amazon EKS. Si otros temas de la documentación de Amazon EKS entran en conflicto con las consideraciones que se presentan aquí, siga las consideraciones que se indican a continuación.
> - Estas consideraciones están sujetas a modificaciones y pueden cambiar con frecuencia. Por lo tanto, le recomendamos que revise este tema con regularidad.
> - Muchas de las consideraciones son diferentes a las consideraciones para crear un clúster en la Nube de AWS.

- Los clústeres locales solo admiten bastidores de Outpost. Un único clúster local puede ejecutarse en varios bastidores de Outpost físicos que comprenden un único Outpost lógico. Un único clúster local no puede ejecutarse en varios Outposts lógicos. Cada Outpost lógico tiene un único ARN de Outpost.
- Los clústeres locales ejecutan y administran el plano de control de Kubernetes en su cuenta en Outpost. No se pueden ejecutar cargas de trabajo en instancias del plano de control de Kubernetes ni modificar los componentes del plano de control de Kubernetes. Estos nodos están administrados por el servicio de Amazon EKS. Los cambios en el plano de control de Kubernetes no persisten a través de las acciones de administración automáticas de Amazon EKS, como la aplicación de parches.
- Los clústeres locales admiten complementos autoadministrados y grupos de nodos autoadministrados de Amazon Linux. Los complementos Amazon VPC CNI plugin for Kubernetes, `kube-proxy` y CoreDNS se instalan automáticamente en los clústeres locales.
- Los clústeres locales requieren el uso de Amazon EBS en Outposts. Su Outpost debe tener Amazon EBS disponible para el almacenamiento del plano de control de Kubernetes.

- Los clústeres locales usan Amazon EBS en Outposts. Su Outpost debe tener Amazon EBS disponible para el almacenamiento del plano de control de Kubernetes. Los Outposts admiten únicamente volúmenes `gp2` de Amazon EBS.
- Los `PersistentVolumes` de Kubernetes respaldados por Amazon EBS se admiten mediante el controlador de CSI de Amazon EBS.

Requisitos previos

- Familiaridad con las opciones de implementación de Outposts, las Consideraciones de capacidad y los Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS.
- Un Outpost existente. Para obtener más información, consulte ¿Qué es AWS Outposts?
- La herramienta de la línea de comandos de `kubectl` está instalada en la computadora o AWS CloudShell. La versión puede ser la misma o hasta una versión secundaria anterior o posterior a la versión de Kubernetes de su clúster. Por ejemplo, si la versión del clúster es `1.29`, puede usar la versión `1.28`, `1.29` o `1.30` de `kubectl` con él. Para instalar o actualizar `kubectl`, consulte Instalación o actualización del `kubectl`.
- La versión `2.12.3` o posterior, o bien, la versión `1.27.160` o posterior de la AWS Command Line Interface (AWS CLI) instalada y configurada en su dispositivo o AWS CloudShell. Para comprobar su versión actual, utilice **`aws --version | cut -d / -f2 | cut -d ' ' -f1`**. Los administradores de paquetes tales como `yum`, `apt-get` o Homebrew para macOS suelen estar atrasados varias versiones respecto de la versión de la AWS CLI más reciente. Para instalar la versión más reciente, consulte Instalar, actualizar y desinstalar la AWS CLI y Configuración rápida con aws configure en la Guía del usuario de AWS Command Line Interface. La versión de AWS CLI instalada en AWS CloudShell también puede estar atrasada varias versiones respecto de la versión más reciente. Para actualizarla, consulte Instalar la AWS CLI en su directorio de inicio en la Guía del usuario de AWS CloudShell.
- Una entidad principal de IAM (usuario o rol) con permisos para `create` y `describe` un clúster de Amazon EKS. Para obtener más información, consulte Crea un clúster local de Kubernetes en un Outpost y Enumeración o descripción de todos los clústeres.

Cuando se crea un clúster local de Amazon EKS, la entidad principal de IAM que crea el clúster se agrega de manera permanente. La entidad principal de IAM se agrega específicamente a la tabla de autorizaciones RBAC de Kubernetes como administrador. Esta entidad tiene permisos `system:masters`. La identidad de esta entidad no está visible en la configuración del clúster. Por lo tanto, es importante anotar la entidad que creó el clúster y asegurarse de no eliminarlo nunca.

Al principio, solo la entidad principal que creó el servidor puede realizar llamadas al servidor de la API de Kubernetes con `kubectl`. Si usa la consola para crear el clúster, debe asegurarse de que las mismas credenciales de IAM se encuentren en la cadena de credenciales del SDK de AWS al ejecutar comandos de `kubectl` en el clúster. Una vez que se crea el clúster, puede conceder acceso a su clúster a otras entidades principales de IAM.

Para crear un clúster local de Amazon EKS

Puede crear un clúster local con `eksctl`, la AWS Management Console, la AWS CLI, la API de Amazon EKS, los SDK de AWS, AWS CloudFormation o Terraform.

1. Cree un clúster local.

   eksctl

   Requisito previo

   La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para instalar o actualizar `eksctl`, consulte Instalación en la documentación de `eksctl`.

   Para crear el clúster con **`eksctl`**

   1. Copie los siguientes contenidos en su dispositivo. Reemplace los siguientes valores y, a continuación, ejecute el comando modificado para crear el archivo `outpost-control-plane.yaml`:
      - Reemplace *`region-code`* por la Región de AWS admitida en la que desea crear su clúster.
      - Reemplace *`my-cluster`* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
      - Reemplace *`vpc-ExampleID1`* y *`subnet-ExampleID1`* con los ID de su VPC y subred existentes. La subred y la VPC deben cumplir los requisitos de Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS.
      - Reemplace *`uniqueid`* con el ID de su Outpost.

- Reemplace *m5.large* con un tipo de instancia disponible en su Outpost. Antes de elegir un tipo de instancia, consulte Consideraciones de capacidad. Se implementan tres instancias del plano de control. No puede cambiar este número.

```
cat >outpost-control-plane.yaml <<EOF
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: my-cluster
  region: region-code
  version: "1.24"

vpc:
  clusterEndpoints:
    privateAccess: true
  id: "vpc-vpc-ExampleID1"
  subnets:
    private:
      outpost-subnet-1:
        id: "subnet-subnet-ExampleID1"

outpost:
  controlPlaneOutpostARN: arn:aws:outposts:region-code:111122223333:outpost/
op-uniqueid
  controlPlaneInstanceType: m5.large
EOF
```

Para obtener una lista completa de todas las opciones y valores predeterminados disponibles, consulte Soporte de AWS Outposts y Esquema del archivo de configuración en la documentación de `eksctl`.

2. Para crear el clúster, use el archivo de configuración que creó en el paso anterior. `eksctl` crea una VPC y una subred en su Outpost para implementar el clúster.

```
eksctl create cluster -f outpost-control-plane.yaml
```

El aprovisionamiento de clústeres tarda varios minutos. Mientras se crea el clúster, aparecen varias líneas de salida. La última línea de salida es similar a la siguiente línea de ejemplo.

```
[#]  EKS cluster "my-cluster" in "region-code" region is ready
```

> **Tip**
>
> Para ver la mayoría de las opciones que se pueden especificar al crear un clúster con `eksctl`, utilice el comando **`eksctl create cluster --help`**. Para ver todas las opciones disponibles, puede utilizar un archivo `config`. Para obtener más información, consulte Uso de archivos de configuración y el esquema de archivos de configuración en la documentación de `eksctl`. Puede encontrar ejemplos de archivos de configuración en GitHub.

`Eksctl` creó automáticamente una entrada de acceso para la entidad principal de IAM (usuario o rol) que creó el clúster y concedió al administrador de la entidad principal de IAM permisos para los objetos de Kubernetes en el clúster. Si no desea que el creador del clúster tenga acceso de administrador a los objetos de Kubernetes en el clúster, agregue el siguiente texto al archivo de configuración anterior: **`bootstrapClusterCreatorAdminPermissions: false`** (al mismo nivel que `metadata`, `vpc` y `outpost`). Si agregó la opción, después de crear el clúster, tendrá que crear una entrada de acceso para al menos una entidad principal de IAM; de lo contrario, ninguna entidad principal de IAM tendrá acceso a los objetos de Kubernetes en el clúster.

AWS Management Console

Requisito previo

Una VPC y subred existentes que cumplen con los requisitos de Amazon EKS. Para obtener más información, consulte Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS.

Para crear el clúster con la AWS Management Console

1. Si ya tiene un rol de IAM de clúster local o va a crear su clúster con `eksctl`, puede omitir este paso. Por defecto, `eksctl` crea un rol para usted.
   a. Ejecute el siguiente comando para crear un archivo de política de confianza JSON de IAM.

```
cat >eks-local-cluster-role-trust-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

b. Cree el rol de IAM del clúster de Amazon EKS. Para crear un rol de IAM, a la entidad principal de IAM que está creando el rol se le debe asignar la acción `iam:CreateRole` (permiso).

```
aws iam create-role --role-name myAmazonEKSLocalClusterRole --assume-role-
policy-document file://"eks-local-cluster-role-trust-policy.json"
```

c. Adjunte la política administrada de IAM por Amazon EKS denominada `AmazonEKSLocalOutpostClusterPolicy` al rol. Para adjuntar una política de IAM a una entidad principal de IAM, se debe asignar una de las siguientes acciones de IAM (permisos) a la entidad principal que adjunta la política: `iam:AttachUserPolicy` o `iam:AttachRolePolicy`.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSLocalOutpostClusterPolicy --role-name myAmazonEKSLocalClusterRole
```

2. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
3. En la parte superior de la pantalla de la consola, asegúrese de haber seleccionado una Región de AWS admitida.
4. Elija Agregar clúster y, a continuación, elija Crear.
5. En la página Configurar clúster, rellene o seleccione los valores para los siguientes campos:
   - Ubicación del plano de control deKubernetes: elija AWS Outposts.

- ID de Outpost: elija el ID del Outpost en el que desea crear su plano de control.
- Tipo de instancia: seleccione un tipo de instancia. Solo se muestran los tipos de instancias disponibles en su Outpost. En la lista desplegable, cada tipo de instancia describe para cuántos nodos se recomienda el tipo de instancia. Antes de elegir un tipo de instancia, consulte Consideraciones de capacidad. Todas las réplicas se implementan con el mismo tipo de instancia. Después de crear el clúster, no se puede cambiar el tipo de instancia. Se implementan tres instancias del plano de control. No puede cambiar este número.
- Nombre: un nombre único para el clúster. Debe ser único en su Cuenta de AWS. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
- Versión de Kubernetes: elija la versión de Kubernetes que desea utilizar para el clúster. Le recomendamos seleccionar la versión más reciente, a menos que necesite usar una versión anterior.
- Rol de servicio del clúster: elija el rol de IAM del clúster de Amazon EKS que creó en un paso anterior para permitir que el plano de control de Kubernetes administre los recursos de AWS.
- Acceso de administrador del clúster de Kubernetes: si desea que la entidad principal de IAM (rol o usuario) que está creando el clúster tenga acceso de administrador a los objetos de Kubernetes en el clúster, acepte la opción predeterminada (permitir). Amazon EKS crea una entrada de acceso para la entidad principal de IAM y concede permisos de administrador de clúster a la entrada de acceso. Para obtener más información acerca de las entradas de acceso, consulte Administración de entradas de acceso.

  Si desea que una entidad principal de IAM diferente a la entidad principal que creó el clúster tenga acceso de administrador a los objetos del clúster de Kubernetes, seleccione la opción no permitir. Después de crear el clúster, cualquier entidad principal de IAM que tenga permisos de IAM para crear entradas de acceso puede añadir una entrada de acceso para cualquier entidad principal de IAM que necesite acceder a los objetos del clúster de Kubernetes. Para obtener más información sobre los permisos necesarios de IAM, consulte Acciones definidas por Amazon Elastic Kubernetes Service en la Referencia de autorización de servicios. Si elige la opción de no permitir y no

crea ninguna entrada de acceso, ninguna entidad principal de IAM tendrá acceso a los objetos del clúster de Kubernetes.

- Etiquetas: (opcional) agregue las etiquetas a su clúster. Para obtener más información, consulte Etiquetado de los recursos de Amazon EKS.

Cuando haya terminado con esta página, seleccione Siguiente.

6. En la página Especificar red, seleccione valores para los siguientes campos:
   - VPC: elija una VPC existente. La VPC debe tener un número suficiente de direcciones IP disponibles para el clúster, los nodos y otros recursos de Kubernetes que desee crear. La VPC debe cumplir los requisitos de Requisitos y consideraciones de la VPC.
   - Subredes: de forma predeterminada, se preseleccionan todas las subredes disponibles de la VPC especificada en el campo anterior. Las subredes que elija deben cumplir los requisitos de Requisitos y consideraciones de la subred.

     Grupos de seguridad: (opcional) especifique uno o varios grupos de seguridad que desea que Amazon EKS asocie a las interfaces de red que crea. Amazon EKS crea automáticamente un grupo de seguridad que habilita la comunicación entre el clúster y la VPC. Amazon EKS asocia este grupo de seguridad, y el que elija, a las interfaces de red que crea. Para obtener más información acerca del grupo de seguridad de clúster que crea Amazon EKS, consulte Requisitos y consideraciones sobre grupos de seguridad de Amazon EKS. Puede modificar las reglas del grupo de seguridad del clúster que crea Amazon EKS. Si elige agregar sus propios grupos de seguridad, no puede cambiar los que elija tras la creación del clúster. Para que los hosts en las instalaciones se comuniquen con el punto de conexión del clúster, debe permitir el tráfico saliente desde el grupo de seguridad del clúster. Para los clústeres que no tienen una conexión a Internet de entrada y salida (también conocidos como clústeres privados), debe realizar una de las siguientes acciones:
     - Agregue el grupo de seguridad asociado a los puntos de conexión de VPC requeridos. Para obtener más información acerca de los puntos de conexión requeridos, consulte Puntos de conexión de VPC de interfaz en Acceso de la subred a Servicios de AWS.
     - Modifique el grupo de seguridad que creó Amazon EKS para permitir el tráfico del grupo de seguridad asociado a los puntos de conexión de VPC.

   Cuando haya terminado con esta página, seleccione Siguiente.

7. En la página Configurar observabilidad, si lo desea, puede elegir qué opciones de métricas y registro de plano de control quiere activar. De forma predeterminada, cada tipo de registro está desactivado.
   - Para obtener más información sobre la opción de métricas de Prometheus, consulte Active las métricas de Prometheus al crear un clúster.
   - Para obtener más información sobre las opciones de Registro de plano de control, consulte Registro de plano de control de Amazon EKS.

   Cuando haya terminado con esta página, seleccione Siguiente.
8. En la página Revisar y crear, revise la información que introdujo o seleccionó en las páginas anteriores. Si necesita realizar cambios, seleccione Edit (Editar). Cuando esté satisfecho, seleccione Create (Crear). El campo Estado muestra CREANDO mientras se aprovisiona el clúster.

   El aprovisionamiento de clústeres tarda varios minutos.

2. Una vez creado el clúster, puede ver las instancias del plano de control de Amazon EC2 que se crearon.

```
aws ec2 describe-instances --query 'Reservations[*].Instances[*].{Name:Tags[?
Key==`Name`]|[0].Value}' | grep my-cluster-control-plane
```

Un ejemplo de salida sería el siguiente.

```
"Name": "my-cluster-control-plane-id1"
"Name": "my-cluster-control-plane-id2"
"Name": "my-cluster-control-plane-id3"
```

Cada instancia tiene un taint `node-role.eks-local.amazonaws.com/control-plane` aplicado, de modo que nunca se programen cargas de trabajo en las instancias del plano de control. Para obtener más información sobre las taints, consulte Taints y toleraciones en la documentación de Kubernetes. Amazon EKS supervisa continuamente el estado de los clústeres locales. Realizamos acciones de administración automáticas, como la aplicación de parches de seguridad y la reparación de instancias en mal estado. Cuando los clústeres locales se desconectan de la nube, realizamos acciones para garantizar que el clúster vuelva a un buen estado al volver a conectarse.

3. Si ha creado el clúster mediante `eksctl`, puede omitir este paso. `eksctl` completa este paso por usted. Habilite `kubectl` para comunicarse con el clúster agregando un nuevo contexto al archivo `kubectl config`. Para obtener instrucciones acerca de cómo crear y actualizar el archivo, consulte Creación o actualización de un archivo `kubeconfig` para un clúster de Amazon EKS.

   ```
   aws eks update-kubeconfig --region region-code --name my-cluster
   ```

   Un ejemplo de salida sería el siguiente.

   ```
   Added new context arn:aws:eks:region-code:111122223333:cluster/my-cluster to /home/username/.kube/config
   ```

4. Para conectarse al servidor de la API de Kubernetes de su clúster local, debe tener acceso a la puerta de enlace local de la subred o conectarse desde la VPC. Para obtener más información acerca de la conexión de un bastidor de Outpost a su red en las instalaciones, consulte Cómo funcionan las puertas de enlace locales para bastidores en la Guía del usuario de AWS Outposts. Si usa el enrutamiento directo de VPC y la subred de Outpost tiene una ruta a la puerta de enlace local, las direcciones IP privadas de las instancias del plano de control de Kubernetes se transmiten automáticamente a través de la red local. El punto de conexión del servidor de la API de Kubernetes del clúster local está alojado en Amazon Route 53 (Route 53). Los servidores de DNS públicos pueden resolver el punto de conexión del servicio de API en las direcciones IP privadas de los servidores de la API de Kubernetes.

   Las instancias del plano de control de Kubernetes de los clústeres locales se configuran con interfaces de red elásticas estáticas con direcciones IP privadas fijas que no cambian a lo largo del ciclo de vida del clúster. Es posible que las máquinas que interactúan con el servidor de API de Kubernetes no tengan conectividad con Route 53 durante desconexiones de red. Si este es el caso, recomendamos configurar `/etc/hosts` con las direcciones IP privadas estáticas para continuar con las operaciones. También recomendamos configurar servidores de DNS locales y conectarlos a su Outpost. Para obtener más información, consulte la Documentación de AWS Outposts. Ejecute el siguiente comando para confirmar que se ha establecido la comunicación con el clúster.

   ```
   kubectl get svc
   ```

   Un ejemplo de salida sería el siguiente.

```
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   28h
```

5. (Opcional) Pruebe la autenticación en su clúster local cuando se encuentra desconectado de la Nube de AWS. Para obtener instrucciones, consulte Preparación para las desconexiones de red.

Recursos internos

Amazon EKS crea los siguientes recursos en su clúster. Los recursos son para uso interno de Amazon EKS. Para que el clúster funcione correctamente, no edite ni modifique estos recursos.

- Los siguientes Pods de reflejo:
  - `aws-iam-authenticator-`*`node-hostname`*
  - `eks-certificates-controller-`*`node-hostname`*
  - `etcd-`*`node-hostname`*
  - `kube-apiserver-`*`node-hostname`*
  - `kube-controller-manager-`*`node-hostname`*
  - `kube-scheduler-`*`node-hostname`*
- Los siguientes complementos autoadministrados:
  - `kube-system/coredns`
  - `kube-system/kube-proxy` (no se crean hasta que agregue su primer nodo)
  - `kube-system/aws-node` (no se crea hasta que agregue su primer nodo). Los clústeres locales usan el complemento Amazon VPC CNI plugin for Kubernetes para redes de clústeres. No cambie la configuración de las instancias del plano de control (pods denominados `aws-node-controlplane-*`). Hay variables de configuración que puede usar para cambiar el valor predeterminado para cuando el complemento crea interfaces de red nuevas. Para obtener más información, consulte la documentación en GitHub.
- Los siguientes servicios:
  - `default/kubernetes`
  - `kube-system/kube-dns`
- Una `PodSecurityPolicy` denominada `eks.system`
- Una `ClusterRole` denominada `eks:system:podsecuritypolicy`
- Una `ClusterRoleBinding` denominada `eks:system`

- Una PodSecurityPolicy predeterminada
- Además del grupo de seguridad del clúster, Amazon EKS crea un grupo de seguridad en su Cuenta de AWS llamado `eks-local-internal-do-not-use-or-edit-`*`cluster-name`*`-`*`uniqueid`*. Este grupo de seguridad permite que el tráfico fluya libremente entre los componentes de Kubernetes que se ejecutan en las instancias del plano de control.

Siguientes pasos recomendados:

- Conceda a la entidad principal de IAM que creó el clúster los permisos necesarios para ver los recursos de Kubernetes en la AWS Management Console
- Conceda acceso a las entidades de IAM al clúster. Si desea que las entidades vean los recursos de Kubernetes en la consola de Amazon EKS, conceda los Permisos necesarios a las entidades.
- Configure el registro para el clúster
- Familiarícese con lo que sucede durante las desconexiones de red.
- Agregue nodos al clúster
- Considere la posibilidad de configurar un plan de copia de seguridad para su `etcd`. Amazon EKS no admite la copia de seguridad ni la restauración automatizadas de `etcd` para clústeres locales. Para obtener más información, consulte Copia de seguridad de un clúster `etcd` en la documentación de Kubernetes. Las dos opciones principales son usar `etcdctl` para automatizar la toma de instantáneas o usar la copia de seguridad del volumen de almacenamiento de Amazon EBS.

Versiones de la plataforma de clústeres locales de Amazon EKS

Las versiones de la plataforma del clúster local representan las capacidades del clúster de Amazon EKS en AWS Outposts. Las versiones incluyen los componentes que se ejecutan en el plano de control de Kubernetes, cuyos indicadores del servidor de API de Kubernetes están habilitados. También incluyen la versión del parche de Kubernetes actual. Cada versión secundaria de Kubernetes tiene una o varias versiones de la plataforma asociadas. Las versiones de la plataforma para las diferentes versiones secundarias de Kubernetes son independientes. Las versiones de plataforma para los clústeres locales y los clústeres de Amazon EKS en la nube son independientes.

Cuando está disponible una nueva versión secundaria de Kubernetes para clústeres locales, como `1.28`, la versión inicial de la plataforma para esa versión secundaria de Kubernetes comienza con `eks-local-outposts.1`. Sin embargo, Amazon EKS lanza nuevas versiones de la plataforma

de forma periódica para habilitar nuevos ajustes de plano de control de Kubernetes y proporcionar revisiones de seguridad.

Cuando están disponibles nuevas versiones de la plataforma de clústeres locales para una versión secundaria:

- El número de versión de la plataforma se incrementa (`eks-local-outposts.n+1`).
- Amazon EKS actualiza automáticamente todos los clústeres locales existentes a la última versión de la plataforma para su versión secundaria de Kubernetes correspondiente. Las actualizaciones automáticas de las versiones de la plataforma existentes se implementan de forma incremental. El proceso de implementación puede tardar algún tiempo. Si necesita las características de la versión más reciente de la plataforma de forma inmediata, le recomendamos que cree un nuevo clúster local.
- Amazon EKS puede publicar una nueva AMI de nodo con la versión de parche correspondiente. Todas las versiones de parche son compatibles entre el plano de control de Kubernetes y las AMI de nodo para una única versión secundaria de Kubernetes.

Las nuevas versiones de la plataforma de no introducen cambios bruscos ni provocan interrupciones de servicio.

Los clústeres locales siempre se crean con la última versión de la plataforma disponible (`eks-local-outposts.n`) para la versión de Kubernetes especificada.

Las versiones de la plataforma actuales y recientes se describen en las siguientes tablas.

Versión **`1.28`** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma `1.28`: `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `DefaultStorageClass`, `DefaultTolerationSeconds`, `ExtendedResourceToleration`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `PersistentVolumeClaimResize`, `Priority`, `PodSecurity`, `ResourceQuota`, `RuntimeClass`, `ServiceAccount`, `StorageObjectInUseProtection`, `TaintNodesByCondition`, `ValidatingAdmissionPolicy` y `ValidatingAdmissionWebhook`.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.28.6	eks-local-outposts.5	Se actualizó la versión de Bottlerocket a la v1.19.3, que incluye las correcciones de errores más recientes para admitir el arranque local en Outposts.	18 de abril de 2024
1.28.6	eks-local-outposts.4	Nueva versión de la plataform a con mejoras y correccio nes de seguridad. Soporte restaurado o arranque local en Outposts. Versión de Bottleroc ket degradada a v1.15.1 por motivos de compatibilidad.	2 de abril de 2024
1.28.6	eks-local-outposts.3	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	22 de marzo de 2024
1.28.6	eks-local-outposts.2	Nueva versión de la plataform a con mejoras y correccio nes de seguridad kube-proxy actualizada av1.28.6. AWS Autenticador de IAM actualiza do a v0.6.17. Complemen to CNI de Amazon VPC para Kubernetes adaptado a la versión v1.13.2 por motivos de compatibilidad. Versión de Bottlerocket actualizada a v1.19.2.	8 de marzo de 2024

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.28.1	eks-local-outposts.1	Lanzamiento inicial de la versión v1.28 de Kubernetes para los clústeres locales de Amazon EKS en Outpost.	4 de octubre de 2023

Versión **1.27** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma 1.27: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurity, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition, ValidatingAdmissionPolicy y ValidatingAdmissionWebhook.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.27.10	eks-local-outposts.5	Nueva plataforma con mejoras y correcciones de seguridad.	2 de abril de 2024
1.27.10	eks-local-outposts.4	Nueva plataforma con mejoras y correcciones de seguridad. Kube-proxy actualizado a v1.27.10. AWS Authenticador de IAM actualizado a v0.6.17. Versión de Bottlerocket actualizada a v1.19.2.	22 de marzo de 2024
1.27.3	eks-local-outposts.3	Nueva versión de la plataforma con mejoras y correcciones	14 de julio de 2023

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
		de seguridad. kube-prox y actualizado a v1.27.3. Complemento CNI de Amazon VPC para Kubernetes actualiza do a v1.13.2.	
1.27.1	eks-local-outposts.2	Se actualizó la imagen de CoreDNS a v1.10.1	22 de junio de 2023
1.27.1	eks-local-outposts.1	Lanzamiento inicial de la versión 1.27 de Kubernetes para los clústeres locales de Amazon EKS en Outposts.	30 de mayo de 2023

Versión **1.26** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma 1.26: CertificateApproval, CertificateSigning, CertificateSubjectRestriction, DefaultIngressClass, DefaultStorageClass, DefaultTolerationSeconds, ExtendedResourceToleration, LimitRanger, MutatingAdmissionWebhook, NamespaceLifecycle, NodeRestriction, PersistentVolumeClaimResize, Priority, PodSecurity, ResourceQuota, RuntimeClass, ServiceAccount, StorageObjectInUseProtection, TaintNodesByCondition, ValidatingAdmissionPolicy y ValidatingAdmissionWebhook.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.26.13	eks-local-outposts.5	Nueva versión de la plataform a con mejoras y correccio nes de seguridad kube-proxy actualizada a v1.26.13. AWS	22 de marzo de 2024

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
		Autenticador de IAM actualizado a `v0.6.17`. Versión de Bottlerocket actualizada a `v1.19.2`.	

Versión **1.25** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de `1.25`: `CertificateApproval`, `CertificateSigning`, `CertificateSubjectRestriction`, `DefaultIngressClass`, `DefaultStorageClass`, `DefaultTolerationSeconds`, `ExtendedResourceToleration`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `PersistentVolumeClaimResize`, `Priority`, `PodSecurity`, `ResourceQuota`, `RuntimeClass`, `ServiceAccount`, `StorageObjectInUseProtection`, `TaintNodesByCondition` y `ValidatingAdmissionWebhook`.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
`1.25.16`	`eks-local-outposts.7`	Nueva versión de la plataforma con mejoras y correcciones de seguridad kube-proxy actualizada a `v1.25.16`. AWS Autenticador de IAM actualizado a `v0.6.17`. Versión de Bottlerocket actualizada a `v1.19.2`.	22 de marzo de 2024
`1.25.11`	`eks-local-outposts.6`	Nueva versión de la plataforma con mejoras y correcciones de seguridad. `kube-prox` y actualizado a `v1.25.11`.	14 de julio de 2023

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
		Complemento CNI de Amazon VPC para Kubernetes actualizado a `v1.13.2`.	
`1.25.9`	`eks-local-outposts.5`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	13 de julio de 2023
`1.25.6`	`eks-local-outposts.4`	Se actualizó la versión de Bottlerocket a la `1.13.2`	2 de mayo de 2023
`1.25.6`	`eks-local-outposts.3`	Se actualizó el sistema operativo de instancias del plano de control de Amazon EKS a la versión `v1.13.1` de Bottlerocket y el complemento CNI de Amazon VPC de Kubernetes se actualizó a la versión `v1.12.6`.	14 de abril de 2023
`1.25.6`	`eks-local-outposts.2`	Recopilación de diagnósticos mejorada para instancias del plano de control de Kubernetes.	8 de marzo de 2023
`1.25.6`	`eks-local-outposts.1`	Lanzamiento inicial de la versión `1.25` de Kubernetes para los clústeres locales de Amazon EKS en Outposts.	1 de marzo de 2023

Versión **`1.24`** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de `1.24`: `DefaultStorageClass`, `DefaultTolerationSeconds`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`,

NodeRestriction, ResourceQuota, ServiceAccount, ValidatingAdmissionWebhook, PodSecurityPolicy, TaintNodesByCondition, StorageObjectInUseProtection, PersistentVolumeClaimResize, ExtendedResourceToleration, CertificateApproval, PodPriority, CertificateSigning, CertificateSubjectRestriction, RuntimeClass y DefaultIngressClass.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.24.17	eks-local-outposts.7	Nueva versión de la plataforma con mejoras y correcciones de seguridad kube-proxy actualizada a v1.25.16. AWS Autenticador de IAM actualizado a v0.6.17. Versión de Bottlerocket actualizada a v1.19.2.	22 de marzo de 2024
1.24.15	eks-local-outposts.6	Nueva versión de la plataforma con mejoras y correcciones de seguridad. kube-prox y actualizado a v1.24.15. Complemento CNI de Amazon VPC para Kubernetes actualizado a v1.13.2.	14 de julio de 2023
1.24.13	eks-local-outposts.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	13 de julio de 2023
1.24.9	eks-local-outposts.4	Se actualizó la versión de Bottlerocket a la 1.13.2	2 de mayo de 2023
1.24.9	eks-local-outposts.3	Se actualizó el sistema operativo de instancias del plano de control de Amazon EKS a la versión v1.13.1 de	14 de abril de 2023

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
		Bottlerocket y el complemento CNI de Amazon VPC de Kubernetes se actualizó a la versión `v1.12.6`.	
`1.24.9`	`eks-local-outposts.2`	Recopilación de diagnósticos mejorada para instancias del plano de control de Kubernetes.	8 de marzo de 2023
`1.24.9`	`eks-local-outposts.1`	Lanzamiento inicial de la versión `1.24` de Kubernetes para los clústeres locales de Amazon EKS en Outposts.	17 de enero de 2023

Versión **`1.23`** de Kubernetes

Los siguientes controladores de admisión están habilitados para todas las versiones de plataforma de `1.23`: `DefaultStorageClass`, `DefaultTolerationSeconds`, `LimitRanger`, `MutatingAdmissionWebhook`, `NamespaceLifecycle`, `NodeRestriction`, `ResourceQuota`, `ServiceAccount`, `ValidatingAdmissionWebhook`, `PodSecurityPolicy`, `TaintNodesByCondition`, `StorageObjectInUseProtection`, `PersistentVolumeClaimResize`, `ExtendedResourceToleration`, `CertificateApproval`, `PodPriority`, `CertificateSigning`, `CertificateSubjectRestriction`, `RuntimeClass` y `DefaultIngressClass`.

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
`1.23.17`	`eks-local-outposts.6`	Nueva versión de la plataforma con mejoras y correcciones de seguridad.	13 de julio de 2023

Versión de Kubernetes	Versión de la plataforma de Amazon EKS	Notas de la versión	Fecha de publicación
1.23.17	eks-local-outposts.5	Nueva versión de la plataforma con mejoras y correcciones de seguridad kube-proxy actualizada a v1.23.17. Versión de Bottlerocket actualizada a v1.14.1.	6 de julio de 2023
1.23.15	eks-local-outposts.4	Se actualizó el sistema operativo de instancias del plano de control de Amazon EKS a la versión v1.13.1 de Bottlerocket y el complemento CNI de Amazon VPC de Kubernetes se actualizó a la versión v1.12.6.	14 de abril de 2023
1.23.15	eks-local-outposts.3	Recopilación de diagnósticos mejorada para instancias del plano de control de Kubernetes.	8 de marzo de 2023
1.23.15	eks-local-outposts.2	Lanzamiento inicial de la versión 1.23 de Kubernetes para los clústeres locales de Amazon EKS en Outposts.	17 de enero de 2023

Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS

Al crear un clúster local, especifica una VPC y al menos una subred privada que se ejecuta en Outposts. En este tema, se proporciona una descripción general de los requisitos y las consideraciones de la VPC y las subredes para el clúster local.

Requisitos y consideraciones de la VPC

Al crear un clúster local, la VPC que especifique debe cumplir los siguientes requisitos y consideraciones:

- Asegúrese de que la VPC tenga suficientes direcciones IP para el clúster local, los nodos y otros recursos de Kubernetes que desee crear. Si la VPC que desea utilizar no tiene suficientes direcciones IP, aumente el número de direcciones IP disponibles. Puede hacerlo asociando bloques adicionales de enrutamiento entre dominios sin clase (CIDR) con su VPC. Puede asociar bloques de CIDR privados (RFC 1918) y públicos (no RFC 1918) a su VPC antes o después de crear el clúster. Un clúster puede tardar hasta cinco horas en reconocer un bloque de CIDR asociado a una VPC.
- La VPC no puede tener prefijos IP asignados ni bloques de CIDR IPv6. Debido a estas restricciones, la información incluida en Aumentar la cantidad de direcciones IP disponibles para sus nodos de Amazon EC2 y Direcciones `IPv6` de clústeres, Pods y services no aplica a su VPC.
- La VPC tiene un nombre de host de DNS y la resolución de DNS habilitados. Sin estas características, se produce un error en la creación del clúster local y tendrá que habilitarlas y volver a crear el clúster. A fin de obtener más información, consulte Atributos de DNS para su VPC en la Guía del usuario de Amazon VPC.
- Para acceder al clúster local a través de la red local, la VPC debe estar asociada a la tabla de enrutamiento de la puerta de enlace local del Outpost. Para obtener más información, consulte Asociaciones de la VPC en la Guía del usuario de AWS Outposts.

Requisitos y consideraciones de la subred

Debe especificar al menos una subred privada al crear el clúster. Si especifica más de una subred, las instancias del plano de control de Kubernetes se distribuyen de manera uniforme en las subredes. Si se especifica más de una subred, las subredes deben existir en el mismo Outpost. Además, las subredes también deben tener las rutas y los permisos de grupos de seguridad adecuados para comunicarse entre sí. Las subredes que especifique al crear un clúster local deben cumplir los siguientes requisitos:

- Todas las subredes están en el mismo Outpost lógico.
- En conjunto, las subredes tienen al menos tres direcciones IP disponibles para las instancias del plano de control de Kubernetes. Si se especifican tres subredes, cada subred debe tener al menos una dirección IP disponible. Si se especifican dos subredes, cada subred debe tener al menos

dos direcciones IP disponibles. Si se especifica una subred, la subred debe tener al menos tres direcciones IP disponibles.

- Las subredes tienen una ruta a la puerta de enlace local del bastidor del Outpost para acceder al servidor de la API de Kubernetes a través de su red local. Si las subredes no tienen una ruta a la puerta de enlace local del bastidor del Outpost, debe comunicarse con el servidor de la API de Kubernetes desde dentro de la VPC.
- Las subredes deben utilizar nombres basados en direcciones IP. La nomenclatura basada en recursos de Amazon EC2 no es compatible con Amazon EKS.

Acceso de la subred a Servicios de AWS

Las subredes privadas del clúster local en Outposts deben poder comunicarse con Servicios de AWS regionales. Para ello, puede utilizar una puerta de enlace NAT para el acceso saliente a Internet o, si desea mantener la privacidad de todo el tráfico dentro de su VPC, mediante los puntos de conexión de la interfaz de la VPC.

Uso de una puerta de enlace NAT

Las subredes privadas del clúster local en Outposts deben tener una tabla de enrutamiento asociada con una ruta a una puerta de enlace NAT en una subred pública en la zona de disponibilidad principal del Outpost. La subred pública debe tener una ruta hacia una puerta de enlace de Internet. La puerta de enlace NAT permite el acceso a Internet saliente y evita las conexiones entrantes no solicitadas desde Internet hacia las instancias del Outpost.

Uso de los puntos de conexión de VPC de interfaz

Si las subredes privadas del clúster local en Outposts no tienen una conexión a Internet de salida o si desea mantener todo el tráfico privado dentro de su VPC, debe crear los siguientes puntos de conexión de VPC de interfaz y punto de conexión de puerta de enlace en una subred regional antes de crear el clúster.

Punto de conexión	Tipo de punto de conexión
`com.amazonaws.` *`region-code`* `.ssm`	Interfaz
`com.amazonaws.` *`region-code`* `.ssmmessages`	Interfaz

Punto de conexión	Tipo de punto de conexión
com.amazonaws. *region-code* .ec2messages	Interfaz
com.amazonaws. *region-code* .ec2	Interfaz
com.amazonaws. *region-code* .secretsmanager	Interfaz
com.amazonaws. *region-code* .logs	Interfaz
com.amazonaws. *region-code* .sts	Interfaz
com.amazonaws. *region-code* .ecr.api	Interfaz
com.amazonaws. *region-code* .ecr.dkr	Interfaz
com.amazonaws. *region-code* .s3	Puerta de enlace

Los puntos de conexión debe cumplir los siguientes requisitos:

- Deben crearse en una subred privada ubicada en la zona de disponibilidad principal de su Outpost
- Deben tener habilitados los nombres de DNS privados
- Deben tener un grupo de seguridad adjunto que permita el tráfico HTTPS entrante desde el rango CIDR de la subred de Outpost privada.

La creación de puntos de conexión incurre en costos. Para más información, consulte Precios de AWS PrivateLink. Si sus Pods necesitan acceso a otros Servicios de AWS, debe crear puntos de conexión adicionales. Para obtener una lista completa de puntos de conexión, consulte Servicios de AWS que se integren con AWS PrivateLink.

Creación de una VPC

Puede crear una VPC que cumpla los requisitos anteriores mediante una de las siguientes plantillas de AWS CloudFormation:

- Plantilla 1: esta plantilla crea una VPC con una subred privada en el Outpost y una subred pública en la Región de AWS. La subred privada tiene una ruta a Internet a través de una puerta de enlace

NAT que reside en la subred pública de la Región de AWS. Esta plantilla se puede usar para crear un clúster local en una subred con acceso a Internet de salida.

- Plantilla 2: esta plantilla crea una VPC con una subred privada en el Outpost y el conjunto mínimo de puntos de conexión de VPC necesarios para crear un clúster local en una subred que no tenga acceso a Internet de entrada o salida (también denominada subred privada).

Preparación para las desconexiones de red

Puede seguir usando su clúster local de Amazon EKS en un Outpost si su red local ha perdido la conectividad con la Nube de AWS. En este tema, se explica cómo preparar el clúster local para las desconexiones de red y las consideraciones relacionadas.

Consideraciones a la hora de preparar el clúster local para una desconexión de red:

- Los clústeres locales permiten la estabilidad y las operaciones continuas durante las desconexiones de red temporales e imprevistas. AWS Outposts sigue siendo una oferta completamente conectada que funciona como una extensión de la Nube de AWS en su centro de datos. En caso de que la red se desconecte entre su Outpost y la Nube de AWS, le recomendamos que intente restablecer la conexión. Para obtener instrucciones, consulte la lista de verificación para la solución de problemas de red de los bastidores de AWS Outposts en la Guía del usuario de AWS Outposts. Para obtener más información acerca de cómo solucionar problemas con clústeres locales, consulte Solución de problemas de clústeres locales para Amazon EKS en AWS Outposts.
- Los Outposts emiten una métrica `ConnectedStatus` que puede usar para supervisar el estado de conectividad de su Outpost. Para obtener más información, consulte Métricas de Outposts en la Guía del usuario de AWS Outposts.
- Los clústeres locales usan IAM como mecanismo de autenticación predeterminado mediante el Autenticador de AWS Identity and Access Management para Kubernetes. IAM no está disponible durante las desconexiones de red. Entonces, los clústeres locales admiten un mecanismo de autenticación alternativo mediante certificados `x.509` que puede usar para conectarse a su clúster durante las desconexiones de red. Para obtener información acerca de cómo obtener y usar un certificado `x.509` para su clúster, consulte Autenticación en el clúster local durante una desconexión de red.
- Si no puede acceder a Route 53 durante las desconexiones de red, considere la posibilidad de usar servidores de DNS locales en su entorno en las instalaciones. Las instancias del plano de control de Kubernetes usan direcciones IP estáticas. Puede configurar los hosts que usa para

conectarse a su clúster con el nombre de host y las direcciones IP del punto de conexión como alternativa al uso de servidores de DNS locales. Para obtener más información, consulte DNS en la Guía del usuario de AWS Outposts.

- Si prevé aumentos en el tráfico de aplicaciones durante las desconexiones de red, puede aprovisionar capacidad de computación sobrante en su clúster cuando se conecte a la nube. Las instancias de Amazon EC2 están incluidas en el precio de AWS Outposts. Por lo tanto, la ejecución de instancias de repuesto no afecta al costo de uso de AWS.
- Durante las desconexiones de red, para habilitar las operaciones de creación, actualización y escalado de las cargas de trabajo, las imágenes del contenedor de la aplicación deben ser accesibles a través de la red local y el clúster debe tener capacidad suficiente. Los clústeres locales no alojan un registro de contenedores en su nombre. Las imágenes del contenedor se almacenan en caché en los nodos si los Pods se ejecutaron anteriormente en esos nodos. Si normalmente extrae las imágenes del contenedor de su aplicación de Amazon ECR en la nube, considere la posibilidad de ejecutar una caché o un registro local. Una caché o un registro local es útil si necesita crear, actualizar y escalar operaciones para los recursos de las cargas de trabajo durante desconexiones de red.
- Los clústeres locales usan Amazon EBS como clase de almacenamiento predeterminada para los volúmenes persistentes y el controlador de CSI de Amazon EBS para administrar el ciclo de vida de los volúmenes persistentes de Amazon EBS. Durante las desconexiones de red, las Pods respaldadas por Amazon EBS no se pueden crear, actualizar ni escalar. Esto se debe a que estas operaciones requieren llamadas a la API de Amazon EBS en la nube. Si está implementando cargas de trabajo con estado en clústeres locales y necesita crear, actualizar o escalar operaciones durante las desconexiones de red, considere la posibilidad de usar un mecanismo de almacenamiento alternativo.
- Las instantáneas de Amazon EBS no se pueden crear ni eliminar si AWS Outposts no puede acceder a las API pertinentes de AWS en la región (como las API de Amazon EBS o Amazon S3).
- Al integrar ALB (Ingress) con AWS Certificate Manager (ACM), los certificados se cargan y almacenan en la memoria de la instancia ALB Compute de AWS Outposts. La terminación actual de TLS seguirá funcionando en caso de que se desconecte de la Región de AWS. Las operaciones de mutación en este contexto fallarán (como las nuevas definiciones de entrada, las nuevas operaciones de API de certificados basadas en ACM, la escala de computación de ALB o la rotación de certificados). Para obtener más información, consulte Solución de problemas de renovación administrada de certificados en la Guía del usuario de AWS Certificate Manager.
- Los registros del plano de control de Amazon EKS se almacenan en caché de forma local en instancias del plano de control de Kubernetes durante las desconexiones de red. Al volver a

conectarse, los registros se envían a Registros de CloudWatch en la Región de AWS principal. Puede utilizar Prometheus, Grafana o las soluciones de socios de Amazon EKS para supervisar el clúster a nivel local mediante el punto de conexión de métricas del servidor de la API de Kubernetes o usar Fluent Bit para registros.

- Si está usando el AWS Load Balancer Controller en Outposts para el tráfico de aplicaciones, los Pods existentes encabezados por el AWS Load Balancer Controller seguirán recibiendo tráfico durante las desconexiones de red. Los nuevos Pods creados durante las desconexiones de red no reciben tráfico hasta que el Outpost se vuelve a conectar a la Nube de AWS. Considere la posibilidad de configurar el recuento de réplicas de sus aplicaciones mientras esté conectado a la Nube de AWS para satisfacer sus necesidades de escalado durante las desconexiones de red.
- El Amazon VPC CNI plugin for Kubernetes predeterminado es un modo IP secundario. Está configurado con `WARM_ENI_TARGET=1`, que permite al complemento mantener "una interfaz de red elástica completa" de las direcciones IP disponibles. Considere cambiar los valores `WARM_ENI_TARGET`, `WARM_IP_TARGET` y `MINIMUM_IP_TARGET` de acuerdo con sus necesidades de escalado durante un estado desconectado. Para obtener más información, consulte el archivo `readme` para el complemento en GitHub. Para obtener una lista del número máximo de Pods admitidos por cada tipo de instancia, consulte el archivo `eni-max-pods.txt` en GitHub.

Autenticación en el clúster local durante una desconexión de red

AWS Identity and Access Management (IAM) no está disponible durante las desconexiones de red. No puede autenticarse en su clúster local con las credenciales de IAM mientras esté desconectado. Sin embargo, puede conectarse a su clúster a través de su red local mediante certificados `x509` cuando esté desconectado. Debe descargar y almacenar un certificado cliente `X509` para usar durante las desconexiones. En este tema, aprenderá a crear y usar el certificado para autenticarse en su clúster cuando está en estado desconectado.

1. Crear una solicitud de firma de certificado.

   a. Genere una solicitud de firma de certificado.

   ```
   openssl req -new -newkey rsa:4096 -nodes -days 365 \
       -keyout admin.key -out admin.csr -subj "/CN=admin"
   ```

   b. Cree una solicitud de firma de certificado en Kubernetes.

```
BASE64_CSR=$(cat admin.csr | base64 -w 0)
cat << EOF > admin-csr.yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: admin-csr
spec:
  signerName: kubernetes.io/kube-apiserver-client
  request: ${BASE64_CSR}
  usages:
  - client auth
EOF
```

2. Cree una solicitud de firma de certificado con `kubectl`.

```
kubectl create -f admin-csr.yaml
```

3. Compruebe el estado de la solicitud de firma de certificado.

```
kubectl get csr admin-csr
```

Un ejemplo de salida sería el siguiente.

```
NAME       AGE   REQUESTOR                      CONDITION
admin-csr  11m   kubernetes-admin               Pending
```

Kubernetes ha creado la solicitud de firma de certificado.

4. Apruebe la solicitud de firma de certificado.

```
kubectl certificate approve admin-csr
```

5. Vuelva a comprobar el estado de la solicitud de firma de certificado para aprobarla.

```
kubectl get csr admin-csr
```

Un ejemplo de salida sería el siguiente.

```
NAME       AGE   REQUESTOR                      CONDITION
admin-csr  11m   kubernetes-admin               Approved
```

6. Recupere y verifique el certificado.

   a. Recupere el certificado.

   ```
   kubectl get csr admin-csr -o jsonpath='{.status.certificate}' | base64 --decode
    > admin.crt
   ```

   b. Verifique el certificado.

   ```
   cat admin.crt
   ```

7. Cree un enlace de roles de clúster para un usuario `admin`.

   ```
   kubectl create clusterrolebinding admin --clusterrole=cluster-admin \
       --user=admin --group=system:masters
   ```

8. Genere un kubeconfig con ámbito de usuario para un estado desconectado.

   Puede generar un archivo `kubeconfig` con los certificados de `admin` descargados. Reemplace *`my-cluster`* y *`apiserver-endpoint`* en los siguientes comandos.

   ```
   aws eks describe-cluster --name my-cluster \
       --query "cluster.certificateAuthority" \
       --output text | base64 --decode > ca.crt
   ```

   ```
   kubectl config --kubeconfig admin.kubeconfig set-cluster my-cluster \
       --certificate-authority=ca.crt --server apiserver-endpoint --embed-certs
   ```

   ```
   kubectl config --kubeconfig admin.kubeconfig set-credentials admin \
       --client-certificate=admin.crt --client-key=admin.key --embed-certs
   ```

   ```
   kubectl config --kubeconfig admin.kubeconfig set-context admin@my-cluster \
       --cluster my-cluster --user admin
   ```

   ```
   kubectl config --kubeconfig admin.kubeconfig use-context admin@my-cluster
   ```

9. Vea el archivo `kubeconfig`.

   ```
   kubectl get nodes --kubeconfig admin.kubeconfig
   ```

10. Si ya tiene servicios en producción en su Outpost, omita este paso. Si Amazon EKS es el único servicio que se ejecuta en su Outpost y el Outpost no está en producción actualmente, puede simular una desconexión de red. Antes de iniciar la producción con el clúster local, simule una desconexión para asegurarse de que puede acceder al clúster cuando esté en estado desconectado.

    a. Aplique las reglas de firewall en los dispositivos de red que conectan su Outpost a la Región de AWS. Esto desconecta el enlace de servicio del Outpost. No puede crear ninguna instancia nueva. Las instancias que se están ejecutando actualmente pierden la conectividad con la Región de AWS e Internet.

    b. Puede probar la conexión a su clúster local mientras está desconectado mediante el certificado `x509`. Asegúrese de cambiar su `kubeconfig` a la `admin.kubeconfig` que creó en un paso anterior. Reemplace *`my-cluster`* por el nombre de su clúster local.

    ```
    kubectl config use-context admin@my-cluster --kubeconfig admin.kubeconfig
    ```

    Si observa algún problema con sus clústeres locales mientras están desconectados, le recomendamos que abra un tique de soporte.

Consideraciones de capacidad

En este tema, se proporciona orientación para seleccionar el tipo de instancia del plano de control de Kubernetes y (opcionalmente) para usar grupos de ubicación para cumplir con los requisitos de alta disponibilidad del clúster local de Amazon EKS en un Outpost.

Antes de seleccionar un tipo de instancia (como `m5`, `c5` o `r5`) para usarlo en el plano de control de Kubernetes de su clúster local en Outposts, confirme los tipos de instancias disponibles en su configuración de Outpost. Una vez que haya identificado los tipos de instancias disponibles, seleccione el tamaño de la instancia (como `large`, `xlarge` o `2xlarge`) en función de la cantidad de nodos que requieren sus cargas de trabajo. La siguiente tabla proporciona recomendaciones para elegir el tamaño de una instancia.

> **Note**
>
> Los tamaños de las instancias deben estar establecidos en sus Outposts. Asegúrese de tener capacidad suficiente para tres instancias del tamaño disponible en sus Outposts durante la vida útil de su clúster local. Para ver la lista de los tipos de instancias Amazon EC2

disponibles, consulte las secciones Computación y almacenamiento en las características de los bastidores de AWS Outposts.

Número de nodos	Tamaño de instancia del plano de control de Kubernetes
1–20	`large`
21–100	`xlarge`
101–250	`2xlarge`
251–500	`4xlarge`

El almacenamiento para el plano de control de Kubernetes requiere 246 GB de almacenamiento de Amazon EBS por cada clúster local para cumplir con la IOPS requerida de `etcd`. Los volúmenes de Amazon EBS se aprovisionan automáticamente cuando se crea el clúster local.

Ubicación del plano de control

Si no especifica un grupo de ubicación en la propiedad `OutpostConfig.ControlPlanePlacement.GroupName`, las instancias de Amazon EC2 aprovisionadas para su plano de control de Kubernetes no reciben ninguna obligación específica de ubicación de hardware en toda la capacidad subyacente disponible en su Outpost.

Puede usar grupos de ubicación para cumplir con los requisitos de alta disponibilidad de su clúster local de Amazon EKS en un Outpost. Al especificar un grupo de ubicaciones durante la creación del clúster, se influye en la ubicación de las instancias del plano de control de Kubernetes. Las instancias se distribuyen en un hardware subyacente independiente (bastidores o hosts), lo que minimiza el impacto correlacionado de las instancias en caso de errores de hardware.

Requisitos

El tipo de distribución que puede configurar depende de la cantidad de bastidores de Outpost que tenga en su implementación.

- Implementaciones con uno o dos bastidores físicos en un único puesto avanzado lógico: debe tener al menos tres hosts configurados con el tipo de instancia que elija para las instancias del

plano de control de Kubernetes. Un grupo de ubicación de dispersión que utilice la dispersión a nivel de host garantiza que todas las instancias del plano de control de Kubernetes se ejecuten en distintos hosts dentro de los bastidores subyacentes disponibles en su implementación de Outpost.

- Implementaciones con tres o más bastidores físicos en un único puesto avanzado lógico: debe tener al menos tres hosts configurados con el tipo de instancia que elija para las instancias del plano de control de Kubernetes. Un grupo de ubicación de dispersión que utilice la dispersión a nivel de host garantiza que todas las instancias del plano de control de Kubernetes se ejecutan en distintos bastidores en su implementación de Outpost. También puede utilizar el grupo de ubicación de dispersión a nivel de host de servidor tal y como se describe en la opción anterior.

Usted es responsable de crear el grupo de ubicación deseado. Especifique el grupo de ubicación al llamar a la API de `CreateCluster`. Para más información sobre los grupos de colocación y cómo crearlos, consulte Grupos de ubicación en la Guía del usuario de Amazon EC2.

Consideraciones

- Cuando se especifica un grupo de ubicación, debe haber capacidad distribuida disponible en su Outpost para crear correctamente un clúster local de Amazon EKS. La capacidad varía en función de si se utiliza el tipo de distribución de host o de bastidores. Puede que no haya suficiente capacidad, el clúster permanece en el estado de `Creating`. Puede comprobar el `Insufficient Capacity Error` en el campo de estado de la respuesta de la API de `DescribeCluster`. Debe liberar capacidad para que el proceso de creación progrese.
- Durante las actualizaciones de la plataforma y la versión del clúster local de Amazon EKS, las instancias del plano de control de Kubernetes del clúster se sustituyen por nuevas instancias mediante una estrategia de actualización continua. Durante este proceso de reemplazo, se termina cada instancia del plano de control, lo que libera su ranura respectiva. Se aprovisiona una nueva instancia actualizada en su lugar. Es posible que la instancia actualizada se coloque en la ranura que se publicó. Si la ranura la consume otra instancia no relacionada y no queda más capacidad que respete el requisito de topología de dispersión requerido, el clúster permanece en el estado de `Updating`. Puede comprobar el `Insufficient Capacity Error` correspondiente en el campo de estado de la respuesta de la API de `DescribeCluster`. Debe liberar capacidad para que el proceso de actualización pueda avanzar y restablecer los niveles de alta disponibilidad anteriores.
- Puede crear un máximo de 500 grupos de ubicación por cuenta en cada Región de AWS. Para obtener más información, consulte Normas generales y limitaciones en la Guía del usuario de Amazon EC2.

Solución de problemas de clústeres locales para Amazon EKS en AWS Outposts

En este tema, se tratan algunos errores habituales que pueden aparecer al usar clústeres locales y cómo solucionarlos. Si bien los clústeres locales son similares a los clústeres de Amazon EKS en la nube, existen algunas diferencias en la forma en que Amazon EKS los administra.

Comportamiento de la API

Los clústeres locales se crean mediante la API de Amazon EKS, pero se ejecutan de forma asincrónica. Esto significa que las solicitudes a la API de Amazon EKS se devuelven inmediatamente para los clústeres locales. Sin embargo, estas solicitudes pueden tener éxito, responder rápido a los errores debido a errores de validación de entradas o fallar y tener errores de validación descriptivos. Este comportamiento es similar a la API de Kubernetes.

Los clústeres locales no pasan a un estado `FAILED`. Amazon EKS intenta conciliar el estado del clúster con el estado deseado solicitado por el usuario de forma continua. Como resultado, un clúster local puede permanecer en el estado `CREATING` durante un periodo prolongado hasta que se resuelva el problema subyacente.

Describir el campo de estado del clúster

Los problemas de los clústeres locales se pueden detectar con el comando `describe-cluster` de la AWS CLI de Amazon EKS. Los problemas de los clústeres locales aparecen en el campo `cluster.health` de la respuesta del comando `describe-cluster`. El mensaje contenido en este campo incluye un código de error, un mensaje descriptivo y los ID de los recursos relacionados. Esta información se encuentra disponible solo a través de la API y la AWS CLI de Amazon EKS. En el siguiente ejemplo, reemplace *`my-cluster`* por el nombre de su clúster local.

```
aws eks describe-cluster --name my-cluster --query 'cluster.health'
```

Un ejemplo de salida sería el siguiente.

```
{
    "issues": [
        {
            "code": "ConfigurationConflict",
            "message": "The instance type 'm5.large' is not supported in Outpost 'my-
outpost-arn'.",
            "resourceIds": [
```

```
                "my-cluster-arn"
            ]
        }
    ]
}
```

Si el problema no se puede solucionar, es posible que tenga que eliminar el clúster local y crear uno nuevo. Por ejemplo, intentar aprovisionar un clúster con un tipo de instancia que no está disponible en su Outpost. La siguiente tabla incluye errores comunes relacionados con el estado.

Situación de error	Código	Mensaje	ResourceIds
No se encontraron las subredes proporcionadas.	`ResourceNotFound`	`The subnet ID subnet-id does not exist`	Todos los ID de subred proporcionados
Las subredes proporcionadas no pertenecen a la misma VPC.	`ConfigurationConflict`	`Subnets specified must belong to the same VPC`	Todos los ID de subred proporcionados
Algunas subredes proporcionadas no pertenecen al Outpost especificado.	`ConfigurationConflict`	`Subnet subnet-id expected to be in outpost-arn , but is in other-outpost-arn`	ID de subred problemático
Algunas subredes proporcionadas no pertenecen a ningún Outpost.	`ConfigurationConflict`	`Subnet subnet-id is not part of any Outpost`	ID de subred problemático
Algunas subredes proporcionadas no tienen suficientes direcciones libres para crear interfaces de red elásticas para	`ResourceLimitExceeded`	`The specified subnet does not have enough free addresses to satisfy the request.`	ID de subred problemático

Situación de error	Código	Mensaje	ResourceIds
instancias del plano de control			
El tipo de instancia del plano de control especificado no es compatible con su Outpost.	`ConfigurationConflict`	`The instance type type is not supported in Outpost outpost-arn`	ARN del clúster
Terminó una instancia de Amazon EC2 del plano de control o `run-instance` se ejecutó correctamente, pero el estado observó cambios en `Terminated` . Esto puede suceder durante un periodo después de que el Outpost se vuelva a conectar y los errores internos de Amazon EBS provoquen una falla en el flujo de trabajo interno de Amazon EC2.	`InternalFailure`	`EC2 instance state "Terminated" is unexpected`	ARN del clúster

Situación de error	Código	Mensaje	ResourceIds
La capacidad de su Outpost es insuficiente. Esto también puede ocurrir durante la creación del clúster si un Outpost está desconectado de la Región de AWS.	`ResourceLimitExceeded`	`There is not enough capacity on the Outpost to launch or start the instance.`	ARN del clúster
Su cuenta superó la cuota del grupo de seguridad.	`ResourceLimitExceeded`	Mensaje de error devuelto por la API de Amazon EC2	ID de la VPC de destino
Su cuenta superó la cuota de la interfaz de red elástica.	`ResourceLimitExceeded`	Mensaje de error devuelto por la API de Amazon EC2	ID de subred de destino
No se pudo acceder a las instancias del plano de control a través de AWS Systems Manager. Para obtener una resolución, consulte No se puede acceder a las instancias del plano de control a través de AWS Systems Manager.	`ClusterUnreachable`	No se puede acceder a las instancias del plano de control de Amazon EKS mediante SSM. Verifique su SSM y la configuración de red y consulte la documentación de solución de problemas de EKS en Outposts.	ID de instancia de Amazon EC2

Situación de error	Código	Mensaje	ResourceIds
Se produjo un error al obtener detalles de un grupo de seguridad administrado o una interfaz de red elástica.	Basado en el código de error del cliente Amazon EC2.	Mensaje de error devuelto por la API de Amazon EC2	Todos los ID de grupo de seguridad administrados
Se produjo un error al autorizar o revocar las reglas de entrada de los grupos de seguridad. Esto se aplica a los grupos de seguridad del clúster y del plano de control.	Basado en el código de error del cliente Amazon EC2.	Mensaje de error devuelto por la API de Amazon EC2	ID de grupo de seguridad problemático
Se produjo un error al eliminar una interfaz de red elástica para una instancia del plano de control	Basado en el código de error del cliente Amazon EC2.	Mensaje de error devuelto por la API de Amazon EC2	ID de interfaz de red elástica problemático

En la siguiente tabla se enumeran los errores de otros Servicios de AWS que se presentan en el campo de estado de la respuesta `describe-cluster`.

Código de error de Amazon EC2	Código del problema del estado del clúster	Descripción
`AuthFailure`	`AccessDenied`	Este error se puede producir por diversas razones. La razón más común es cuando una etiqueta que el servicio usa para determinar el alcance de la política de roles vinculados al servicio se

Código de error de Amazon EC2	Código del problema del estado del clúster	Descripción
		elimina accidentalmente de la instancia del plano de control. Si esto ocurre, Amazon EKS ya no podrá administrar ni supervisar estos recursos de AWS.
`UnauthorizedOperation`	`AccessDenied`	Este error se puede producir por diversas razones. La razón más común es cuando una etiqueta que el servicio usa para determinar el alcance de la política de roles vinculados al servicio se elimina accidentalmente de la instancia del plano de control. Si esto ocurre, Amazon EKS ya no podrá administrar ni supervisar estos recursos de AWS.
`InvalidSubnetID.NotFound`	`ResourceNotFound`	Este error se produce cuando no se encuentra el ID de subred de las reglas de entrada de un grupo de seguridad.
`InvalidPermission.NotFound`	`ResourceNotFound`	Este error se produce cuando los permisos de las reglas de entrada de un grupo de seguridad no son correctos.

Código de error de Amazon EC2	Código del problema del estado del clúster	Descripción
`InvalidGroup.NotFound`	`ResourceNotFound`	Este error se produce cuando no se encuentra el grupo de reglas de entrada de un grupo de seguridad.
`InvalidNetworkInterfaceID.NotFound`	`ResourceNotFound`	Este error se produce cuando no se encuentra el ID de la interfaz de red para las reglas de entrada de un grupo de seguridad.
`InsufficientFreeAddressesInSubnet`	`ResourceLimitExceeded`	Este error se produce cuando se supera la cuota de recursos de subred.
`InsufficientCapacityOnOutpost`	`ResourceLimitExceeded`	Este error se produce cuando se supera la cuota de capacidad del outpost.
`NetworkInterfaceLimitExceeded`	`ResourceLimitExceeded`	Este error se produce cuando se supera la cuota de la interfaz de red elástica.
`SecurityGroupLimitExceeded`	`ResourceLimitExceeded`	Este error se produce cuando se supera la cuota del grupo de seguridad.

Código de error de Amazon EC2	Código del problema del estado del clúster	Descripción
`VcpuLimitExceeded`	`ResourceLimitExceeded`	Esto se observa al crear una instancia de Amazon EC2 en una cuenta nueva. El error podría ser similar al siguiente : “`You have requested more vCPU capacity than your current vCPU limit of 32 allows for the instance bucket that the specified instance type belongs to. Please visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to this limit."`
`InvalidParameterValue`	`ConfigurationConflict`	Amazon EC2 devuelve este código de error si el tipo de instancia especificado no es compatible con Outpost.
Todas las demás fallas	`InternalFailure`	Ninguna

Incapaz de crear o modificar clústeres

Los clústeres locales requieren permisos y políticas diferentes a los de los clústeres de Amazon EKS alojados en la nube. Cuando un clúster no se puede crear y produce un error `InvalidPermissions`, compruebe que el rol de clúster que está usando tiene la política administrada AmazonEKSLocalOutpostClusterPolicy asociada. Todas las demás llamadas a la API requieren el mismo conjunto de permisos que los clústeres de Amazon EKS en la nube.

El clúster está atascado en el estado **`CREATING`**

La cantidad de tiempo que tarda en crearse un clúster local varía según varios factores. Estos factores incluyen la configuración de la red, la configuración de Outpost y la configuración del clúster. En general, se crea un clúster local y cambia al estado `ACTIVE` en un plazo de 15 a 20 minutos. Si un clúster local permanece en el estado `CREATING`, puede llamar a `describe-cluster` para solicitar información sobre la causa en el campo de resultado `cluster.health`.

Los problemas más comunes son los siguientes:

AWS Systems Manager (Systems Manager) encuentra los siguientes problemas:

- El clúster no puede conectarse a la instancia del plano de control desde la Región de AWS en que se encuentra Systems Manager. Para verificarlo, llame a `aws ssm start-session --target` *`instance-id`* desde un host bastión dentro de la región. Si ese comando no funciona, compruebe si Systems Manager se está ejecutando en la instancia del plano de control. Otra solución alternativa es eliminar el clúster y, a continuación, volver a crearlo.
- Es posible que las instancias del plano de control de Systems Manager no tengan acceso a Internet. Compruebe si la subred que proporcionó al crear el clúster tiene una puerta de enlace NAT y una VPC con puerta de enlace de Internet. Use el analizador de accesibilidad de la VPC para verificar que la instancia del plano de control puede llegar a la puerta de enlace de Internet. Para obtener más información, consulte Introducción al Analizador de accesibilidad de la VPC.
- Faltan políticas en el ARN de rol que ha proporcionado. Verifique si Política administrada de AWS: AmazonEKSLocalOutpostClusterPolicy se eliminó del rol. Esto también puede ocurrir si una pila AWS CloudFormation está mal configurada.

Varias subredes están mal configuradas y especificadas al crear un clúster:

- Todas las subredes proporcionadas deben estar asociadas al mismo Outpost y poder comunicarse entre sí. Cuando se especifican varias subredes al crear un clúster, Amazon EKS intenta distribuir las instancias del plano de control en varias subredes.
- Los grupos de seguridad administrados de Amazon EKS se aplican en la interfaz de red elástica. Sin embargo, otros elementos de configuración, como las reglas del firewall NACL, pueden entrar en conflicto con las reglas de la interfaz de red elástica.

Falta o está mal configurada la configuración de DNS de la VPC y de la subred

Consulte Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS.

No puede asociar nodos a un clúster

Causas comunes:

- Problemas con las AMI:
  - Está usando una AMI no admitida. Debe usar la versión v20220620 o posterior para una AMI de Amazon Linux optimizada para Amazon EKS de Amazon Linux optimizada para Amazon EKS.
  - Si usó una plantilla de AWS CloudFormation para crear sus nodos, asegúrese de que no estaba usando una AMI no compatible.
- Falta el `ConfigMap` del Autenticador de IAM de AWS: si falta, tiene que crearlo. Para obtener más información, consulte Aplique el `ConfigMap` de `aws-auth` en su clúster.
- Se ha usado un grupo de seguridad incorrecto: asegúrese de usar `eks-cluster-sg-`*`cluster-name`*`-`*`uniqueid`* para el grupo de seguridad de sus nodos de trabajo. AWS CloudFormation cambia el grupo de seguridad seleccionado para permitir un nuevo grupo de seguridad cada vez que se use la pila.
- Siguientes pasos inesperados de VPC de enlace privado: se pasan datos de CA incorrectos (`--b64-cluster-ca`) o del punto de conexión de la API (`--apiserver-endpoint`).
- Política de seguridad del Pod mal configurada:
  - Los Daemonsets de CoreDNS y Amazon VPC CNI plugin for Kubernetes deben ejecutarse en los nodos para que los nodos puedan unirse al clúster y comunicarse correctamente con él.
  - El Amazon VPC CNI plugin for Kubernetes requiere algunas características de red privilegiadas para funcionar correctamente. Puede ver las características de red privilegiadas con el siguiente comando: `kubectl describe psp eks.privileged`.

  No recomendamos modificar la política de seguridad de pod predeterminada. Para obtener más información, consulte Política de seguridad del pod.

Recopilación de registros

Cuando un Outpost se desconecta del servidor al que está asociado la Región de AWS, es probable que el clúster Kubernetes siga funcionando con normalidad. Sin embargo, si el clúster no funciona correctamente, siga los pasos de solución de problemas que se indican en Preparación para las desconexiones de red. Si tiene otros problemas, póngase en contacto con AWS Support. AWS Support puede guiarlo para descargar y ejecutar una herramienta de recopilación de registros. De esta forma, puede recopilar registros de sus instancias del plano de control del clúster de Kubernetes y enviarlas al soporte de AWS Support para una investigación en mayor profundidad.

No se puede acceder a las instancias del plano de control a través de AWS Systems Manager

Cuando no se puede acceder a las instancias del plano de control de Amazon EKS a través de AWS Systems Manager (Systems Manager), Amazon EKS muestra el siguiente error para su clúster.

```
Amazon EKS control plane instances are not reachable through SSM. Please verify your
 SSM and network configuration, and reference the EKS on Outposts troubleshooting
 documentation.
```

Para resolver este problema, asegúrese de que su VPC y las subredes cumplen con los requisitos de Requisitos y consideraciones de VPC y subred del clúster local de Amazon EKS y que ha completado los pasos de la Configuración del administrador de sesiones en la Guía del usuario de AWS Systems Manager.

Lanzamiento de nodos autoadministrados de Amazon Linux en un Outpost

En este tema, se describe cómo puede lanzar grupos de escalado automático de nodos de Amazon Linux en un Outpost que se registrarán con el clúster de Amazon EKS. El clúster puede estar en la Nube de AWS o en un Outpost.

Requisitos previos

- Un Outpost existente. Para obtener más información, consulte ¿Qué es AWS Outposts?
- Un clúster existente de Amazon EKS. Para implementar un clúster en la Nube de AWS, consulte Creación de un clúster de Amazon EKS. Para implementar un clúster en un Outpost, consulte Clústeres locales para Amazon EKS en AWS Outposts.
- Supongamos que está creando sus nodos en un clúster en la Nube de AWS y tiene subredes en la Región de AWS que está habilitado AWS Outposts, AWS Wavelength o AWS Local Zones. Esas subredes no deberían haberse transferido al crear el clúster. Si está creando sus nodos en un clúster de un Outpost, debe haber pasado una subred de Outpost al crear el clúster.
- (Recomendado para clústeres en la Nube de AWS) El complemento Amazon VPC CNI plugin for Kubernetes configurado con su propio rol de IAM que tenga asociada la política de IAM necesaria. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA). Los clústeres locales no admiten roles de IAM para cuentas de servicio.

Puede crear un grupo de nodos autoadministrados de Amazon Linux con `eksctl` o la AWS Management Console (con una plantilla de AWS CloudFormation). También puede usar Terraform.

eksctl

Requisito previo

La versión `0.183.0` o posterior de la herramienta de línea de comandos `eksctl` instalada en su dispositivo o AWS CloudShell. Para Instalar o actualizar `eksctl`, consulte la sección de Instalación en la documentación de `eksctl`.

Cómo lanzar nodos de Linux autoadministrados mediante **`eksctl`**

1. Si su clúster está en la Nube de AWS y la política de IAM administrada AmazonEKS_CNI_Policy se asocia a su Rol de IAM de nodo de Amazon EKS, recomendamos asignarlo a un rol de IAM asociado a la cuenta de servicios `aws-node` de Kubernetes en su lugar. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA). Si el clúster está en su Outpost, la política debe estar asociada a su rol de nodo.
2. El siguiente comando crea un grupo de nodos en un clúster existente. El clúster debe haberse creado con `eksctl`. Reemplace *`al-nodes`* por un nombre para su grupo de nodos. El nombre del grupo de nodos no puede tener más de 63 caracteres. Debe empezar por una letra o un dígito, pero también puede incluir guiones y guiones bajos como caracteres no iniciales. Reemplace *`my-cluster`* por el nombre del clúster. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster. Si el clúster existe en un Outpost, reemplace *`id`* con el ID de una subred de Outpost. Si su clúster existe en la Nube de AWS, reemplace *`id`* con el ID de una subred que no especificó al crear el clúster. Reemplace *`instance-type`* con un tipo de instancia compatible con su Outpost. Reemplace los *`example values`* restantes por sus propios valores. Los nodos se crean de forma predeterminada con la misma versión de Kubernetes que el plano de control.

   Reemplace *`instance-type`* con un tipo de instancia disponible en su Outpost.

   Reemplace *`my-key`* con el nombre de su par de claves de Amazon EC2 o la clave pública. Esta clave se utiliza para SSH en sus nodos después de que se lancen. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para

obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2.

Cree el grupo de nodos con el siguiente comando.

```
eksctl create nodegroup --cluster my-cluster --name al-nodes --node-
type instance-type \
    --nodes 3 --nodes-min 1 --nodes-max 4 --managed=false --node-volume-type gp2
 --subnet-ids subnet-id
```

Si su clúster está implementado en la Nube de AWS:

- El grupo de nodos que implemente puede asignar direcciones `IPv4` a Pods de un bloque de CIDR diferente que el de la instancia. Para obtener más información, consulte Redes personalizadas para los pods.
- El grupo de nodos que implemente no requiere acceso a Internet saliente. Para obtener más información, consulte Requisitos del clúster privado.

Para obtener una lista completa de todas las opciones y valores predeterminados disponibles, consulte Soporte de AWS Outposts en la documentación de `eksctl`.

Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en Solución de problemas de Amazon EKS y No puede asociar nodos a un clúster en Solución de problemas de clústeres locales para Amazon EKS en AWS Outposts.

Un ejemplo de salida sería el siguiente. Se generan varias líneas mientras se crean los nodos. Una de las últimas líneas de salida es la siguiente línea de ejemplo.

```
[#]  created 1 nodegroup(s) in cluster "my-cluster"
```

3. (Opcional) Implemente una aplicación de muestra para probar el clúster y los nodos de Linux.

AWS Management Console

Paso 1: lanzar nodos autoadministrados de Amazon Linux mediante la AWS Management Console

1. Descargue la versión más reciente de la plantilla de AWS CloudFormation.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2022-12-23/
amazon-eks-nodegroup.yaml
```

2. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
3. Seleccione Create stack (Crear pila) y, a continuación, seleccione With new resources (standard) (Con nuevos recursos [estándar]).
4. Para Especificar plantilla, seleccione Cargar un archivo de plantilla y, a continuación, elija Elegir archivo. Seleccione el archivo `amazon-eks-nodegroup.yaml` que ha descargado en el paso anterior y, a continuación, seleccione Siguiente.
5. En la página Especificar detalles de la pila, ingrese los siguientes parámetros según corresponda y luego seleccione Siguiente:

   - Nombre de pila: elija un nombre para la pila de AWS CloudFormation. Por ejemplo, puede llamarla ***`al-nodes`***. El nombre solo puede contener caracteres alfanuméricos (con distinción de mayúsculas y minúsculas) y guiones. Debe comenzar con un carácter alfanumérico y no puede tener más de 100 caracteres. El nombre debe ser único dentro de la Región de AWS y la Cuenta de AWS en las que va a crear el clúster.
   - ClusterName: ingrese el nombre del clúster. Si este nombre no coincide con el nombre del clúster, los nodos no pueden unirse al clúster.
   - ClusterControlPlaneSecurityGroup: elija el valor de SecurityGroups en la salida de AWS CloudFormation que generó al crear la VPC.

     En los siguientes pasos, se muestra una operación para recuperar el grupo aplicable.

     1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters.
     2. Elija el nombre del clúster.
     3. Elija la pestaña Redes.
     4. Use el valor de Grupo de seguridad adicional como referencia al realizar una selección en la lista desplegable ClusterControlPlaneSecurityGroup.
   - NodeGroupName: escriba un nombre para el grupo de nodos. Este nombre se puede utilizar más adelante para identificar el grupo de nodos de Auto Scaling que se crea para los nodos.
   - NodeAutoScalingGroupMinSize: ingrese el número mínimo de nodos al que se pueda reducir horizontalmente el grupo de escalado automático de nodos.

- NodeAutoScalingGroupDesiredCapacity: escriba el número deseado de nodos que desea escalar cuando se crea la pila.
- NodeAutoScalingGroupMaxSize: ingrese el número máximo de nodos que pueda alcanzar el grupo de Auto Scaling de nodos.
- NodeInstanceType: elija un tipo de instancia para los nodos. Si su clúster se ejecuta en la Nube de AWS, para obtener más información, consulte Elección de un tipo de instancia de Amazon EC2. Si su clúster se ejecuta en un Outpost, solo puede seleccionar un tipo de instancia disponible en su Outpost.
- NodeImageIdSSMParam: rellenado previamente con el parámetro de Amazon EC2 Systems Manager de una AMI optimizada recientemente para Amazon EKS para una versión de Kubernetes variable. Para utilizar otra versión secundaria de Kubernetes compatible con Amazon EKS, reemplace *1.XX* por una versión admitida diferente. Recomendamos especificar la misma versión de Kubernetes que el clúster.

  Para utilizar la AMI acelerada optimizada para Amazon EKS, reemplace *amazon-linux-2* por **amazon-linux-2-gpu**. Para utilizar la AMI Arm optimizada para Amazon EKS, reemplace *amazon-linux-2* por **amazon-linux-2-arm64**.

  > **Note**
  >
  > La AMI del nodo de Amazon EKS se basa en Amazon Linux. Puede realizar un seguimiento de los eventos de seguridad o privacidad de Amazon Linux 2 en el Centro de seguridad de Amazon Linux o suscribirse a la fuente RSS asociada. Los eventos de seguridad y privacidad incluyen información general del problema, qué paquetes están afectados y cómo actualizar las instancias para corregir el problema.

- NodeImageId: (opcional) si utiliza su propia AMI personalizada (en lugar de la AMI optimizada para Amazon EKS), ingrese un ID de AMI de nodo para su Región de AWS. Si especifica un valor aquí, anula cualquier valor del campo NodeImageIdSSMParam.
- NodeVolumeSize: especifique un tamaño de volumen raíz para los nodos en GiB.
- NodeVolumeType: especifique un tipo de volumen raíz para sus nodos.
- KeyName: ingrese el nombre de un par de claves SSH de Amazon EC2 que pueda utilizar para conectar mediante SSH con los nodos después de haberlos lanzado. Si aún no tiene un par de claves de Amazon EC2, puede crear uno en la AWS Management Console. Para

obtener más información, consulte Pares de claves de Amazon EC2 en la Guía del usuario de Amazon EC2.

> **Note**
>
> Si no proporciona un par de claves aquí, se produce un error al crear la pila de AWS CloudFormation.

- BootstrapArguments: hay varios argumentos opcionales que puede pasar a sus nodos. Para obtener más información, consulte la información de uso del script de arranque en GitHub. Si agrega nodos a un clúster local de Amazon EKS en AWS Outposts (en el que las instancias del plano de control de Kubernetes se ejecutan en AWS Outposts) y el clúster no tiene una conexión a Internet de entrada y salida (también conocidos como clústeres privados), debe proporcionar los siguientes argumentos de arranque (como una sola línea).

```
--b64-cluster-ca ${CLUSTER_CA} --apiserver-endpoint https://
${APISERVER_ENDPOINT} --enable-local-outpost true --cluster-id ${CLUSTER_ID}
```

- DisableIMDSv1: cada nodo admite de forma predeterminada la versión 1 (IMDSv1) e IMDSv2 del servicio de metadatos de la instancia. Puede desactivar IMDSv1. Para evitar que los nodos y Pods futuros del grupo de nodos usen IMDSv1, establezca DisableIMDSv1 en true (verdadero). Para obtener más información, consulte Configuración del servicio de metadatos de instancia. Para obtener más información sobre cómo restringir el acceso en sus nodos, consulte Restringir el acceso al perfil de instancias asignado al nodo de trabajo.
- VpcId: ingrese el ID de la VPC que ha creado. Antes de elegir una VPC, revise Requisitos y consideraciones de la VPC.
- Subredes: si su clúster está en un Outpost, elija al menos una subred privada en la VPC. Antes de elegir las subredes, revise Requisitos y consideraciones de las subredes. Puede ver qué subredes son privadas abriendo cada enlace de subred desde la pestaña Redes de su clúster.

6. Seleccione las opciones que desee en la página Configurar las opciones de pila y, a continuación, elija Siguiente.
7. Seleccione la casilla de verificación situada a la izquierda de I acknowledge that AWS CloudFormation might create IAM resources. (Reconozco que podría crear recursos de IAM) y luego seleccione Create stack (Crear pila).
8. Una vez completada la creación de la pila, selecciónela en la consola y elija Salidas.

9. Anote el valor de NodeInstanceRoles correspondiente al grupo de nodos creado. Lo necesitará al configurar los nodos de Amazon EKS de .

Paso 2: permitir a los nodos unirse al clúster

1. Verifique si ya tiene el `ConfigMap` de `aws-auth`.

```
kubectl describe configmap -n kube-system aws-auth
```

2. Si se le muestra un `ConfigMap` de `aws-auth`, actualícelo según sea necesario.

   a. Abra el icono `ConfigMap` para editar.

```
kubectl edit -n kube-system configmap/aws-auth
```

   b. Añada una nueva entrada de `mapRoles` según sea necesario. Establezca el valor de `rolearn` en el valor de NodeInstanceRole que registró en el procedimiento anterior.

```
[...]
data:
  mapRoles: |
    - rolearn: <ARN of instance role (not instance profile)>
      username: system:node:{{EC2PrivateDNSName}}
      groups:
        - system:bootstrappers
        - system:nodes
[...]
```

   c. Guarde el archivo y salga del editor de texto.

3. Si recibe un error que indica “`Error from server (NotFound): configmaps "aws-auth" not found`, aplique el `ConfigMap` bursátil.

   a. Descargue el mapa de configuración.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-
eks/cloudformation/2020-10-29/aws-auth-cm.yaml
```

   b. En el archivo `aws-auth-cm.yaml`, establezca el `rolearn` al valor de NodeInstanceRole que ha registrado en el procedimiento anterior. Puede hacerlo con

un editor de texto o reemplazando *my-node-instance-role* y ejecute el siguiente comando:

```
sed -i.bak -e 's|<ARN of instance role (not instance profile)>|my-node-instance-role|' aws-auth-cm.yaml
```

c. Aplique la configuración. Este comando puede tardar varios minutos en finalizar.

```
kubectl apply -f aws-auth-cm.yaml
```

4. Observe el estado de los nodos y espere a que aparezca el estado `Ready`.

```
kubectl get nodes --watch
```

Ingrese `Ctrl+C` para obtener un símbolo del intérprete de comandos.

> **Note**
>
> Si recibe cualquier error de tipo de recurso o autorización, consulte Acceso denegado o no autorizado (`kubectl`) en el tema de solución de problemas.

Si los nodos no se unen al clúster, consulte Los nodos no pueden unirse al clúster en Solución de problemas de Amazon EKS y No puede asociar nodos a un clúster en Solución de problemas de clústeres locales para Amazon EKS en AWS Outposts.

5. Instale el controlador de CSI de Amazon EBS. Para obtener más información, consulte Installation (Instalación) en GitHub. En la sección Set up driver permission (Configurar permiso de controlador), asegúrese de seguir las instrucciones de la opción Using IAM instance profile (Uso del perfil de instancia de IAM). Debe usar la clase de almacenamiento `gp2`. No se admite la clase de almacenamiento `gp3`.

Para crear una clase de almacenamiento `gp2` en el clúster, realice los siguientes pasos.

1. Ejecute el siguiente comando para crear un archivo `gp2-storage-class.yaml`.

```
cat >gp2-storage-class.yaml <<EOF
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
```

```
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
  name: ebs-sc
provisioner: ebs.csi.aws.com
volumeBindingMode: WaitForFirstConsumer
parameters:
  type: gp2
  encrypted: "true"
allowVolumeExpansion: true
EOF
```

2. Aplique el manifiesto al clúster.

```
kubectl apply -f gp2-storage-class.yaml
```

6. (Solo para nodos de GPU) Si ha elegido un tipo de instancia de GPU y la AMI acelerada optimizada para Amazon EKS, debe aplicar el complemento de dispositivo NVIDIA para Kubernetes como un DaemonSet en su clúster. Reemplace *vX.X.X* con la versión Plugin de dispositivo NVidia/K8S deseada antes de ejecutar el siguiente comando.

```
kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-
plugin/vX.X.X/nvidia-device-plugin.yml
```

Paso 3: acciones adicionales

1. (Opcional) Implemente una aplicación de muestra para probar el clúster y los nodos de Linux.
2. Si el clúster se implementa en un Outpost, omita este paso. Si el clúster se implementa en la Nube de AWS, la siguiente información es opcional. Si la política de IAM administrada AmazonEKS_CNI_Policy se asocia a su Rol de IAM de nodo de Amazon EKS, recomendamos asignarla a un rol de IAM asociado a la cuenta de servicios `aws-node` de Kubernetes en su lugar. Para obtener más información, consulte Configuración de Amazon VPC CNI plugin for Kubernetes para utilizar los roles de IAM en las cuentas de servicio (IRSA).

Proyectos relacionados

Estos proyectos de código abierto extienden la funcionalidad de los clústeres de Kubernetes que se ejecutan en AWS o por fuera, incluidos los clústeres administrados por Amazon EKS.

Herramientas de administración

Herramientas de administración relacionadas para clústeres de Kubernetes y Amazon EKS.

eksctl

`eksctl` es una herramienta de CLI sencilla para crear clústeres de Amazon EKS.

- URL del proyecto
- Documentación del proyecto
- Blog de código abierto de AWS: eksctl: Amazon EKS cluster with one command

Controladores de AWS para Kubernetes

Con los controladores de AWS para Kubernetes, puede crear y administrar recursos de AWS directamente desde su clúster de Kubernetes.

- URL del proyecto
- Blog de código abierto de AWS: operador de servicios de AWS para Kubernetes ahora disponible

Flux CD

Flux es una herramienta que puede utilizar para administrar la configuración del clúster con Git. Utiliza un operador en el clúster para desencadenar las implementaciones dentro de Kubernetes. Para obtener más información sobre los operadores, consulte OperatorHub.io en GitHub.

- URL del proyecto
- Documentación del proyecto

CDK para Kubernetes

Con el CDK para Kubernetes (cdk8s), puede definir aplicaciones y componentes de Kubernetes mediante lenguajes de programación conocidos. Las aplicaciones cdk8s se sintetizan en manifiestos estándar de Kubernetes, que se pueden aplicar a cualquier clúster de Kubernetes.

- URL del proyecto
- Documentación del proyecto
- Blog de contenedores de AWS: Introducción a cdk8s+: API impulsadas por intensión para objetos de Kubernetes

Red

Proyectos de redes relacionados para clústeres de Kubernetes y Amazon EKS.

Amazon VPC CNI plugin for Kubernetes

Amazon EKS admite redes de VPC nativas gracias al Amazon VPC CNI plugin for Kubernetes. El complemento asigna una dirección IP de su VPC a cada Pod.

- URL del proyecto
- Documentación del proyecto

AWS Load Balancer Controller para Kubernetes

El AWS Load Balancer Controller ayuda a administrar los equilibradores de carga elástica de AWS para un clúster de Kubernetes. Satisface los recursos de entrada de Kubernetes mediante el aprovisionamiento de los equilibradores de carga de aplicaciones de AWS. Satisface los recursos de servicio Kubernetes mediante el aprovisionamiento de los equilibradores de carga de red de AWS.

- URL del proyecto
- Documentación del proyecto

ExternalDNS

ExternalDNS sincroniza las entradas y los servicios de Kubernetes expuestos con proveedores de DNS como Amazon Route 53 y detección de servicios de AWS.

- URL del proyecto
- Documentación del proyecto

Machine learning

Proyectos de machine learning relacionados para clústeres de Kubernetes y Amazon EKS.

Kubeflow

Un conjunto de herramientas de machine learning para Kubernetes.

- URL del proyecto
- Documentación del proyecto
- Blog de código abierto de AWS: Kubeflow on Amazon EKS

Auto Scaling

Proyectos de escalado automático relacionados para clústeres de Kubernetes y Amazon EKS.

Escalador automático del clúster

Es una herramienta que ajusta automáticamente el tamaño del clúster de Kubernetes en función de la presión de memoria y CPU.

- URL del proyecto
- Documentación del proyecto
- Taller de Amazon EKS: https://www.eksworkshop.com/

Escalador

Es un escalador automático horizontal optimizado de trabajo o lote para Kubernetes.

- URL del proyecto
- Documentación del proyecto

Monitorización

Proyectos de monitoreo relacionados para clústeres de Kubernetes y Amazon EKS.

Prometheus

Prometheus es un conjunto de herramientas de alerta y monitoreo de sistemas de código abierto.

- URL del proyecto
- Documentación del proyecto
- Taller de Amazon EKS: https://eksworkshop.com/intermediate/240_monitoring/

Integración continua/implementación continua

Proyectos de integración continua/implementación continua relacionados para clústeres de Kubernetes y Amazon EKS.

Jenkins X

Solución de integración continua/implementación continua para aplicaciones en la nube modernas en clústeres de Kubernetes y Amazon EKS.

- URL del proyecto
- Documentación del proyecto

Nuevas características y plan de desarrollo de Amazon EKS

Para obtener más información sobre las nuevas características de Amazon EKS, desplácese hasta la fuente en la página Novedades de AWS. También puede revisar el plan de desarrollo en GitHub, que le permite conocer las próximas características y prioridades para poder planificar cómo desea utilizar Amazon EKS en el futuro. Puede proporcionarnos comentarios directos sobre las prioridades del plan de desarrollo.

Historial de revisión de Amazon EKS

En la siguiente tabla se describen las principales actualizaciones y nuevas características de la Guía del usuario de Amazon EKS. Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envía.

Cambio	Descripción	Fecha
Versión Kubernetes `1.30`	Se ha agregado compatibilidad con la versión de Kubernetes `1.30` para nuevas actualizaciones de versión y clústeres.	23 de mayo de 2024
Actualización de la versión de la plataforma de Amazon EKS	Se trata de una nueva versión de la plataforma con mejoras y correcciones de seguridad. Esto incluye las nuevas versiones de parches de Kubernetes `1.29.4`, `1.28.9` y `1.27.13`.	14 de mayo de 2024
Escalado automático de CoreDNS	El escalador automático CoreDNS adaptará dinámicamente la cantidad de réplicas de la implementación de CoreDNS en un clúster de EKS en función de la cantidad de nodos y núcleos de CPU. Esta característica funciona en CoreDNS `v1.9` y en la versión de la plataforma de lanzamiento de EKS `1.25` y versiones superiores.	14 de mayo de 2024

Soporte de CloudWatch Container Insights para Windows	Ahora, el complemento Amazon CloudWatch Observability Operator también admite Container Insights en nodos de trabajo de Windows en el clúster.	10 de abril de 2024
Conceptos de Kubernetes	Se ha agregado un nuevo tema sobre los conceptos de Kubernetes.	5 de abril de 2024
Reestructuración del acceso y el contenido de IAM	Mueva las páginas existentes relacionadas con temas de acceso e IAM, como el mapa de configuración de autenticación, las entradas de acceso, el ID de pod y el IRSA, a una nueva sección. Revise el contenido de la descripción general.	2 de abril de 2024
Bottlerocket Soporte del sistema operativo para el controlador de CSI de Amazon S3	El Mountpoint para el controlador CSI de Amazon S3 ahora es compatible con Bottlerocket.	13 de marzo de 2024
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	4 de marzo de 2024

Amazon Linux 2023	Amazon Linux 2023 (AL2023) es un nuevo sistema operativo basado en Linux diseñado para proporcionar un entorno seguro, estable y de alto rendimiento para las aplicaciones en la nube.	29 de febrero de 2024
La Pod Identity de EKS e IRSA admiten asociados en Kubernetes 1.29	En Kubernetes 1.29, los contenedores asociados están disponibles en los clústeres de Amazon EKS. Los contenedores asociados son compatibles con los roles de IAM para las cuentas de servicio o Pod Identity de EKS. Para más información sobre estos contenedores, consulte Contenedores asociados en la documentación de Kubernetes.	26 de febrero de 2024
Versión Kubernetes 1.29	Se ha agregado compatibilidad con la versión de Kubernetes 1.29 para nuevas actualizaciones de versión y clústeres.	23 de enero de 2024
Versión completa: soporte extendido de Amazon EKS para las versiones de Kubernetes	El soporte extendido de las versiones de Kubernetes permite permanecer en una versión específica de Kubernetes durante más de 14 meses.	16 de enero de 2024

Detección del estado del clúster de Amazon EKS en la Nube de AWS	Amazon EKS detecta problemas con los clústeres de Amazon EKS y con la infraestructura de los requisitos previos del clúster en el estado del clúster. Puede ver los problemas relacionados con sus clústeres de EKS dentro de la AWS Management Console y en el `health` del clúster en la API de EKS. Estos problemas se suman a los problemas detectados y mostrados por la consola. Anteriormente, el estado del clúster solo estaba disponible para los clústeres locales en AWS Outposts.	28 de diciembre de 2023
Ampliación de la Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS del Oeste de Canadá (Calgary) (`ca-west-1`).	20 de diciembre de 2023
Información sobre el clúster	Ahora puede obtener recomendaciones sobre su clúster en función de las comprobaciones periódicas.	20 de diciembre de 2023

Ahora puede conceder a los usuarios y roles de IAM el acceso a su clúster mediante las entradas de acceso	Antes de introducir las entradas de acceso, usted concedió a los usuarios y roles de IAM el acceso a su clúster añadiendo entradas a `aws-auth ConfigMap` . Ahora, cada clúster tiene un modo de acceso y puede cambiar a utilizar las entradas de acceso según su horario. Tras cambiar de modo, puede añadir usuarios agregando entradas de acceso en la AWS CLI, el AWS CloudFormation y en los SDK de AWS.	18 de diciembre de 2023
Actualización de la versión de la plataforma de Amazon EKS	Se trata de una nueva versión de la plataforma con mejoras y correcciones de seguridad . Esto incluye las nuevas versiones de parches de Kubernetes `1.28.4`, `1.27.8`, `1.26.11` y `1.25.16`.	12 de diciembre de 2023
Mountpoint para el controlador CSI de Amazon S3	Ahora puede instalar el Mountpoint para el controlador CSI de Amazon S3 en los clústeres de Amazon EKS.	27 de noviembre de 2023
Active las métricas de Prometheus al crear un clúster	En la AWS Management Console, ahora puede activar las métricas de Prometheus al crear un clúster. También puede ver los detalles del raspador Prometheus en la pestaña Observabilidad.	26 de noviembre de 2023

Pod Identities de Amazon EKS	Pod Identities de Amazon EKS asocian el rol de IAM con una cuenta de servicio de Kubernetes. Con esta característica, ya no es necesario proporcionar permisos extendidos al rol de IAM del nodo. De esta forma, los Pods de ese nodo pueden llamar a las API AWS. A diferencia de los roles de IAM para cuentas de servicio, Pod Identities de EKS están completamente integrada s en EKS; no necesita un proveedor de identidades OIDC.	26 de noviembre de 2023
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualiza do una política administrada existente de AWS.	26 de noviembre de 2023
Controlador de instantáneas CSI	Ahora puede instalar el controlador de instantáneas CSI para usarlo con controlad ores CSI compatibles, como el controlador CSI de Amazon EBS.	17 de noviembre de 2023

Reescritura del tema ADOT Operator	El soporte del componente de Amazon EKS para la sección ADOT Operator era redundante con la AWS Distro de la documentación de OpenTelemetry. Migramos la información esencial restante a ese recurso para reducir la información obsoleta e incoherente.	14 de noviembre de 2023
Compatibilidad del componente CoreDNS EKS para las métricas de Prometheus	Las versiones `v1.10.1-eksbuild.5` , `v1.9.3-eksbuild.9` y `v1.8.7-eksbuild.8` del complemento EKS para CoreDNS exponen el puerto en el que CoreDNS publicó las métricas, en el servicio `kube-dns`. Esto facilita la inclusión de las métricas de CoreDNS en sus sistemas de monitoreo.	10 de noviembre de 2023
Complemento Amazon EKS CloudWatch Observability Operator	Se agregó la página Amazon EKS CloudWatch Observability Operator.	6 de noviembre de 2023
Bloques de capacidad para instancias P5 autoadministradas en la región Este de EE. UU. (Ohio)	En la región Este de EE. UU. (Ohio), ahora puede utilizar bloques de capacidad para instancias P5 autoadministradas.	31 de octubre de 2023

Los clústeres permiten modificar las subredes y los grupos de seguridad	Puede actualizar el clúster para cambiar las subredes y los grupos de seguridad que utiliza el clúster. Puede actualizar desde la AWS Management Console, la última versión de la AWS CLI, AWS CloudFormation, y `eksctl` versión `v0.164.0-rc.0` o posterior. Es posible que tenga que hacer esto para proporcionar a las subredes más direcciones IP disponibles con las que poder actualizar correctamente la versión de un clúster.	24 de octubre de 2023
El rol del clúster y el rol del grupo de nodos administrados permiten políticas de AWS Identity and Access Management administradas por el cliente	Puede utilizar una política de IAM personalizada en el rol del clúster en lugar de la política administrada de AWS `AmazonEKSClusterPolicy`. Además, puede usar una política de IAM personalizada en el rol del nodo de un grupo de nodos administrados en lugar de la política administrada de AWS `AmazonEKSWorkerNodePolicy`. Haga esto para crear una política con el privilegio mínimo para cumplir con exigencias de conformidad estrictas.	23 de octubre de 2023

Corregido el enlace a la instalación de eksctl	Corregido el enlace de instalación de eksctl después de trasladar la página.	6 de octubre de 2023
Versión preliminar: soporte extendido de Amazon EKS para las versiones de Kubernetes	El soporte extendido de las versiones de Kubernete s permite permanecer en una versión específica de Kubernetes durante más de 14 meses.	4 de octubre de 2023
Eliminación de las referencias a la integración de AWS App Mesh	Las integraciones de Amazon EKS con AWS App Mesh solo se conservarán para los clientes actuales de App Mesh.	29 de septiembre de 2023
Versión Kubernetes `1.28`	Se ha agregado compatibi lidad con la versión de Kubernetes `1.28` para nuevas actualizaciones de versión y clústeres.	26 de septiembre de 2023
Los clústeres existentes admiten la aplicación de la política de red de Kubernetes en el Amazon VPC CNI plugin for Kubernetes	Puede utilizar la política de red de Kubernetes en los clústeres existentes con el Amazon VPC CNI plugin for Kubernetes, en lugar de tener que usar una solución de terceros.	15 de septiembre de 2023

El complemento CoreDNS de Amazon EKS admite la modificación de PDB	Puede modificar el objeto `PodDisruptionBudget` del complemento CoreDNS de EKS en las versiones `v1.9.3-eksbuild.7` y posteriores, así como `v1.10.1-eksbuild.4` y posteriores.	15 de septiembre de 2023
Soporte de Amazon EKS para subredes compartidas	Nuevos Requisitos y consideraciones sobre las subredes compartidas para crear clústeres de Amazon EKS en subredes compartidas.	7 de septiembre de 2023
Actualizaciones de ¿Qué es Amazon EKS?	Se agregaron los temas nuevos Casos de uso comunes y Arquitectura. Se actualizaron otros temas.	6 de septiembre de 2023
Aplicación de la política de red de Kubernetes en el Amazon VPC CNI plugin for Kubernetes	Puede utilizar la política de red de Kubernetes con el Amazon VPC CNI plugin for Kubernetes, en lugar de requerir una solución de terceros.	29 de agosto de 2023
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en Israel (Tel Aviv) (`il-central-1`) Región de AWS.	1 de agosto de 2023
Almacenamiento efímero configurable para Fargate	Puede aumentar la cantidad total de almacenamiento efímero para cada Pod que se ejecuta en Amazon EKS Fargate.	31 de julio de 2023

Compatibilidad del complemento para el controlador de CSI de Amazon EFS	A partir de ahora, puede utilizar la AWS Management Console, la AWS CLI y la API para administrar el controlador de CSI de Amazon EFS.	26 de julio de 2023
Actualizaciones de políticas administradas de AWS: nueva política	Amazon EKS ha agregado una nueva política administrada de AWS.	26 de julio de 2023
Las actualizaciones de Kubernetes de las versiones 1.27, 1.26, 1.25 y 1.24 ya están disponibles para los clústeres locales en AWS Outposts	Las actualizaciones de Kubernetes de las versiones 1.27.3, 1.26.6, 1.25.11 y 1.24.15 ya están disponibles para los clústeres locales en AWS Outposts	20 de julio de 2023
Soporte de prefijos IP para nodos de Windows	La asignación de prefijos IP a los nodos puede permitirle alojar un número significativamente mayor de servidores Pods en sus nodos que al asignar direcciones IP secundarias individuales a sus nodos.	6 de julio de 2023
Controlador de CSI de Amazon FSx para OpenZFS	Ahora puede instalar el controlador CSI de Amazon FSx para OpenZFS en los clústeres de Amazon EKS.	30 de junio de 2023
Los nodos Pods en Linux en clústeres `IPv4` ahora pueden comunicarse con puntos de conexión`IPv6`.	Después de asignar una dirección IPv6 a su nodo, su Pods «dirección `IPv4`» es la dirección de red traducida a la dirección `IPv6` del nodo en el que se ejecuta.	19 de junio de 2023

Grupos de nodos administrados de Windows en AWS GovCloud (US) Regions	En AWS GovCloud (US) Regions, los grupos de nodos gestionados de Amazon EKS ahora pueden ejecutar contenedores de Windows.	30 de mayo de 2023
Versión Kubernetes `1.27`	Se ha agregado compatibilidad con la versión de Kubernetes `1.27` para nuevas actualizaciones de versión y clústeres.	24 de mayo de 2023
Versión Kubernetes `1.26`	Se ha agregado compatibilidad con la versión de Kubernetes `1.26` para nuevas actualizaciones de versión y clústeres.	11 de abril de 2023
gMSA sin dominio	Ahora puede usar gMSA sin dominio con Pods de Windows.	27 de marzo de 2023
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Asia-Pacífico (Melbourne) (`ap-southeast-4`).	10 de marzo de 2023
Controlador CSI de Amazon File Cache	Ahora puede instalar el controlador CSI de Amazon File Cache en los clústeres de Amazon EKS.	3 de marzo de 2023
La versión 1.25 de Kubernetes ya está disponible para clústeres locales en AWS Outposts.	Ahora puede crear un clúster local de Amazon EKS en un Outpost mediante las versiones de Kubernetes de la `1.22` a la `1.25`.	1 de marzo de 2023

Versión Kubernetes 1.25	Se ha agregado compatibilidad con la versión de Kubernetes 1.25 para nuevas actualizaciones de versión y clústeres.	22 de febrero de 2023
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	7 de febrero de 2023
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en Asia Pacífico (Hyderabad) (ap-south-2), Europa (Zúrich) (eu-central-2) y Europa (España) (eu-south-2) Regiones de AWS.	6 de febrero de 2023
Las versiones de Kubernetes de la 1.21 a la 1.24 ahora están disponibles para los clústeres locales en AWS Outposts.	Ahora puede crear un clúster local de Amazon EKS en un Outpost mediante las versiones de Kubernetes de la 1.21 a la 1.24. Anteriormente, solo estaba disponible la versión 1.21.	17 de enero de 2023
Amazon EKS ahora admite AWS PrivateLink	Puede usar un AWS PrivateLink para crear una conexión privada entre la VPC y Amazon EKS.	16 de diciembre de 2022
Soporte de Windows para grupos de nodos administrados	Ahora puede usar Windows para grupos de nodos administrados por Amazon EKS.	15 de diciembre de 2022

Los complementos de Amazon EKS de proveedores de software independientes ya están disponibles en AWS Marketplace	Ahora puede buscar y suscribirse a los complementos de Amazon EKS de proveedores de software independientes a través del AWS Marketplace.	28 de noviembre de 2022
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	17 de noviembre de 2022
Versión Kubernetes `1.24`	Se ha agregado compatibilidad con la versión de Kubernetes `1.24` para nuevas actualizaciones de versión y clústeres.	15 de noviembre de 2022
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Medio Oriente (EAU) (`me-central-1`).	3 de noviembre de 2022
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	24 de octubre de 2022
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	20 de octubre de 2022
Ya están disponibles los clústeres locales en AWS Outposts	Ahora puede crear un clúster local de Amazon EKS en un Outpost.	19 de septiembre de 2022

Cuotas basadas en vCPU de Fargate	Fargate pasa de cuotas basadas en Pod a cuotas basadas en vCPU.	8 de septiembre de 2022
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	31 de agosto de 2022
Supervisión de costos	Amazon EKS ya admite Kubecost, lo que le permite supervisar los costos desglosados por recursos de Kubernetes, que incluyen Pods, nodos, espacios de nombres y etiquetas.	24 de agosto de 2022
Actualizaciones de políticas administradas de AWS: nueva política	Amazon EKS ha agregado una nueva política administrada de AWS.	24 de agosto de 2022
Actualizaciones de políticas administradas de AWS: nueva política	Amazon EKS ha agregado una nueva política administrada de AWS.	23 de agosto de 2022
Recursos de etiquetas para facturación	Se agregó el soporte de la etiqueta de asignación de costos generada por `aws:eks:cluster-name` para todos los clústeres.	16 de agosto de 2022
Comodines de perfil de Fargate	Se agregó soporte para los comodines de perfil de Fargate en los criterios del selector para los espacios de nombres, las claves de las etiquetas y los valores de las etiquetas.	16 de agosto de 2022

Versión Kubernetes 1.23	Se ha agregado compatibilidad con la versión de Kubernetes 1.23 para nuevas actualizaciones de versión y clústeres.	11 de agosto de 2022
Vea los recursos Kubernetes en la AWS Management Console	Ahora puede ver la información sobre los recursos Kubernetes desplegados en el clúster mediante el AWS Management Console.	3 de mayo de 2022
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Asia-Pacífico (Yakarta) (ap-southeast-3).	2 de mayo de 2022
Compatibilidad con la página de observabilidad y el complemento ADOT	Se agregó la página Observabilidad y AWS Distro for OpenTelemetry (ADOT).	21 de abril de 2022
Versión Kubernetes1.22	Se ha agregado compatibilidad con la versión de Kubernetes 1.22 para nuevas actualizaciones de versión y clústeres.	4 de abril de 2022
Actualizaciones de políticas administradas de AWS: nueva política	Amazon EKS ha agregado una nueva política administrada de AWS.	4 de abril de 2022

Se agregaron detalles de los parches de Pod de Fargate	Al actualizar los Pods de Fargate, Amazon EKS intenta expulsar los Pods según los presupuestos de interrupción de los Pod. Puede crear reglas de eventos para reaccionar ante las expulsiones fallidas antes de que se eliminen los Pods.	1 de abril de 2022
Lanzamiento completo: compatibilidad del complemento para el controlador de CSI de Amazon EBS	A partir de ahora, puede utilizar la AWS Management Console, la AWS CLI y la API para administrar el controlador de CSI de Amazon EBS.	31 de marzo de 2022
Actualización del contenido AWS Outposts	Instrucciones para implementar un clúster de Amazon EKS en AWS Outposts.	22 de marzo de 2022
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	21 de marzo de 2022
Compatibilidad con Windows `containerd`	Ahora puede seleccionar el tiempo de funcionamiento `containerd` para nodos de Windows.	14 de marzo de 2022
Se agregaron consideraciones sobre Amazon EKS Connector a la documentación de seguridad	Describe el modelo de responsabilidad compartida en lo que se refiere a los clústeres conectados.	25 de febrero de 2022

Asigne direcciones `IPv6` a los Pods y los servicios	Ahora, puede crear un clúster `1.21` o posterior que asigne direcciones `IPv6` a sus Pods y servicios.	6 de enero de 2022
Actualizaciones de política administrada de AWS: actualización de una política existente	Amazon EKS ha actualizado una política administrada existente de AWS.	13 de diciembre de 2021
Lanzamiento previo: compatibilidad del complemento para el controlador de CSI de Amazon EBS	A partir de ahora, puede hacer una vista previa de la AWS Management Console, la AWS CLI y la API para administrar el controlador de CSI de Amazon EBS.	9 de diciembre de 2021
Compatibilidad con el escalador automático de Karpenter	Ahora puede utilizar el proyecto de código abierto Karpenter para escalar los nodos de forma automática.	29 de noviembre de 2021
Compatibilidad con filtros de Fluent Bit de Kubernetes en el registro de Fargate	Ahora puede utilizar el filtro de Fluent Bit de Kubernetes con el registro de Fargate.	10 de noviembre de 2021
Compatibilidad con Windows disponible en el plano de control	La compatibilidad con Windows ahora se encuentra disponible en el plano de control. Ya no tendrá que habilitarla en su plano de datos.	9 de noviembre de 2021

Se ha agregado Bottlerocket como tipo de AMI para grupos de nodos administrados	Anteriormente, Bottlerocket solo estaba disponible como opción de nodo autoadmin istrado. Ahora se puede configurar como un grupo de nodos administrados, lo que reduce el esfuerzo necesario para cumplir los requisitos de conformidad de nodos.	28 de octubre de 2021
Compatibilidad con controladores DL1	Las AMI personalizadas de Amazon Linux ahora admiten cargas de trabajo de aprendizaje profundo para Linux 2 de Amazon. Esta habilitación permite una configuración genérica de base de referencia en las instalaciones o en la nube.	25 de octubre de 2021
Soporte de vídeo VT1	Las AMI personalizadas de Amazon Linux ahora admiten VT1 para algunas distribuciones. Esta habilitación anuncia dispositivos Xilinx U30 en su clúster de Amazon EKS.	13 de septiembre de 2021
Amazon EKS Connector ya está disponible	Puede utilizar Amazon EKS Connector a fin de registrar y conectar cualquier clúster de Kubernetes conforme a AWS y visualizarlo en la consola de Amazon EKS.	8 de septiembre de 2021

Amazon EKS Anywhere ya está disponible	Amazon EKS Anywhere es una nueva opción de implementación para Amazon EKS que puede usar para crear y operar fácilmente clústeres de Kubernetes en las instalaciones.	8 de septiembre de 2021
Controlador de CSI de Amazon FSx para ONTAP de NetApp	Se ha agregado un tema que resume el controlador de CSI de Amazon FSx para ONTAP de NetApp y proporciona enlaces a otras referencias.	2 de septiembre de 2021
Los grupos de nodos administrados calculan automáticamente el número máximo de Pods recomendados por Amazon EKS para los nodos	Los grupos de nodos administrados ahora calculan automáticamente el número máximo de Pods de Amazon EKS para los nodos que implementa sin una plantilla de lanzamiento o con una plantilla de lanzamiento en la que no ha especificado un ID de AMI.	30 de agosto de 2021
Eliminación de la administración de Amazon EKS de la configuración del complemento sin eliminar el software del complemento de Amazon EKS	Ahora puede eliminar un complemento de Amazon EKS sin quitar el software del complemento del clúster.	20 de agosto de 2021
Cree Pods de múltiples anfitriones mediante Multus	Ahora puede agregar varias interfaces de red a un Pod mediante Multus.	2 de agosto de 2021

Agregue más direcciones IP a los nodos Linux de Amazon EC2	Ahora puede agregar una cantidad significativa de direcciones IP más a los nodos de Amazon EC2 de Linux. Esto significa que puede ejecutar una mayor densidad de Pods en cada nodo.	27 de julio de 2021
Arranque en tiempo de ejecución de `containerd`	La Amazon Machine Image (AMI) acelerada, optimizada para Amazon EKS de Amazon Linux, ahora contiene un indicador de arranque para habilitar opcionalmente el tiempo de ejecución de en las `containerd` optimizadas para Amazon EKS y las AMI de Bottlerocket. Este indicador está disponible en todas las versiones de Kubernetes compatibles de la AMI.	19 de julio de 2021
Versión Kubernetes`1.21`	Se agregó la compatibilidad de la versión Kubernetes `1.21`.	19 de julio de 2021
Se ha agregado un tema de políticas administradas	Una lista de todas las políticas administradas por IAM de Amazon EKS y los cambios realizados en ellas desde el 17 de junio de 2021.	17 de junio de 2021
Use grupos de seguridad para Pods con Fargate	Ahora puede utilizar grupos de seguridad para Pods con Fargate, además de utilizarlos con nodos de Amazon EC2.	1 de junio de 2021

Se han agregado complementos de CoreDNS y `kube-proxy` de Amazon EKS	Amazon EKS ahora puede ayudarlo a administrar los complementos de CoreDNS y `kube-proxy` de Amazon EKS para su clúster.	19 de mayo de 2021
Versión Kubernetes `1.20`	Se ha agregado compatibilidad con la versión de Kubernetes `1.20` para nuevas actualizaciones de versión y clústeres.	18 de mayo de 2021
Se ha lanzado AWS Load Balancer Controller `2.2.0`	A partir de ahora, puede utilizar el AWS Load Balancer Controller para crear equilibradores de carga elástica mediante instancias o destinos IP.	14 de mayo de 2021
Taints de nodos para grupos de nodos administrados	Amazon EKS ahora admite la agregación de taints de notas a los grupos de nodos administrados.	11 de mayo de 2021
Cifrado de secretos para clústeres existentes	Amazon EKS ahora admite la adición de cifrado de secretos a los clústeres existentes.	26 de febrero de 2021
Versión Kubernetes `1.19`	Se ha agregado compatibilidad con la versión de Kubernetes `1.19` para nuevas actualizaciones de versión y clústeres.	16 de febrero de 2021

Amazon EKS ahora admite la utilización de proveedores de identidad OpenID Connect (OIDC) como método para autenticar usuarios de la versión `1.16` o posterior.	Los proveedores de identidad de OIDC se pueden utilizar con AWS Identity and Access Management (IAM) o como una alternativa de IAM.	12 de febrero de 2021
Vea los recursos de nodo y carga de trabajo en la AWS Management Console	Ahora puede ver detalles sobre los nodos administr ados, autoadministrados y de Fargate, así como las cargas de trabajo de Kubernetes implementadas en la AWS Management Console.	1 de diciembre de 2020
Implemente tipos de instancias de spot en un grupo de nodos administrado	Ahora puede implementar varios tipos de instancias de spot o instancias bajo demanda en un grupo de nodos administrado.	1 de diciembre de 2020
Amazon EKS ahora puede administrar complementos específicos para su clúster	Puede administrar los complementos usted mismo o dejar que Amazon EKS controle el lanzamiento y la versión de un complemento a través de la API de Amazon EKS.	1 de diciembre de 2020
Comparta un ALB en varias entradas	A partir de ahora, puede compartir un equilibrador de carga de aplicación (ALB) de AWS entre múltiples entradas de Kubernetes. En el pasado, tenía que implementar un ALB independiente para cada entrada.	23 de octubre de 2020

Compatibilidad con destino IP NLB	Ahora puede implementar un equilibrador de carga de red con destinos IP. Esto significa que puede utilizar un NLB para equilibrar la carga del tráfico de red a los Pods de Fargate y directamente a los Pods que se ejecutan en nodos de Amazon EC2.	23 de octubre de 2020
Versión Kubernetes`1.18`	Se ha agregado compatibilidad con la versión de Kubernetes `1.18` para nuevas actualizaciones de versión y clústeres.	13 de octubre de 2020
Especifique un bloque de CIDR personalizado para la asignación de direcciones IP del servicio Kubernetes.	Ahora puede especificar un bloque de CIDR personalizado desde el que Kubernetes asigna direcciones IP de servicio.	29 de septiembre de 2020
Asigne los grupos de seguridad a Pods individuales	Ahora puede asociar diferentes grupos de seguridad a algunos de los Pods individuales que se ejecutan en muchos tipos de instancias de Amazon EC2.	9 de septiembre de 2020
Implemente Bottlerocket en sus nodos	Ahora puede implementar nodos que ejecuten Bottlerocket.	31 de agosto de 2020
La capacidad de lanzar nodos de Arm normalmente está disponible	Ahora puede lanzar nodos de Arm en grupos de nodos administrados y autoadministrados.	17 de agosto de 2020

Plantillas de lanzamiento de grupos de nodos administrados y AMI personalizadas	Ahora puede implementar un grupo de nodos administrado mediante una plantilla de lanzamiento de Amazon EC2. Si lo desea, la plantilla de lanzamiento puede especificar una AMI personalizada.	17 de agosto de 2020
Compatibilidad de EFS con AWS Fargate	Ahora puede utilizar Amazon EFS con AWS Fargate.	17 de agosto de 2020
Actualización de la versión de la plataforma de Amazon EKS	Se trata de una nueva versión de la plataforma con mejoras y correcciones de seguridad. Esto incluye soporte UDP para servicios de tipo `LoadBalancer` cuando se utilizan equilibradores de carga de red con la versión `1.15` de Kubernetes o posterior. Para obtener más información, consulte Permitir UDP para el equilibrador de carga de red de AWS en GitHub.	12 de agosto de 2020
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en las Regiones de AWS África (Ciudad del Cabo) (`af-south-1`) y Europa (Milán) (`eu-south-1`).	6 de agosto de 2020
Métricas de uso de Fargate	AWS Fargate incorpora métricas de uso de CloudWatch que proporcionan visibilidad del uso de recursos bajo demanda de Fargate de su cuenta.	3 de agosto de 2020

Versión Kubernetes1.17	Se ha agregado compatibilidad con la versión de Kubernetes 1.17 para nuevas actualizaciones de versión y clústeres.	10 de julio de 2020
Cree y administre recursos de App Mesh desde Kubernetes con el controlador de App Mesh para Kubernetes	Puede crear y administrar recursos de App Mesh desde Kubernetes. El controlador también inyecta automáticamente el proxy de Envoy y los contenedores init en los Pods que se implementan.	18 de junio de 2020
Amazon EKS ahora admite nodos Inf1 de Amazon EC2	Puede agregar nodos Inf1 de Amazon EC2 al clúster.	4 de junio de 2020
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en las Regiones de AWS de GovCloud (EE. UU. Este) de AWS (`us-gov-east-1`) y GovCloud (EE. UU. Oeste) de AWS (`us-gov-west-1`).	13 de mayo de 2020
Kubernetes1.12 ya no se admite en Amazon EKS	La versión Kubernetes 1.12 ya no se admite en Amazon EKS. Actualice todos los clústeres 1.12 a la versión 1.13 o posterior para evitar la interrupción del servicio.	12 de mayo de 2020
Versión Kubernetes1.16	Se ha agregado compatibilidad con la versión de Kubernetes 1.16 para nuevas actualizaciones de versión y clústeres.	30 de abril de 2020

Se ha agregado el rol vinculado a servicios AWSServiceRoleForAmazonEKS	Se ha agregado el rol vinculado al servicio de AWSServiceRoleForAmazonEKS.	16 de abril de 2020
Versión Kubernetes1.15	Se ha agregado compatibilidad con la versión de Kubernetes 1.15 para nuevas actualizaciones de versión y clústeres.	10 de marzo de 2020
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ahora está disponible en las Regiones de AWS Beijing (cn-north-1) y Ningxia (cn-northwest-1).	26 de febrero de 2020
Controlador de CSI de FSx para Lustre	Se agregó el tema para instalar el controlador de CSI de FSx para Lustre en los clústeres de Amazon EKS de Kubernetes 1.14.	23 de diciembre de 2019
Restrinja el acceso a la red al punto de conexión de acceso público de un clúster	Con esta actualización, puede utilizar Amazon EKS para restringir los rangos de CIDR que pueden comunicarse con el punto de conexión de acceso público del servidor de API de Kubernetes.	20 de diciembre de 2019
Resuelva la dirección de punto de conexión de acceso privado para un clúster desde fuera de una VPC	Con esta actualización, puede utilizar Amazon EKS para resolver el punto de conexión de acceso privado del servidor de API de Kubernetes desde fuera de una VPC.	13 de diciembre de 2019

(Beta) Nodos de instancia de Amazon EC2 de Amazon EC2 A1	Lanzamiento de nodos de instancia de Amazon EC2 de Amazon EC2 A1 que se registran con su clúster de Amazon EKS.	4 de diciembre de 2019
Creación de un clúster en Outposts de AWS	Amazon EKS ahora admite la creación de clústeres en AWS Outposts.	3 de diciembre de 2019
AWS Fargate en Amazon EKS	Los clústeres de Kubernetes de Amazon EKS ahora admiten la ejecución de Pods en Fargate.	3 de diciembre de 2019
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Canadá (centro) (`ca-central-1`).	21 de noviembre de 2019
Grupos de nodos administrados	Los grupos de nodos administrados por Amazon EKS automatizan el aprovisionamiento y la administración del ciclo de vida de nodos (instancias de Amazon EC2) para clústeres de Kubernetes de Amazon EKS.	18 de noviembre de 2019
Actualización de la versión de la plataforma de Amazon EKS	Nuevas versiones de la plataforma como respuesta a `CVE-2019-11253` .	6 de noviembre de 2019

Kubernetes1.11 ya no se admite en Amazon EKS	La versión Kubernetes 1.11 ya no se admite en Amazon EKS. Actualice todos los clústeres 1.11 a la versión 1.12 o posterior para evitar la interrupción del servicio.	4 de noviembre de 2019
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS América del Sur (São Paulo) (sa-east-1).	16 de octubre de 2019
Windows compatibilidad	Los clústeres de Amazon EKS que ejecutan la versión 1.14 de Kubernetes ya admiten las cargas de trabajo de Windows.	7 de octubre de 2019
Escalado automático	Se ha agregado un capítulo para tratar algunos de los diferentes tipos de escalado automático de Kubernetes compatibles con los clústeres de Amazon EKS.	30 de septiembre de 2019
Actualización del panel de Kubernetes	Se ha actualizado el tema para instalar el panel de Kubernetes en clústeres de Amazon EKS para utilizar la versión beta 2.0.	28 de septiembre de 2019
Controlador de CSI de Amazon EFS	Se ha agregado el tema para instalar el controlador CSI de Amazon EFS en clústeres de Amazon EKS de Kubernetes 1.14.	19 de septiembre de 2019

Parámetro de Amazon EC2 Systems Manager para el ID de AMI optimizada para Amazon EKS	Se ha agregado un tema para recuperar el ID de AMI optimizada para Amazon EKS mediante un parámetro de Amazon EC2 Systems Manager. El parámetro elimina la necesidad de buscar los ID de AMI.	18 de septiembre de 2019
Etiquetado de recursos de Amazon EKS	Puede administrar el etiquetado de los clústeres de Amazon EKS.	16 de septiembre de 2019
Controlador de CSI de Amazon EBS	Se ha agregado el tema para instalar el controlador CSI de Amazon EBS en clústeres de Amazon EKS de Kubernetes `1.14`.	9 de septiembre de 2019
Nueva AMI optimizada para Amazon EKS con parches para `CVE-2019-9512` y `CVE-2019-9514`	Amazon EKS ha actualizado la AMI optimizada para Amazon EKS para abordar `CVE-2019-9512` y `CVE-2019-9514`.	6 de septiembre de 2019
Anuncio de baja de Kubernetes `1.11` en Amazon EKS	Amazon EKS retiró la compatibilidad con la versión `1.11` de Kubernetes el 4 de noviembre de 2019.	4 de septiembre de 2019
Versión Kubernetes `1.14`	Se ha agregado compatibilidad con la versión de Kubernetes `1.14` para nuevas actualizaciones de versión y clústeres.	3 de septiembre de 2019

Roles de IAM para cuentas de servicio	Con los roles de IAM de las cuentas de servicio en clústeres de Amazon EKS, puede asociar un rol de IAM a una cuenta de servicio de Kubernetes. Con esta característica, ya no es necesario proporcionar permisos extendidos al rol de IAM del nodo. De esta forma, los Pods de ese nodo pueden llamar a las API AWS.	3 de septiembre de 2019
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Medio Oriente (Baréin) (`me-south-1`).	29 de agosto de 2019
Actualización de la versión de la plataforma de Amazon EKS	Nuevas versiones de la plataforma como respuesta a `CVE-2019-9512` y `CVE-2019-9514`.	28 de agosto de 2019
Actualización de la versión de la plataforma de Amazon EKS	Nuevas versiones de la plataforma como respuesta a `CVE-2019-11247` y `CVE-2019-11249`.	5 de agosto de 2019
Ampliación de las regiones de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Asia-Pacífico (Hong Kong) (`ap-east-1`).	31 de julio de 2019

Kubernetes1.10 ya no se admite en Amazon EKS	La versión Kubernetes 1.10 ya no se admite en Amazon EKS. Actualice cualquier clúster 1.10 a la versión 1.11 o superior para evitar la interrupción del servicio.	30 de julio de 2019
Se ha agregado un tema sobre el controlador de entrada de ALB	El controlador de entrada de ALB de AWS para Kubernetes desencadena la creación de un ALB cuando se crean los recursos de entrada.	11 de julio de 2019
Nueva AMI optimizada para Amazon EKS	Eliminación de kubectl binarios de las AMI.	3 de julio de 2019
Versión Kubernetes1.13	Se ha agregado compatibilidad con la versión de Kubernetes 1.13 para nuevas actualizaciones de versión y clústeres.	18 de junio de 2019
Nueva AMI optimizada para Amazon EKS con parches para AWS-2019-005	Amazon EKS ha actualizado la AMI optimizada para Amazon EKS para resolver los problemas de vulnerabilidad descritos en AWS-2019-005 .	17 de junio de 2019
Anuncio de retirada de la compatibilidad de Kubernetes1.10 en Amazon EKS	Amazon EKS dejó de admitir la versión 1.10 de Kubernetes el 22 de julio de 2019.	21 de mayo de 2019

Actualización de la versión de la plataforma de Amazon EKS	Nueva versión de la plataforma para los clústeres de Kubernetes `1.11` y `1.10` para admitir nombres de DNS personalizados en el certificado de `kubelet` y mejorar el rendimiento de `etcd`.	21 de mayo de 2019
Comando de la AWS CLI `get-token`	El comando `aws eks get-token` se ha agregado a la AWS CLI. Ya no necesita instalar el autenticador de IAM de AWS para Kubernetes para crear tokens de seguridad del cliente para la comunicación del servidor de API del clúster. Actualice la instalación de la AWS CLI a la versión más reciente para usar esta nueva funcionalidad. Para obtener más información, consulte Installing the AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface.	10 de mayo de 2019
Introducción a `eksctl`	En esta guía de introducción se describe cómo puede instalar todos los recursos necesarios para comenzar a utilizar Amazon EKS con `eksctl`. Se trata de una utilidad sencilla de línea de comandos para crear y administrar clústeres de Kubernetes en Amazon EKS.	10 de mayo de 2019

Actualización de la versión de la plataforma de Amazon EKS	Nueva versión de la plataforma para los clústeres de Kubernetes `1.12` para admitir nombres de DNS personalizados en el certificado de `kubelet` y mejorar el rendimiento de `etcd`. Esto corrige un error que provocaba que los daemons de `kubelet` de nodos solicitaran un certificado nuevo cada pocos segundos.	8 de mayo de 2019
Tutorial de Prometheus	Se ha agregado un tema para implementar Prometheus en su clúster de Amazon EKS.	5 de abril de 2019
Registro de plano de control de Amazon EKS	Con esta actualización, puede obtener registros de auditoría y diagnóstico directamente desde el panel de control de Amazon EKS. Puede utilizar estos CloudWatch Logs en su cuenta como referencia para proteger y ejecutar clústeres.	4 de abril de 2019
Versión Kubernetes`1.12`	Se ha agregado compatibilidad con la versión de Kubernetes `1.12` para nuevas actualizaciones de versión y clústeres.	28 de marzo de 2019
Se ha agregado la guía de introducción a App Mesh	Se ha agregado documentación de introducción a App Mesh y Kubernetes.	27 de marzo de 2019

Acceso privado al punto de conexión del servidor de la API de Amazon EKS	Se ha agregado documentación para deshabilitar el acceso público al punto de conexión del servidor de la API de Kubernetes del clúster de Amazon EKS.	19 de marzo de 2019
Se ha agregado un tema para instalar el servidor de métricas de Kubernetes	El servidor de métricas de Kubernetes es un agregador de datos de uso de recursos en el clúster.	18 de marzo de 2019
Se ha agregado la lista de proyectos de código abierto relacionados	Estos proyectos de código abierto extienden la funcionalidad de los clústeres de Kubernetes que se ejecutan en AWS, incluidos los clústeres administrados por Amazon EKS.	15 de marzo de 2019
Se ha agregado un tema para instalar Helm localmente	El administrador de paquetes `helm` para Kubernetes lo ayuda a instalar y administrar aplicaciones en su clúster de Kubernetes. En este tema se muestra cómo instalar y ejecutar los `helm` y `tiller`binarios localmente. De esta forma, puede instalar y administrar gráficos mediante la CLI de Helm en su sistema local.	11 de marzo de 2019

Actualización de la versión de la plataforma de Amazon EKS	Nueva versión de la plataforma que actualiza los clústeres de Kubernetes `1.11` de Amazon EKS con el nivel de parche `1.11.8` para solucionar el problema `CVE-2019-1002100`.	8 de marzo de 2019
Se ha aumentado el límite del clúster	Amazon EKS ha aumentado el número de clústeres que puede crear en una Región de AWS de 3 a 50.	13 de febrero de 2019
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en las Regiones de AWS de Europa (Londres) (`eu-west-2`), Europa (París) (`eu-west-3`) y Asia-Pacífico (Bombay) (`ap-south-1`).	13 de febrero de 2019
Nueva AMI optimizada para Amazon EKS con parches para `ALAS-2019-1156`	Amazon EKS ha actualizado la AMI optimizada para Amazon EKS a fin de resolver los problemas de vulnerabilidad descritos en `ALAS-2019-1156`.	11 de febrero de 2019
Nueva AMI optimizada para Amazon EKS con parches para `ALAS2-2019-1141`	Amazon EKS ha actualizado la AMI optimizada para Amazon EKS a fin de abordar las CVE a las que se hace referencia en `ALAS2-2019-1141`.	9 de enero de 2019
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Asia-Pacífico (Seúl) (`ap-northeast-2`).	9 de enero de 2019

Ampliación de las regiones de Amazon EKS	Amazon EKS ya está disponible en las siguientes Regiones de AWS adicionales: Europa (Fráncfort) (`eu-central-1`), Asia-Pacífico (Tokio) (`ap-northeast-1`), Asia-Pacífico (Singapur) (`ap-southeast-1`) y Asia-Pacífico (Sídney) (`ap-southeast-2`).	19 de diciembre de 2018
Actualizaciones de clúster de Amazon EKS	Se agregó documentación para las actualizaciones de la versión de Kubernetes del clúster y sustitución de nodos de Amazon EKS.	12 de diciembre de 2018
Ampliación de Región de AWS de Amazon EKS	Amazon EKS ya está disponible en la Región de AWS Europa (Estocolmo) (`eu-north-1`).	11 de diciembre de 2018
Actualización de la versión de la plataforma de Amazon EKS	Actualización de la versión de la nueva plataforma de Kubernetes al nivel de parches `1.10.11` para abordar `CVE-2018-1002105`.	4 de diciembre de 2018
Se ha agregado compatibilidad con la versión `1.0.0` del controlador de entrada de ALB	El controlador de entrada de ALB publica la versión `1.0.0` con compatibilidad formal de AWS.	20 de noviembre de 2018

Se ha agregado compatibilidad con la configuración de red de CNI	La versión `1.2.1` del Amazon VPC CNI plugin for para Kubernetes es compatible ahora con la configuración de red personalizada para las interfaces de red de Pod secundarias.	16 de octubre de 2018
Se ha agregado compatibilidad con `MutatingAdmissionWebhook` y `ValidatingAdmissionWebhook`	La versión `1.10-eks.2` de la plataforma de Amazon EKS ahora admite los controladores de admisión `MutatingAdmissionWebhook` y `ValidatingAdmissionWebhook`.	10 de octubre de 2018
Se ha agregado información de las AMI de socios	Canonical se ha asociado con Amazon EKS para crear AMI de nodo que puede utilizar en sus clústeres.	3 de octubre de 2018
Se han agregado instrucciones para el comando de la AWS CLI `update-kubeconfig`	Amazon EKS ha agregado `update-kubeconfig` a la AWS CLI para simplificar el proceso de creación de un archivo `kubeconfig` para obtener acceso al clúster.	21 de septiembre de 2018
Nuevas AMI optimizadas para Amazon EKS	Amazon EKS ha actualizado las AMI optimizadas para Amazon EKS (con y sin compatibilidad con GPU) para proporcionar varias correcciones de seguridad y optimizaciones de las AMI.	13 de septiembre de 2018

Ampliación de Región de AWS de Amazon EKS	Amazon EKS ahora está disponible en la región de Europa (Irlanda) (`eu-west-1`).	5 de septiembre de 2018
Actualización de la versión de la plataforma de Amazon EKS	Nueva versión de la plataforma compatible con la capa de agregación y Horizontal Pod Autoscaler (HPA) de Kubernetes.	31 de agosto de 2018
Nuevas AMI optimizadas para Amazon EKS y compatibilidad con GPU	Amazon EKS ha actualizado la AMI optimizada para Amazon EKS a fin de utilizar una plantilla de nodos de AWS CloudFormation y script de proceso de arranque nuevos. Además, hay disponible una nueva AMI optimizada para Amazon EKS compatible con GPU.	22 de agosto de 2018
Nueva AMI optimizada para Amazon EKS con parches para `ALAS2-2018-1058`	Amazon EKS ha actualizado la AMI optimizada para Amazon EKS a fin de abordar las CVE a las que se hace referencia en `ALAS2-2018-1058`.	14 de agosto de 2018
Scripts de compilación de la AMI optimizada para Amazon EKS	Amazon EKS ha establecido en código abierto los scripts de compilación que se utilizan para crear la AMI optimizada para Amazon EKS. Estos scripts de compilación están ahora disponibles en GitHub.	10 de julio de 2018

Versión inicial de Amazon EKS	Documentación inicial para el lanzamiento del servicio	5 de junio de 2018