



AWS ParallelCluster Guía del usuario (v2)

AWS ParallelCluster



AWS ParallelCluster: AWS ParallelCluster Guía del usuario (v2)

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

| | |
|---|----|
| ¿Qué es AWS ParallelCluster? | 1 |
| Precios | 1 |
| Configuración AWS ParallelCluster | 2 |
| Instalación AWS ParallelCluster | 2 |
| Instalación AWS ParallelCluster en un entorno virtual (recomendado) | 2 |
| Instalación AWS ParallelCluster en un entorno no virtual utilizando pip | 3 |
| Pasos que seguir después de la instalación | 3 |
| Instrucciones detalladas para cada entorno | 4 |
| Entorno virtual | 4 |
| Linux | 6 |
| macOS | 10 |
| Windows | 13 |
| Configuración AWS ParallelCluster | 15 |
| Prácticas recomendadas | 24 |
| Prácticas recomendadas: selección del tipo de instancia principal | 24 |
| Prácticas recomendadas: rendimiento de la red | 24 |
| Prácticas recomendadas: alertas de presupuesto | 25 |
| Mejores prácticas: trasladar un clúster a un nuevo AWS ParallelCluster versión secundaria o de parche | 26 |
| ¿Pasar de a CfnCluster AWS ParallelCluster | 27 |
| Regiones admitidas | 28 |
| Usando AWS ParallelCluster | 31 |
| Configuraciones de red | 31 |
| AWS ParallelCluster en una única subred pública | 32 |
| AWS ParallelCluster utilizando dos subredes | 33 |
| AWS ParallelCluster en una única subred privada conectada mediante AWS Direct Connect | 34 |
| AWS ParallelCluster con awsbatch programador | 35 |
| Acciones de arranque personalizadas | 36 |
| Configuración | 38 |
| Argumentos | 38 |
| Ejemplo | 38 |
| Uso de Amazon S3 | 40 |
| Ejemplos | 40 |

| | |
|---|-----|
| Uso de instancias de spot | 41 |
| Escenario 1: Se interrumpe una instancia de spot sin trabajos en ejecución | 41 |
| Escenario 2: Se interrumpe una instancia de spot que ejecuta trabajos de un solo nodo | 42 |
| Escenario 3: Se interrumpe una instancia de spot que ejecuta trabajos de varios nodos | 43 |
| AWS Identity and Access Management funciones en AWS ParallelCluster | 45 |
| Configuración predeterminada para la creación de clústeres | 45 |
| Uso de un IAM rol existente para Amazon EC2 | 45 |
| AWS ParallelCluster ejemplos de políticas de instancia y usuario | 46 |
| Planificadores compatibles con AWS ParallelCluster | 87 |
| Son of Grid Engine | 88 |
| Slurm Workload Manager | 88 |
| Torque Resource Manager | 101 |
| AWS Batch | 101 |
| Etiquetado | 109 |
| CloudWatch Panel de control de Amazon | 112 |
| Integración con Amazon CloudWatch Logs | 114 |
| Elastic Fabric Adapter | 116 |
| Soluciones Intel Select | 117 |
| Habilite Intel MPI | 119 |
| Especificación de plataforma Intel HPC | 120 |
| Bibliotecas de rendimiento de Arm | 121 |
| Conéctese al nodo principal a través de Amazon DCV | 123 |
| DCVHTTPSCertificado de Amazon | 124 |
| Licenciamiento Amazon DCV | 124 |
| Uso de <code>pcluster update</code> | 125 |
| AMI parcheo y EC2 reemplazo de instancias | 127 |
| Actualización o sustitución de la instancia del nodo principal | 128 |
| Limitaciones del almacén de instancias | 129 |
| Soluciones alternativas para las limitaciones del almacén de instancias | 129 |
| Detención e inicio del nodo principal de un clúster | 130 |
| AWS ParallelCluster CLI comandos | 133 |
| <code>pcluster</code> | 133 |
| Argumentos | 133 |
| Subcomandos: | 133 |
| <code>pcluster configure</code> | 134 |
| <code>pcluster create</code> | 135 |

| | |
|--------------------------------|-----|
| pcluster createami | 137 |
| pcluster dcv | 141 |
| pcluster delete | 143 |
| pcluster instances | 145 |
| pcluster list | 146 |
| pcluster ssh | 147 |
| pcluster start | 148 |
| pcluster status | 149 |
| pcluster stop | 150 |
| pcluster update | 151 |
| pcluster version | 153 |
| pcluster-config | 154 |
| Argumentos con nombre | 154 |
| Configuración | 156 |
| Diseño | 157 |
| Sección [global] | 157 |
| cluster_template | 157 |
| update_check | 158 |
| sanity_check | 158 |
| Sección [aws] | 158 |
| Sección [alias] | 159 |
| Sección de [cluster] | 160 |
| additional_cfn_template | 162 |
| additional_iam_policies | 162 |
| base_os | 163 |
| cluster_resource_bucket | 165 |
| cluster_type | 166 |
| compute_instance_type | 167 |
| compute_root_volume_size | 167 |
| custom_ami | 168 |
| cw_log_settings | 169 |
| dashboard_settings | 169 |
| dcv_settings | 170 |
| desired_vcpus | 170 |
| disable_cluster_dns | 171 |
| disable_hyperthreading | 171 |

| | |
|---------------------------------|-----|
| ebs_settings | 172 |
| ec2_iam_role | 173 |
| efs_settings | 173 |
| enable_efa | 173 |
| enable_efa_gdr | 174 |
| enable_intel_hpc_platform | 175 |
| encrypted_ephemeral | 176 |
| ephemeral_dir | 176 |
| extra_json | 176 |
| fsx_settings | 177 |
| iam_lambda_role | 177 |
| initial_queue_size | 178 |
| key_name | 179 |
| maintain_initial_size | 179 |
| master_instance_type | 180 |
| master_root_volume_size | 180 |
| max_queue_size | 181 |
| max_vcpus | 181 |
| min_vcpus | 182 |
| placement | 182 |
| placement_group | 183 |
| post_install | 184 |
| post_install_args | 184 |
| pre_install | 184 |
| pre_install_args | 185 |
| proxy_server | 185 |
| queue_settings | 185 |
| raid_settings | 186 |
| s3_read_resource | 187 |
| s3_read_write_resource | 187 |
| scaling_settings | 187 |
| scheduler | 188 |
| shared_dir | 189 |
| spot_bid_percentage | 189 |
| spot_price | 190 |
| tags | 190 |

| | |
|----------------------------------|-----|
| template_url | 191 |
| vpc_settings | 192 |
| Sección [compute_resource] | 192 |
| initial_count | 193 |
| instance_type | 193 |
| max_count | 194 |
| min_count | 194 |
| spot_price | 195 |
| Sección [cw_log] | 195 |
| enable | 195 |
| retention_days | 196 |
| Sección [dashboard] | 196 |
| enable | 196 |
| Sección de [dcv] | 197 |
| access_from | 198 |
| enable | 198 |
| port | 199 |
| Sección de [ebs] | 199 |
| shared_dir | 200 |
| ebs_kms_key_id | 200 |
| ebs_snapshot_id | 201 |
| ebs_volume_id | 201 |
| encrypted | 201 |
| volume_iops | 202 |
| volume_size | 203 |
| volume_throughput | 203 |
| volume_type | 204 |
| Sección [efs] | 205 |
| efs_fs_id | 206 |
| efs_kms_key_id | 207 |
| encrypted | 207 |
| performance_mode | 208 |
| provisioned_throughput | 208 |
| shared_dir | 209 |
| throughput_mode | 209 |
| Sección [fsx] | 210 |

| | |
|---|-----|
| auto_import_policy | 212 |
| automatic_backup_retention_days | 213 |
| copy_tags_to_backups | 213 |
| daily_automatic_backup_start_time | 214 |
| data_compression_type | 215 |
| deployment_type | 215 |
| drive_cache_type | 216 |
| export_path | 217 |
| fsx_backup_id | 217 |
| fsx_fs_id | 218 |
| fsx_kms_key_id | 218 |
| import_path | 219 |
| imported_file_chunk_size | 219 |
| per_unit_storage_throughput | 220 |
| shared_dir | 220 |
| storage_capacity | 221 |
| storage_type | 222 |
| weekly_maintenance_start_time | 223 |
| Sección de [queue] | 224 |
| compute_resource_settings | 225 |
| compute_type | 225 |
| disable_hyperthreading | 226 |
| enable_efa | 226 |
| enable_efa_gdr | 226 |
| placement_group | 227 |
| Sección de [raid] | 228 |
| shared_dir | 229 |
| ebs_kms_key_id | 229 |
| encrypted | 230 |
| num_of_raid_volumes | 230 |
| raid_type | 230 |
| volume_iops | 231 |
| volume_size | 232 |
| volume_throughput | 233 |
| volume_type | 233 |
| Sección [scaling] | 234 |

| | |
|--|-----|
| scaledown_idletime | 234 |
| Sección de [vpc] | 235 |
| additional_sg | 235 |
| compute_subnet_cidr | 236 |
| compute_subnet_id | 236 |
| master_subnet_id | 236 |
| ssh_from | 236 |
| use_public_ips | 237 |
| vpc_id | 237 |
| vpc_security_group_id | 237 |
| Ejemplos | 40 |
| SlurmEjemplo de | 239 |
| SGE y un ejemplo Torque | 240 |
| AWS BatchEjemplo de | 241 |
| Cómo funciona AWS ParallelCluster | 243 |
| Procesos de AWS ParallelCluster | 243 |
| SGE and Torque integration processes | 244 |
| Slurm integration processes | 250 |
| AWS servicios utilizados por AWS ParallelCluster | 250 |
| AWS Auto Scaling | 251 |
| AWS Batch | 252 |
| AWS CloudFormation | 252 |
| Amazon CloudWatch | 252 |
| Amazon CloudWatch Logs | 253 |
| AWS CodeBuild | 253 |
| Amazon DynamoDB | 253 |
| Amazon Elastic Block Store | 254 |
| Amazon Elastic Compute Cloud | 254 |
| Amazon Elastic Container Registry | 254 |
| Amazon EFS | 254 |
| Amazon FSx para Lustre | 255 |
| AWS Identity and Access Management | 255 |
| AWS Lambda | 255 |
| Amazon DCV | 256 |
| Amazon Route 53 | 256 |
| Amazon Simple Notification Service | 256 |

| | |
|--|-----|
| Amazon Simple Queue Service | 257 |
| Amazon Simple Storage Service | 257 |
| Amazon VPC | 257 |
| Auto Scaling de AWS ParallelCluster | 258 |
| Ampliación | 259 |
| Reducción | 260 |
| Clúster estático | 260 |
| Tutoriales | 261 |
| Ejecución del primer trabajo en AWS ParallelCluster | 261 |
| Comprobación de la instalación | 261 |
| Creación de su primer clúster | 262 |
| Iniciar sesión en su nodo principal | 262 |
| Ejecución de su primer trabajo con SGE | 263 |
| Creación de una AMI de AWS ParallelCluster personalizada | 264 |
| Cómo personalizar la AMI de AWS ParallelCluster | 265 |
| Modificar una AMI de | 266 |
| Creación de una AMI de AWS ParallelCluster personalizada | 268 |
| Uso de una AMI personalizada en tiempo de ejecución | 269 |
| Ejecución de un trabajo de MPI con AWS ParallelCluster y el programador awsbatch | 270 |
| Creación del clúster | 270 |
| Iniciar sesión en su nodo principal | 262 |
| Ejecución de su primer trabajo con AWS Batch | 272 |
| Ejecución de un trabajo de MPI en un entorno en paralelo de varios nodos | 274 |
| Cifrado de disco con una clave KMS personalizada | 278 |
| Creación de la función | 279 |
| Asigne permisos a la clave | 279 |
| Creación del clúster | 270 |
| Tutorial sobre el modo de cola múltiple | 280 |
| Ejecutar sus trabajos en AWS ParallelCluster con el modo de cola múltiple | 280 |
| Desarrollo | 293 |
| Configuración de un libro de recetas de AWS ParallelCluster personalizado | 293 |
| Pasos | 294 |
| Configuración de un paquete de nodos de AWS ParallelCluster personalizado | 295 |
| Pasos | 294 |
| Resolución de problemas | 297 |
| Recuperación y conservación de registros | 297 |

| | |
|--|-----|
| Solución de problemas de implementación de pilas | 298 |
| Solución de problemas en varios clústeres en modo de cola | 299 |
| Registros clave | 299 |
| Solución de problemas de inicialización de nodos | 300 |
| Solución de problemas de sustituciones y terminaciones inesperadas de nodos | 302 |
| Reemplazar, terminar o apagar las instancias y nodos problemáticos | 303 |
| Solución de otros problemas de nodos y trabajos conocidos | 304 |
| Solución de problemas en clústeres en modo de cola única | 304 |
| Registros clave | 305 |
| Solución de problemas en las operaciones de inicio y unión fallidas | 306 |
| Solución de problemas de escalar | 307 |
| Solución de otros problemas relacionados con el clúster | 307 |
| Problemas con los grupos de ubicación y el lanzamiento de instancias | 307 |
| Directorios que no se pueden reemplazar | 308 |
| Solución de problemas en Amazon DCV | 309 |
| Logs para Amazon DCV | 309 |
| Memoria de tipo DCV instancia de Amazon | 309 |
| DCVProblemas con Ubuntu Amazon | 309 |
| Solución de problemas en clústeres con integración de AWS Batch | 310 |
| Problemas con el nodo principal | 310 |
| AWS Batch problemas de envío de trabajos paralelos de varios nodos | 310 |
| Problemas informáticos | 311 |
| Errores en los trabajos | 311 |
| Solución de problemas cuando un recurso no se crea | 311 |
| Solución de problemas IAM de tamaño de la política | 312 |
| Compatibilidad adicional | 313 |
| Política de compatibilidad de AWS ParallelCluster | 314 |
| Seguridad | 315 |
| Información de seguridad de los servicios utilizados por AWS ParallelCluster | 316 |
| Protección de datos | 316 |
| Cifrado de datos | 317 |
| Véase también | 319 |
| Identity and Access Management | 319 |
| Validación de conformidad | 320 |
| Aplicación de TLS 1.2 | 321 |
| Determinar los protocolos admitidos actualmente | 321 |

| | |
|---|---------|
| Compilar OpenSSL y Python | 323 |
| Notas de la versión e historial de revisión | 325 |
| | ccclxxi |

¿Qué es AWS ParallelCluster?

AWS ParallelCluster es una herramienta de administración de clústeres de código abierto compatible con AWS que le ayuda a implementar y administrar clústeres de computación de alto rendimiento (HPC) en la nube de Nube de AWS. Configura automáticamente los recursos informáticos, programador y el sistema de archivos compartido. Puede usar AWS ParallelCluster con planificadores AWS Batch y Slurm.

Con AWS ParallelCluster, puede crear e implementar rápidamente entornos de cómputo HPC de prueba de concepto y de producción. También puede crear e implementar flujos de trabajo de nivel superior sobre AWS ParallelCluster como, por ejemplo, un portal de genómica que automatice el flujo de trabajo completo de la secuenciación del ADN.

Precios

Al utilizar la interfaz de línea de comandos (CLI) o API de AWS ParallelCluster, solo paga por los recursos de AWS que se crean al crear o actualizar imágenes y clústeres de AWS ParallelCluster. Para obtener más información, consulte [AWS servicios utilizados por AWS ParallelCluster](#).

Configuración AWS ParallelCluster

Temas

- [Instalación AWS ParallelCluster](#)
- [Configuración AWS ParallelCluster](#)
- [Prácticas recomendadas](#)
- [¿Pasar de a CfnCluster AWS ParallelCluster](#)
- [Regiones admitidas](#)

Instalación AWS ParallelCluster

AWS ParallelCluster se distribuye como un paquete de Python y se instala mediante `pip` el administrador de paquetes de Python. Para obtener más información acerca de la instalación de paquetes de Python, consulte el tema sobre la [instalación de paquetes](#) en la Guía del usuario de paquetes de Python.

Maneras de instalar AWS ParallelCluster:

- [Mediante un entorno virtual \(recomendado\)](#)
- [Uso de `pip`](#)

Puedes encontrar el número de versión más reciente CLI en la [página de versiones de GitHub](#).

En esta guía, los comandos de ejemplo dan por hecho que tiene Python v3 instalado. Los comandos de ejemplo `pip` utilizan la versión `pip3`.

Instalación AWS ParallelCluster en un entorno virtual (recomendado)

Le recomendamos que instale AWS ParallelCluster en un entorno virtual. Si tiene problemas al intentar realizar la instalación AWS ParallelCluster con `pip3`, puede [instalar AWS ParallelCluster en un entorno virtual](#) para aislar la herramienta y sus dependencias. También puede utilizar una versión de Python diferente de la que usa habitualmente.

Instalación AWS ParallelCluster en un entorno no virtual utilizando pip

El principal método de distribución de AWS ParallelCluster en Linux, Windows y macOS es `pip`, que es un administrador de paquetes para Python. Proporciona una manera de instalar, actualizar y eliminar paquetes de Python y sus dependencias.

Actuales AWS ParallelCluster Versión

AWS ParallelCluster se actualiza periódicamente. Para determinar si dispone de la versión más reciente, consulte la [página de versiones en GitHub](#).

Si ya tiene `pip` una versión compatible de Python, puede instalar AWS ParallelCluster mediante el siguiente comando. Si tiene instalada la versión 3 o superior de Python, recomendamos que utilice el comando `pip3`.

```
$ pip3 install "aws-parallelcluster<3.0" --upgrade --user
```

Pasos que seguir después de la instalación

Después de instalar AWS ParallelCluster, puede que tenga que añadir la ruta del archivo ejecutable a la PATH variable. Si desea obtener instrucciones específicas de las distintas plataformas, consulte estos temas:

- Linux: [Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos](#)
- macOS: [Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos](#)
- Windows: [Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos](#)

Puedes verificarlo AWS ParallelCluster se instaló correctamente ejecutándolo `pcluster version`.

```
$ pcluster version
2.11.9
```

AWS ParallelCluster se actualiza periódicamente. Para actualizar a la versión más reciente de AWS ParallelCluster, vuelva a ejecutar el comando de instalación. Para obtener más información sobre la versión más reciente de AWS ParallelCluster, consulte la [AWS ParallelCluster notas de publicación](#).

```
$ pip3 install "aws-parallelcluster<3.0" --upgrade --user
```

Para desinstalar AWS ParallelCluster, utilice `pip uninstall`.

```
$ pip3 uninstall "aws-parallelcluster<3.0"
```

Si no dispone de Python ni pip, utilice el procedimiento para su entorno.

Instrucciones detalladas para cada entorno

- [Instalación AWS ParallelCluster en un entorno virtual \(recomendado\)](#)
- [Instalación AWS ParallelCluster en Linux](#)
- [Instalación AWS ParallelCluster en macOS](#)
- [Instalación AWS ParallelCluster en Windows](#)

Instalación AWS ParallelCluster en un entorno virtual (recomendado)

Le recomendamos que instale AWS ParallelCluster en un entorno virtual para evitar conflictos entre las versiones obligatorias y otros pip paquetes.

Requisitos previos

- Compruebe que pip y Python están instalados. Recomendamos pip3 y Python 3 versión 3.6. Si utiliza Python 2, utilice pip en lugar de pip3 y virtualenv en lugar de venv.

Para instalar AWS ParallelCluster en un entorno virtual

1. Si virtualenv no está instalado, instale virtualenv mediante pip3. Si `python3 -m virtualenv help` muestra información de ayuda, vaya al paso 2.

Linux, macOS, or Unix

```
$ python3 -m pip install --upgrade pip
$ python3 -m pip install --user --upgrade virtualenv
```

Ejecute `exit` para salir de la ventana de terminal actual y abrir una nueva para detectar los cambios del entorno.

Windows

```
C:\>pip3 install --user --upgrade virtualenv
```

Ejecute `exit` para salir del símbolo del sistema actual y abrir uno nuevo para detectar los cambios del entorno.

2. Cree un entorno virtual y asígnele un nombre.

Linux, macOS, or Unix

```
$ python3 -m virtualenv ~/apc-ve
```

También puede usar la opción `-p` para especificar una versión específica de Python.

```
$ python3 -m virtualenv -p $(which python3) ~/apc-ve
```

Windows

```
C:\>virtualenv %USERPROFILE%\apc-ve
```

3. Active el entorno virtual nuevo.

Linux, macOS, or Unix

```
$ source ~/apc-ve/bin/activate
```

Windows

```
C:\>%USERPROFILE%\apc-ve\Scripts\activate
```

4. Instalación AWS ParallelCluster en su entorno virtual.

Linux, macOS, or Unix

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster<3.0"
```

Windows

```
(apc-ve) C:\>pip3 install --upgrade "aws-parallelcluster<3.0"
```

5. Compruebe que AWS ParallelCluster está instalado correctamente.

Linux, macOS, or Unix

```
$ pcluster version  
2.11.9
```

Windows

```
(apc-ve) C:\>pcluster version  
2.11.9
```

Puede utilizar el comando `deactivate` para salir del entorno virtual. Cada vez que inicie una sesión, debe [activar el entorno de nuevo](#).

Para actualizar a la versión más reciente de AWS ParallelCluster, vuelva a ejecutar el comando de instalación.

Linux, macOS, or Unix

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster<3.0"
```

Windows

```
(apc-ve) C:\>pip3 install --upgrade "aws-parallelcluster<3.0"
```

Instalación AWS ParallelCluster en Linux

Puedes instalar AWS ParallelCluster y sus dependencias en la mayoría de las distribuciones de Linux mediante el uso `pip` de un administrador de paquetes para Python. En primer lugar, determine si Python y `pip` están instalados:

1. Para determinar si su versión de Linux incluye Python y `pip`, ejecute `pip --version`.

```
$ pip --version
```

Si lo ha `pip` instalado, vaya a la sección de instalación [AWS ParallelCluster con el tema pip](#). De lo contrario, prosiga con el paso 2.

2. Para determinar si Python está instalado, ejecute `python --version`.

```
$ python --version
```

Si tiene instalada la versión 3.6+ de Python 3 o la versión 2.7 de Python 2, vaya a la sección de instalación [AWS ParallelCluster con el tema pip](#). De lo contrario, [instale Python](#) y, a continuación, vuelva a este procedimiento para instalar pip.

3. Instale pip con el script proporcionado por Python Packaging Authority.
4. Utilice el comando `curl` para descargar el script de instalación.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
```

5. Ejecute el script con Python para descargar e instalar la versión más reciente de pip y otros paquetes de soporte necesarios.

```
$ python get-pip.py --user
```

o

```
$ python3 get-pip.py --user
```

Cuando se incluye el modificador `--user`, el script instala pip en la ruta `~/.local/bin`.

6. Para asegurarse de que la carpeta que contiene pip forma parte de la variable PATH, haga lo siguiente:
 - a. Busque el script de perfil de su shell en su carpeta de usuario. Si no está seguro de cuál es el shell que tiene, ejecute `basename $SHELL`.

```
$ ls -a ~  
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash: `.bash_profile`, `.profile` o `.bash_login`.
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` o `.login`.

- b. Añada un comando de exportación al final del script de su perfil igual que en el siguiente ejemplo.

```
export PATH=~/.local/bin:$PATH
```

El comando de exportación inserta la ruta, que es `~/.local/bin` en este ejemplo, delante de la variable `PATH` existente.

c. Para que estos cambios surtan efecto, vuelva a cargar el perfil en la sesión actual.

```
$ source ~/.bash_profile
```

7. Compruebe que `pip` está instalado correctamente.

```
$ pip3 --version
pip 21.3.1 from ~/.local/lib/python3.6/site-packages (python 3.6)
```

Secciones

- [Instalación AWS ParallelCluster con pip](#)
- [Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos](#)
- [Instalación de Python en Linux](#)

Instalación AWS ParallelCluster con **pip**

pipÚselo para instalar AWS ParallelCluster.

```
$ python3 -m pip install "aws-parallelcluster<3.0" --upgrade --user
```

Cuando se utiliza el `--user` conmutador, se `pip` instala AWS ParallelCluster para `~/.local/bin`.

Verifica que AWS ParallelCluster instalado correctamente.

```
$ pcluster version
2.11.9
```

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
$ python3 -m pip install "aws-parallelcluster<3.0" --upgrade --user
```

Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos

Después de realizar la instalación con `pip`, es posible que necesite añadir el archivo ejecutable `pcluster` a la variable de entorno `PATH` del sistema operativo.

Para comprobar la carpeta en la que se `pip` instaló AWS ParallelCluster, ejecute el siguiente comando.

```
$ which pcluster
/home/username/.local/bin/pcluster
```

Si omitió el `--user` conmutador al instalarlo AWS ParallelCluster, el ejecutable puede estar en la `bin` carpeta de la instalación de Python. Si no sabe dónde se ha instalado Python, ejecute este comando.

```
$ which python
/usr/local/bin/python
```

Tenga en cuenta que la salida puede ser la ruta a un symlink, no al archivo ejecutable real. Para ver adónde apunta el symlink, ejecute `ls -al`.

```
$ ls -al $(which python)
/usr/local/bin/python -> ~/.local/Python/3.6/bin/python3.6
```

Si esta es la misma carpeta que ha añadido a la ruta en el paso 3 en [Instalación AWS ParallelCluster](#), ya ha terminado con la instalación. De lo contrario, siga de nuevo esos mismos pasos del 3a al 3c, añadiendo esta carpeta adicional a la ruta.

Instalación de Python en Linux

Si su distribución no venía con Python o venía con una versión anterior, instale Python antes de instalar `pip` y AWS ParallelCluster.

Para instalar Python 3 en Linux

1. Compruebe si Python ya está instalado.

```
$ python3 --version
```

O

```
$ python --version
```

Note

Si su distribución de Linux incluye Python, es posible que tenga que instalar el paquete de desarrollador de Python. El paquete para desarrolladores incluye los encabezados y las bibliotecas necesarios para compilar las extensiones e instalarlas AWS ParallelCluster. Usa tu administrador de paquetes para instalar el paquete de desarrollador. Suele llamarse `python-dev` o `python-devel`.

2. Si no está instalado Python 2.7 o una versión posterior, instale Python con el administrador de paquetes de su distribución. El comando y el nombre del paquete varían:

- En derivados de Debian, como por ejemplo Ubuntu, use `apt`.

```
$ sudo apt-get install python3
```

- En Red Hat y sus derivados, use `yum`.

```
$ sudo yum install python3
```

- SUSE Activado y derivados, utilice `zypper`.

```
$ sudo zypper install python3
```

3. Para verificar que Python está instalado correctamente, abra un símbolo del sistema o shell y ejecute el siguiente comando.

```
$ python3 --version
Python 3.8.11
```

Instalación AWS ParallelCluster en macOS

Secciones

- [Requisitos previos](#)

- [Instalación AWS ParallelCluster en macOS con pip](#)
- [Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos](#)

Requisitos previos

- Python 3 versión 3.7+ o Python 2 versión 2.7

Compruebe su instalación de Python.

```
$ python --version
```

Si aún no se ha instalado Python en el equipo o quiere instalar otra versión distinta, siga el procedimiento detallado en [Instalación AWS ParallelCluster en Linux](#).

Instalación AWS ParallelCluster en macOS con pip

También se puede utilizar pip directamente para instalar AWS ParallelCluster. Si no lo tiene pip, siga las instrucciones del [tema principal de instalación](#). Ejecute `pip3 --version` para saber si su versión de macOS ya incluye Python y pip3.

```
$ pip3 --version
```

Para instalar AWS ParallelCluster en macOS

1. Descargue e instale la última versión de Python de la [página de descargas](#) de [Python.org](#).
2. Descargue y ejecute el script de instalación de pip3 proporcionado por Python Packaging Authority.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py  
$ python3 get-pip.py --user
```

3. Utilice el recién instalado pip3 para instalar AWS ParallelCluster. Se recomienda que si utiliza Python versión 3+, utilice el pip3 comando.

```
$ python3 -m pip install "aws-parallelcluster<3.0" --upgrade --user
```

4. Verifica que AWS ParallelCluster está instalado correctamente.

```
$ pcluster version
2.11.9
```

Si no se encuentra el programa, [añádalo a la ruta de la línea de comandos](#).

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
$ pip3 install "aws-parallelcluster<3.0" --upgrade --user
```

Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos

Después de realizar la instalación con `pip`, es posible que necesite añadir el programa `pcluster` a la variable de entorno `PATH` del sistema operativo. La ubicación del programa depende de dónde se haya instalado Python.

Example AWS ParallelCluster ubicación de instalación: macOS con Python 3.6 y `pip` (modo usuario)

```
~/Library/Python/3.6/bin
```

Sustituya la versión de Python que tiene con la versión del ejemplo anterior.

Si no sabe dónde se ha instalado Python, ejecute `which python`.

```
$ which python3
/usr/local/bin/python3
```

La salida puede ser la ruta a un symlink, no la ruta al programa real. Ejecute `ls -al` para ver adónde apunta.

```
$ ls -al /usr/local/bin/python3
lrwxr-xr-x 1 username admin 36 Mar 12 12:47 /usr/local/bin/python3 -> ../Cellar/
python/3.6.8/bin/python3
```

`pip` instala los programas en la misma carpeta que contiene la aplicación de Python. Añada esta carpeta a la variable `PATH`.

Para modificar la variable **PATH** (Linux, Unix o macOS):

1. Busque el script de perfil de su shell en su carpeta de usuario. Si no está seguro de cuál es el shell que tiene, ejecute `echo $SHELL`.

```
$ ls -a ~  
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash: `.bash_profile`, `.profile` o `.bash_login`.
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` o `.login`.

2. Añada un comando de exportación al script de su perfil.

```
export PATH=~/.local/bin:$PATH
```

Este comando añade una ruta, `~/.local/bin` en este ejemplo, a la variable `PATH` actual.

3. Cargue el perfil en su sesión actual.

```
$ source ~/.bash_profile
```

Instalación AWS ParallelCluster en Windows

Puede instalar AWS ParallelCluster en Windows mediante `pip`, que es un administrador de paquetes para Python. Si ya tiene `pip`, siga las instrucciones del [tema de instalación](#) principal.

Secciones

- [Instalación AWS ParallelCluster usando Python y pip en Windows](#)
- [Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos](#)

Instalación AWS ParallelCluster usando Python y **pip** en Windows

La Python Software Foundation ofrece instaladores para Windows que incluyen `pip`.

Para instalar Python 3.6 y **pip** (Windows)

1. Descargue el instalador de Python para Windows x86-64 desde la [página de descargas](#) de [Python.org](#).
2. Ejecute el instalador.
3. Elija Añadir Python 3 a PATH.
4. Seleccione Install Now (Instalar ahora).

El instalador instala Python en la carpeta del usuario y añade sus carpetas de programas a la ruta del usuario.

Para instalar AWS ParallelCluster con **pip3** (Windows)

Si usa la versión 3 o superior de Python, recomendamos que utilice el comando `pip3`.

1. Abra el Command Prompt (Símbolo del sistema) desde el menú Start (Inicio).
2. Utilice los siguientes comandos para comprobar que tanto Python como `pip` estén instalados correctamente.

```
C:\>py --version
Python 3.8.11
C:\>pip3 --version
pip 21.3.1 from c:\python38\lib\site-packages\pip (python 3.8)
```

3. Instalación AWS ParallelCluster utilizando `pip`.

```
C:\>pip3 install "aws-parallelcluster<3.0"
```

4. Compruebe que AWS ParallelCluster está instalado correctamente.

```
C:\>pcluster version
2.11.9
```

Para actualizar a la versión más reciente, ejecute el comando de instalación de nuevo.

```
C:\>pip3 install --user --upgrade "aws-parallelcluster<3.0"
```

Añada el AWS ParallelCluster ejecutable a la ruta de la línea de comandos

Después de la instalación AWS ParallelCluster con `pip`, añade el `pcluster` programa a la variable de `PATH` entorno de su sistema operativo.

Para encontrar dónde está instalado el programa de `pcluster`, ejecute el siguiente comando.

```
C:\>where pcluster
C:\Python38\Scripts\pcluster.exe
```

Si ese comando no devuelve ningún resultado, debe añadir la ruta manualmente. Utilice la línea de comandos o el Explorador de Windows para averiguar dónde está instalado en el equipo. Las rutas típicas incluyen:

- Python 3 y **pip3**: `C:\Python38\Scripts\`
- Python 3 y opción **pip3** `--user -- %APPDATA%\Python\Python38\Scripts`

Note

Los nombres de carpeta que incluyen los números de versión puede variar. En los ejemplos anteriores se muestra Python38. Sustitúyalos según sea necesario por el número de versión que está utilizando.

Para modificar la `PATH` variable (Windows)

1. Pulse la tecla de Windows y escriba **environment variables**.
2. Elija Edit environment variables for your account (Editar las variables de entorno de esta cuenta).
3. Elija y `PATH`, a continuación, elija Editar.
4. Añada la ruta al campo Variable value (Valor de variable). Por ejemplo: **`C:\new\path`**.
5. Elija OK (Aceptar) dos veces para aplicar la nueva configuración.
6. Cierre los símbolos del sistema en ejecución y vuelva a abrir la ventana de símbolo del sistema.

Configuración AWS ParallelCluster

Después de la instalación AWS ParallelCluster, complete los siguientes pasos de configuración.

Compruebe que su AWS La cuenta tiene una función que incluye los permisos necesarios para ejecutar la `pcluster` CLI. Para obtener más información, consulte [AWS ParallelCluster ejemplos de políticas de instancia y usuario](#).

Configura tu AWS credenciales. Para obtener más información, consulte [Configuración del AWS CLI](#) en la AWS CLI guía del usuario.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default Región de AWS name [us-east-1]: us-east-1
Default output format [None]:
```

La Región de AWS el lugar donde se lance el clúster debe tener al menos un par de EC2 claves de Amazon. Para obtener más información, consulta los [pares de EC2 claves de Amazon](#) en la Guía del EC2 usuario de Amazon.

```
$ pcluster configure
```

El asistente de configuración le pedirá toda la información necesaria para crear su clúster. Los detalles de la secuencia difieren cuando se utiliza AWS Batch como planificador en comparación con el uso Slurm. Para obtener más información sobre la configuración de un clúster, consulte [Configuración](#).

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores. Puedes seguir utilizándolos en las versiones anteriores a la 2.11.4 (inclusive), pero no son aptas para futuras actualizaciones o soporte de solución de problemas por parte del AWS servicio y AWS Equipos de soporte.

Slurm

De la lista de válidos Región de AWS identificadores, elija el Región de AWS dónde desea que se ejecute el clúster.

Note

La lista de Regiones de AWS La que se muestra se basa en la partición de su cuenta y solo incluye Regiones de AWS que estén habilitados para su cuenta. Para obtener más información sobre la activación Regiones de AWS para tu cuenta, consulta [Administrar Regiones de AWS](#) en la Referencia general de AWS. El ejemplo que se muestra proviene del AWS Partición global. Si su cuenta está en el AWS GovCloud (US) partición, únicamente Regiones de AWS en esa partición aparecen listados (gov-us-east-1gov-us-west-1). Del mismo modo, si su cuenta está en AWS Solo se muestran cn-north-1 las particiones cn-northwest-1 de China. Para ver la lista completa de Regiones de AWS con el apoyo de AWS ParallelCluster, consulte [Regiones admitidas](#).

Allowed values for the Región de AWS ID:

1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ca-central-1
9. eu-central-1
10. eu-north-1
11. eu-south-1
12. eu-west-1
13. eu-west-2
14. eu-west-3
15. me-south-1
16. sa-east-1
17. us-east-1
18. us-east-2
19. us-west-1
20. us-west-2

Región de AWS ID [ap-northeast-1]:

Elija el programador que desea utilizar con el clúster.

Allowed values for Scheduler:

```
1. slurm
2. awsbatch
Scheduler [slurm]:
```

Elija el sistema operativo.

```
Allowed values for Operating System:
1. alinux2
2. centos7
3. ubuntu1804
4. ubuntu2004
Operating System [alinux2]:
```

 Note

Support for `alinux2` se agregó en AWS ParallelCluster versión 2.6.0.

Se especifica el tamaño mínimo y máximo del clúster de nodos de computación. Esto se mide en número de instancias.

```
Minimum cluster size (instances) [0]:
Maximum cluster size (instances) [10]:
```

Se especifican los tipos de instancia de los nodos principal y de computación. Para los tipos de instancias, los límites de instancias de su cuenta son lo suficientemente grandes como para satisfacer sus requisitos. Para obtener más información, consulta [los límites de las instancias bajo demanda](#) en la Guía del EC2 usuario de Amazon.

```
Master instance type [t2.micro]:
Compute instance type [t2.micro]:
```

El par de claves se selecciona entre los pares de claves registrados EC2 en Amazon en el Región de AWS.

```
Allowed values for EC2 Key Pair Name:
1. prod-uswest1-key
2. test-uswest1-key
```

```
EC2 Key Pair Name [prod-uswest1-key]:
```

Una vez completados los pasos anteriores, decida si desea utilizar uno existente VPC o alquilarlo AWS ParallelCluster crea una VPC para ti. Si no tiene una configuración adecuada VPC, AWS ParallelCluster puede crear uno nuevo. Usa tanto el nodo principal como los de computación de la misma subred pública, o solo el nodo principal en una subred pública con todos los nodos en una subred privada. Es posible alcanzar tu límite de cantidad VPCs en un Región de AWS. El límite predeterminado es de cinco VPCs para cada Región de AWS. Para obtener más información sobre este límite y sobre cómo solicitar un aumento, consulta [VPCs y subredes](#) en la Guía del VPC usuario de Amazon.

Si dejas AWS ParallelCluster cree una VPC, debe decidir si todos los nodos deben estar en una subred pública.

Important

VPCs creado por AWS ParallelCluster no habilite los registros VPC de flujo de forma predeterminada. VPC Los registros de flujo le permiten capturar información sobre el tráfico IP que entra y sale de las interfaces de red de su VPCs. Para obtener más información, consulta [VPC Flow Logs](#) en la Guía del VPC usuario de Amazon.

Note

Si lo desea 1. Master in a public subnet and compute fleet in a private subnet, AWS ParallelCluster crea una NAT puerta de enlace que se traduce en un coste adicional, incluso si se especifican los recursos de la capa gratuita.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Network Configuration:
1. Master in a public subnet and compute fleet in a private subnet
2. Master and compute fleet in the same public subnet
Network Configuration [Master in a public subnet and compute fleet in a private
subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Si no crea una nueva VPC, debe seleccionar una existente VPC.

Si eliges tener AWS ParallelCluster cree elVPC, anote el VPC ID para que pueda usar el AWS CLI para eliminarlo más tarde.

```
Automate VPC creation? (y/n) [n]: n
Allowed values for VPC ID:
#  id                                     name                                     number_of_subnets
---  -----
 1  vpc-0b4ad9c4678d3c7ad  ParallelClusterVPC-20200118031893      2
 2  vpc-0e87c753286f37eef  ParallelClusterVPC-20191118233938      5
VPC ID [vpc-0b4ad9c4678d3c7ad]: 1
```

Una vez seleccionada, debe decidir si desea utilizar las subredes existentes o crear subredes nuevas. VPC

```
Automate Subnet creation? (y/n) [y]: y
```

```
Creating CloudFormation stack...
Do not leave the terminal until the process has finished
```

AWS Batch

De la lista de válidas Región de AWS identificadores, elija el Región de AWS dónde desea que se ejecute el clúster.

```
Allowed values for Región de AWS ID:
1. ap-northeast-1
2. ap-northeast-2
3. ap-south-1
4. ap-southeast-1
5. ap-southeast-2
6. ca-central-1
7. eu-central-1
8. eu-north-1
9. eu-west-1
10. eu-west-2
11. eu-west-3
12. sa-east-1
13. us-east-1
14. us-east-2
15. us-west-1
16. us-west-2
```

```
Región de AWS ID [ap-northeast-1]:
```

Elija el programador que desea utilizar con el clúster.

```
Allowed values for Scheduler:
```

1. slurm
2. awsbatch

```
Scheduler [awsbatch]:
```

Cuando `awsbatch` se selecciona como programador, `alinux2` se utiliza como sistema operativo.

Se especifica el tamaño mínimo y máximo del clúster de nodos de computación. Esto se mide en vCPUs.

```
Minimum cluster size (vcpus) [0]:
```

```
Maximum cluster size (vcpus) [10]:
```

Se especifica el tipo de instancia del nodo principal. Cuando se utiliza el programador `awsbatch`, los nodos de computación utilizan el tipo de instancia `optimal`.

```
Master instance type [t2.micro]:
```

El par de EC2 claves de Amazon se selecciona entre los pares de claves registrados EC2 en Amazon en el Región de AWS.

```
Allowed values for EC2 Key Pair Name:
```

1. prod-uswest1-key
2. test-uswest1-key

```
EC2 Key Pair Name [prod-uswest1-key]:
```

Decida si desea utilizar los existentes VPCs o alquilarlos AWS ParallelCluster crear VPCs para ti. Si no tiene una configuración adecuada VPC, AWS ParallelCluster puede crear uno nuevo. Usa tanto el nodo principal como los de computación de la misma subred pública, o solo el nodo principal en una subred pública con todos los nodos en una subred privada. Es posible alcanzar tu límite de cantidad VPCs en un Región de AWS. El número predeterminado VPCs es cinco. Para obtener más información sobre este límite y sobre cómo solicitar un aumento, consulta [VPCs y subredes](#) en la Guía del VPC usuario de Amazon.

⚠ Important

VPCs creado por AWS ParallelCluster no habilite los registros VPC de flujo de forma predeterminada. VPC Los registros de flujo le permiten capturar información sobre el tráfico IP que entra y sale de las interfaces de red de su VPCs. Para obtener más información, consulta [VPCFlow Logs](#) en la Guía del VPC usuario de Amazon.

Si dejas AWS ParallelCluster cree una VPC, decida si todos los nodos deben estar en una subred pública.

ℹ Note

Si lo desea, 1. Master in a public subnet and compute fleet in a private subnet AWS ParallelCluster crea una NAT puerta de enlace que se traduce en un coste adicional, incluso si se especifican los recursos de la capa gratuita.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Network Configuration:
1. Master in a public subnet and compute fleet in a private subnet
2. Master and compute fleet in the same public subnet
Network Configuration [Master in a public subnet and compute fleet in a private
subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Si no crea una nueva VPC, debe seleccionar una existente VPC.

Si eliges tener AWS ParallelCluster cree el VPC, anote el VPC ID para que pueda usar el AWS CLI para eliminarlo más tarde.

```
Automate VPC creation? (y/n) [n]: n
Allowed values for VPC ID:
#  id                                     name                                     number_of_subnets
---  -----
1  vpc-0b4ad9c4678d3c7ad  ParallelClusterVPC-20200118031893      2
2  vpc-0e87c753286f37eef  ParallelClusterVPC-20191118233938      5
VPC ID [vpc-0b4ad9c4678d3c7ad]: 1
```

Una VPC vez seleccionada, decida si desea utilizar las subredes existentes o crear nuevas.

```
Automate Subnet creation? (y/n) [y]: y
```

```
Creating CloudFormation stack...  
Do not leave the terminal until the process has finished
```

Cuando haya completado los pasos anteriores, un clúster simple se lanzará en unVPC. VPCUtiliza una subred existente que admite direcciones IP públicas. La tabla de enrutamiento de la subred es `0.0.0.0/0 => igw-xxxxxx`. Tenga en cuenta las siguientes condiciones:

- VPCImprescindible DNS Resolution = yes y. DNS Hostnames = yes
- También VPC deberían tener DHCP opciones con la correcta domain-name para Región de AWS. El conjunto de DHCP opciones por defecto ya especifica lo necesario AmazonProvidedDNS. Si especificas más de un servidor de nombres de dominio, consulta los [conjuntos de DHCP opciones](#) en la Guía del VPC usuario de Amazon. Cuando utilice subredes privadas, utilice una NAT puerta de enlace o un proxy interno para permitir el acceso web a los nodos de cómputo. Para obtener más información, consulte [Configuraciones de red](#).

Cuando todas las opciones tengan valores válidos, podrá lanzar el clúster ejecutando el comando create.

```
$ pcluster create mycluster
```

Cuando el clúster alcance el estado "CREATE_COMPLETE", podrás conectarte a él mediante la configuración normal del SSH cliente. Para obtener más información sobre la conexión a EC2 las instancias de Amazon, consulte la [Guía del EC2 usuario](#) de la Guía del EC2 usuario de Amazon.

Ejecute el siguiente comando para eliminar el clúster.

```
$ pcluster delete --region us-east-1 mycluster
```

Para eliminar los recursos de red delVPC, puedes eliminar la pila CloudFormation de redes. El nombre de la pila comienza con »parallelclusternetworking-«y contiene la hora de creación en formatoYYYYMMDDHHMMSS"». Puede enumerar las pilas mediante el comando [list-stacks](#).

```
$ aws --region us-east-1 cloudformation list-stacks \
```

```
--stack-status-filter "CREATE_COMPLETE" \  
--query "StackSummaries[].StackName" | \  
grep -e "parallelclusternetworking-" \  
"parallelclusternetworking-pubpriv-20191029205804"
```

La pila se puede eliminar mediante el comando [delete-stack](#).

```
$ aws --region us-east-1 cloudformation delete-stack \  
--stack-name parallelclusternetworking-pubpriv-20191029205804
```

Lo VPC que [pcluster configure](#) crea para usted no se crea en la pila CloudFormation de redes. Puede eliminarlo VPC manualmente en la consola o mediante el AWS CLI.

```
$ aws --region us-east-1 ec2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

Prácticas recomendadas

Prácticas recomendadas: selección del tipo de instancia principal

Aunque el nodo maestro no ejecuta ningún trabajo, sus funciones y su tamaño son cruciales para el rendimiento general del clúster.

Al elegir el tipo de instancia que se utilizará para el nodo maestro, querrá evaluar los siguientes elementos:

- **Tamaño del clúster:** el nodo maestro organiza la lógica de escalado del clúster y es responsable de adjuntar los nuevos nodos al programador. Si necesita escalar o reducir verticalmente el clúster de una cantidad considerable de nodos, entonces querrá dotar al nodo maestro de una capacidad de computación adicional.
- **Sistemas de archivos compartidos:** cuando utilice sistemas de archivos compartidos para compartir artefactos entre los nodos de procesamiento y el nodo principal, tenga en cuenta que el maestro es el nodo que expone al NFS servidor. Por este motivo, debes elegir un tipo de instancia con suficiente ancho de banda de red y suficiente ancho de EBS banda dedicado de Amazon para gestionar tus flujos de trabajo.

Prácticas recomendadas: rendimiento de la red

Hay tres consejos que cubren toda la gama de posibilidades para mejorar la comunicación en la red.

- Grupo de agrupación: un grupo con ubicación en clúster es una agrupación lógica de instancias en una misma zona de disponibilidad. Para obtener más información sobre los grupos de ubicación, consulta [los grupos de ubicación](#) en la Guía del EC2 usuario de Amazon. Puede configurar el clúster para que utilice su propio grupo de ubicación `placement_group = your-placement-group-name` o alquilarlo AWS ParallelCluster cree un grupo de ubicación con la "compute" estrategia `placement_group = DYNAMIC`. Para obtener más información, consulte [placement_group](#) el modo de cola múltiple y el modo [placement_group](#) de cola única.
- Redes mejoradas: considere la posibilidad de elegir un tipo de instancia que admita redes mejoradas. Para obtener más información, consulte [redes mejoradas en Linux](#) en la Guía del EC2 usuario de Amazon.
- Adaptador Elastic Fabric: para admitir altos niveles de comunicación escalable entre instancias, considere la posibilidad de elegir interfaces de EFA red para su red. El hardware EFA de derivación del sistema operativo (OS) personalizado mejora las comunicaciones entre instancias con la elasticidad y flexibilidad bajo demanda del AWS nube. Para configurar una sola Slurm configure `enable_efa = true` la cola de clústeres que EFA desee utilizar. Para obtener más información sobre el uso con EFA AWS ParallelCluster, consulte [Elastic Fabric Adapter](#) y [enable_efa](#). Para obtener más información EFA, consulte [Elastic Fabric Adapter](#) en la Guía del EC2 usuario de Amazon para instancias de Linux.
- Ancho de banda de la instancia: el ancho de banda se amplía con el tamaño de la instancia. Considere elegir el tipo de instancia que mejor se adapte a sus necesidades [EBS; consulte Instancias optimizadas para Amazon y tipos de EBS volumen de Amazon](#) en la Guía del EC2 usuario de Amazon.

Prácticas recomendadas: alertas de presupuesto

Para gestionar AWS ParallelCluster costos de recursos, le recomendamos que utilice AWS Budgets acciones para crear un presupuesto y alertas de umbral presupuestario definidas para los seleccionados AWS recursos. Para obtener más información, consulte [Configuración de una acción presupuestaria](#) en AWS Budgets Guía del usuario. También puedes usar Amazon CloudWatch para crear una alarma de facturación. Para obtener más información, consulta [Cómo crear una alarma de facturación para controlar tu presupuesto AWS cargos](#).

Mejores prácticas: trasladar un clúster a un nuevo AWS ParallelCluster versión secundaria o de parche

Actualmente cada AWS ParallelCluster la versión secundaria es autónoma junto con su `pcclusterCLI`. Para mover un clúster a una nueva versión secundaria o de parche, debe volver a crear el clúster con la nueva versión. CLI

Para optimizar el proceso de migración de un clúster a una nueva versión secundaria o para guardar los datos de almacenamiento compartido por otros motivos, le recomendamos que utilice las siguientes prácticas recomendadas.

- Guarde los datos personales en volúmenes externos, como Amazon EFS y FSx para Lustre. De este modo, podrá mover fácilmente los datos de un clúster a otro.
- Cree sistemas de almacenamiento compartido de los tipos que se detallan a continuación utilizando AWS CLI o AWS Management Console:
 - [Sección de \[ebs\]](#)
 - [Sección \[efs\]](#)
 - [Sección \[fsx\]](#)

Añádalos a la nueva configuración del clúster como sistemas de archivos existentes. De esta forma, se conservan al eliminar el clúster y se pueden asociar a un clúster nuevo. Los sistemas de almacenamiento compartido generalmente incurren en cargos tanto si están conectados como separados de un clúster.

Le recomendamos que utilice los sistemas de archivos Amazon o Amazon FSx for Lustre/EFS, ya que se pueden adjuntar a varios clústeres al mismo tiempo y puede adjuntarlos al nuevo clúster antes de eliminar el antiguo. Para obtener más información, consulte [Montar los sistemas de EFS archivos de Amazon](#) en la Guía del EFS usuario de Amazon y [Acceder a FSx los sistemas de archivos Lustre](#) en la Guía del usuario de Amazon FSx for Lustre Lustre.

- Utilice [acciones de arranque personalizadas](#) para personalizar sus instancias en lugar de personalizarlas. AMI Esto optimiza el proceso de creación, ya que AMI no es necesario crear una nueva personalización para cada nueva versión.
- Secuencia de Secuencia recomendada.
 1. Actualice la configuración del clúster para utilizar las definiciones de sistemas de archivos existentes.
 2. Compruebe la versión de `pccluster` y actualícela si es necesario.

3. Cree y pruebe el nuevo clúster.
 - Asegúrese de que sus datos estén disponibles en el clúster nuevo.
 - Asegúrese de que la aplicación funcione en el clúster nuevo.
4. Si su nuevo clúster está completamente probado y en funcionamiento y está seguro de que no va a utilizar el clúster anterior, elimínelo.

¿Pasar de a CfnCluster AWS ParallelCluster

AWS ParallelCluster es una versión mejorada de CfnCluster.

Si la utiliza actualmente CfnCluster, le recomendamos que utilice AWS ParallelCluster en su lugar, y cree nuevos clústeres con él. Aunque puede seguir utilizándolo CfnCluster, ya no se está desarrollando y no se añadirán nuevas características o funcionalidades.

Las principales diferencias entre CfnCluster y AWS ParallelCluster se describen en las siguientes secciones.

AWS ParallelCluster CLI administra un conjunto diferente de clústeres

Los clústeres creados con el no se `cfncluster` CLI pueden gestionarse con `pclusterCLI`. Los siguientes comandos no funcionan en los clústeres creados por CfnCluster:

```
pcluster list
pcluster update cluster_name
pcluster start cluster_name
pcluster status cluster_name
```

Para administrar los clústeres con los que ha creado CfnCluster, debe usar el `cfnclusterCLI`.

Si necesita un CfnCluster paquete para administrar sus clústeres antiguos, le recomendamos que lo instale y lo use desde un [entorno virtual de Python](#).

AWS ParallelCluster y CfnCluster utilice políticas IAM personalizadas diferentes

IAM Las políticas personalizadas que se usaron anteriormente para la creación de CfnCluster clústeres no se pueden usar con AWS ParallelCluster. Si necesita políticas personalizadas para AWS ParallelCluster, debe crear otras nuevas. Vea las AWS ParallelCluster guía.

AWS ParallelCluster y CfnCluster usa diferentes archivos de configuración

La AWS ParallelCluster el archivo de configuración reside en la `~/parallelcluster` carpeta. El archivo `CfnCluster` de configuración reside en la `~/cfncluster` carpeta.

Si desea utilizar un archivo de `CfnCluster` configuración existente con AWS ParallelCluster, a continuación, debe realizar las siguientes acciones:

1. Mueva el archivo de configuración de `~/cfncluster/config` a `~/parallelcluster/config`.
2. Si usa el parámetro de configuración [extra_json](#), cámbielo tal como se muestra.

CfnCluster configuración:

```
extra_json = { "cfncluster" : { } }
```

AWS ParallelCluster configuración:

```
extra_json = { "cluster" : { } }
```

En AWS ParallelCluster, ganglia está deshabilitada de forma predeterminada

En AWS ParallelCluster, ganglia está deshabilitada de forma predeterminada. Para activar los ganglios, complete estos pasos:

1. Establezca el parámetro [extra_json](#) tal como se muestra:

```
extra_json = { "cluster" : { "ganglia_enabled" : "yes" } }
```

2. Cambie el grupo de seguridad principal para permitir las conexiones al puerto 80.

El grupo de seguridad `parallelcluster-<CLUSTER_NAME>-MasterSecurityGroup-<xxx>` debe modificarse añadiendo una nueva regla de grupo de seguridad para permitir la conexión de entrada al puerto 80 desde su IP pública. Para obtener más información, consulta [Cómo añadir reglas a un grupo de seguridad](#) en la Guía del EC2 usuario de Amazon.

Regiones admitidas

AWS ParallelCluster la versión 2.x está disponible en las siguientes versiones Regiones de AWS:

| Nombre de la región | Región |
|--|----------------|
| Este de EE. UU. (Ohio) | us-east-2 |
| Este de EE. UU. (Norte de Virginia) | us-east-1 |
| Oeste de EE. UU. (Norte de California) | us-west-1 |
| Oeste de EE. UU. (Oregón) | us-west-2 |
| África (Ciudad del Cabo) | af-south-1 |
| Asia-Pacífico (Hong Kong) | ap-east-1 |
| Asia Pacífico (Bombay) | ap-south-1 |
| Asia-Pacífico (Seúl) | ap-northeast-2 |
| Asia-Pacífico (Singapur) | ap-southeast-1 |
| Asia-Pacífico (Sídney) | ap-southeast-2 |
| Asia-Pacífico (Tokio) | ap-northeast-1 |
| Canadá (centro) | ca-central-1 |
| China (Beijing) | cn-north-1 |
| China (Ningxia) | cn-northwest-1 |
| Europe (Frankfurt) | eu-central-1 |
| Europa (Irlanda) | eu-west-1 |
| Europa (Londres) | eu-west-2 |
| Europa (Milán) | eu-south-1 |
| Europa (París) | eu-west-3 |
| Europa (Estocolmo) | eu-north-1 |

| Nombre de la región | Región |
|-------------------------------------|---------------|
| Medio Oriente (Baréin) | me-south-1 |
| América del Sur (São Paulo) | sa-east-1 |
| AWS GovCloud (Este de EE. UU.) | us-gov-east-1 |
| AWS GovCloud (Estados Unidos-Oeste) | us-gov-west-1 |

Usando AWS ParallelCluster

Temas

- [Configuraciones de red](#)
- [Acciones de arranque personalizadas](#)
- [Uso de Amazon S3](#)
- [Uso de instancias de spot](#)
- [AWS Identity and Access Management funciones en AWS ParallelCluster](#)
- [Planificadores compatibles con AWS ParallelCluster](#)
- [AWS ParallelCluster recursos y etiquetado](#)
- [CloudWatch Panel de control de Amazon](#)
- [Integración con Amazon CloudWatch Logs](#)
- [Elastic Fabric Adapter](#)
- [Soluciones Intel Select](#)
- [Habilite Intel MPI](#)
- [Especificación de plataforma Intel HPC](#)
- [Bibliotecas de rendimiento de Arm](#)
- [Conéctese al nodo principal a través de Amazon DCV](#)
- [Uso de pcluster update](#)
- [AMI parcheo y EC2 reemplazo de instancias](#)

Configuraciones de red

AWS ParallelCluster utiliza Amazon Virtual Private Cloud (VPC) para la creación de redes. VPC proporciona una plataforma de red flexible y configurable en la que puede implementar clústeres.

Lo VPC imprescindible DNS Hostnames = yes y DNS Resolution = yes las DHCP opciones con el nombre de dominio correcto para la región. El conjunto de DHCP opciones predeterminado ya especifica lo necesario. AmazonProvidedDNS Si especificas más de un servidor de nombres de dominio, consulta los [conjuntos de DHCP opciones](#) en la Guía del VPC usuario de Amazon.

AWS ParallelCluster admite las siguientes configuraciones de alto nivel:

- Una subred para los nodos principal y de computación.
- Dos subredes: el nodo principal en una subred pública, y los nodos de computación en una subred privada. Las subredes pueden ser nuevas o existentes.

Todas estas configuraciones pueden funcionar con o sin direcciones IP públicas. AWS ParallelCluster también se puede implementar para usar un HTTP proxy para todas las AWS solicitudes. La combinación de estas configuraciones se traducen en muchos casos de implementación. Por ejemplo, puede configurar una única subred pública con todos los accesos a través de Internet. O bien, puede configurar una red totalmente privada mediante AWS Direct Connect un HTTP proxy para todo el tráfico.

Consulte los siguientes diagramas de arquitectura para ver ilustraciones de algunos de estos casos:

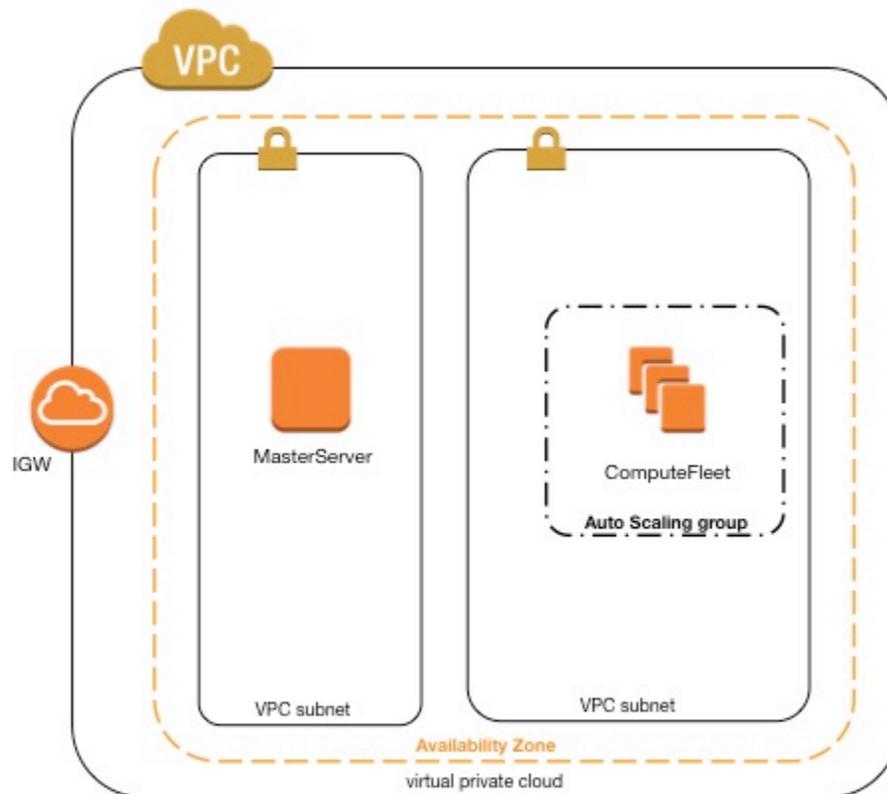
AWS ParallelCluster en una única subred pública

La configuración de esta arquitectura requiere los siguientes valores de configuración:

```
[vpc public]
vpc_id = vpc-xxxxxxx
master_subnet_id = subnet-<public>
use_public_ips = true
```

El valor de [use_public_ips](#) no se puede establecer en `false`, porque la gateway de Internet requiere que todas las instancias tengan una dirección IP única globalmente. Para obtener más información, consulta Cómo [habilitar el acceso a Internet](#) en la Guía VPC del usuario de Amazon.

AWS ParallelCluster utilizando dos subredes



La configuración para crear una nueva subred privada para instancias de computación requiere los siguientes valores de configuración:

Tenga en cuenta que todos los valores son solo ejemplos

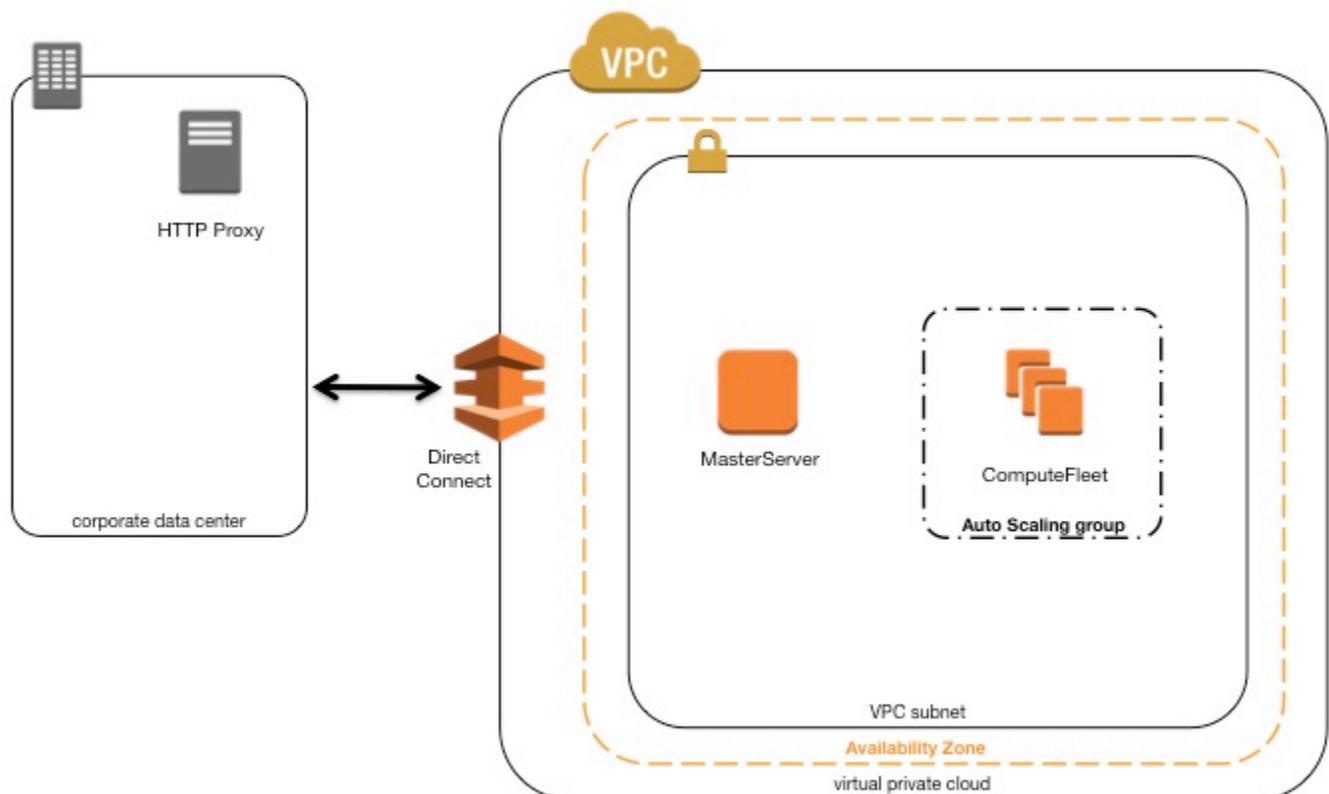
```
[vpc public-private-new]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<public>
compute_subnet_cidr = 10.0.1.0/24
```

La configuración para utilizar una red privada existente requiere los siguientes valores de configuración:

```
[vpc public-private-existing]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<public>
compute_subnet_id = subnet-<private>
```

Ambas configuraciones requieren una [NATpuerta de enlace](#) o un proxy interno para permitir el acceso web a las instancias informáticas.

AWS ParallelCluster en una única subred privada conectada mediante AWS Direct Connect



La configuración de esta arquitectura requiere los siguientes valores de configuración:

```
[cluster private-proxy]
```

```
proxy_server = http://proxy.corp.net:8080

[vpc private-proxy]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<private>
use_public_ips = false
```

Cuando `use_public_ips` está configurado en `false`, VPC debe estar configurado correctamente para usar el proxy para todo el tráfico. Se requiere acceso a Internet tanto para los nodos principales como para los de computación.

AWS ParallelCluster con **awsbatch** programador

Cuando se utiliza `awsbatch` como tipo de planificador, AWS ParallelCluster crea un entorno informático AWS Batch gestionado. El AWS Batch entorno se encarga de administrar las instancias de contenedor de Amazon Elastic Container Service (AmazonECS), que se lanzan en `compute_subnet`. AWS Batch Para funcionar correctamente, las instancias ECS contenedoras de Amazon necesitan acceso a una red externa para comunicarse con el punto de conexión del ECS servicio de Amazon. Esto se traduce en los siguientes casos:

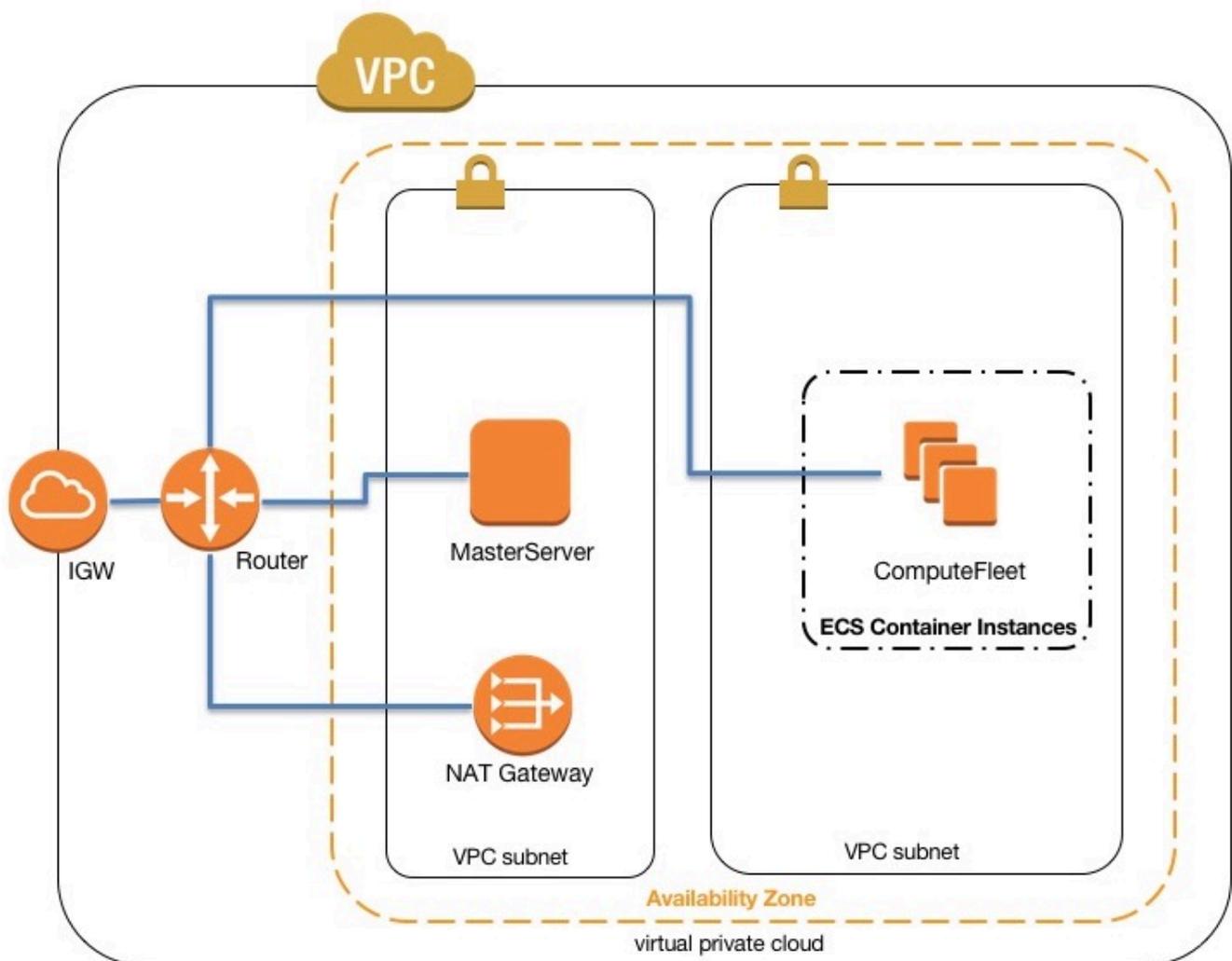
- `compute_subnet` Utiliza una NAT puerta de enlace para acceder a Internet. (Recomendamos este enfoque).
- Las instancias que se lanzan en la `compute_subnet` tienen direcciones IP públicas y pueden llegar a Internet a través de una gateway de Internet.

Además, si le interesan los trabajos paralelos de varios nodos (de los [documentos de AWS Batch](#)):

AWS Batch los trabajos paralelos de varios nodos utilizan el modo de ECS `awsvpc` red de Amazon, que proporciona a los contenedores de trabajos paralelos de varios nodos las mismas propiedades de red que las instancias de AmazonEC2. Cada contenedor de trabajos paralelos de varios nodos tiene su propia interfaz de red elástica, una dirección IP privada principal y un nombre de DNS host interno. La interfaz de red se crea en la misma VPC subred de Amazon que su recurso informático anfitrión. Los grupos de seguridad que se hayan aplicado a los recursos de computación se aplicarán también a ella.

Cuando se utiliza Amazon ECS Task Networking, el modo de `awsvpc` red no proporciona interfaces de red elásticas con direcciones IP públicas para las tareas que utilizan el tipo de EC2 lanzamiento de Amazon. Para acceder a Internet, las tareas que utilizan el tipo de EC2 lanzamiento de Amazon deben lanzarse en una subred privada que esté configurada para usar una NAT puerta de enlace.

Debe configurar una NAT puerta de enlace para permitir que el clúster ejecute trabajos paralelos de varios nodos.



Para obtener más información, consulte los temas siguientes:

- [AWS Batch entornos de cómputo gestionados](#)
- [AWS Batch trabajos paralelos de varios nodos](#)
- [Red de ECS tareas de Amazon con el modo awsvpc de red](#)

Acciones de arranque personalizadas

AWS ParallelCluster puede ejecutar código arbitrario antes (preinstalación) o después (después de la instalación) de la acción de arranque principal al crear el clúster. En la mayoría de los casos,

este código se almacena en Amazon Simple Storage Service (Amazon S3) y se accede a él a través HTTPS de una conexión. El código se ejecuta como raíz y puede estar en cualquier lenguaje de script compatible con el sistema operativo del clúster. A menudo, el código está en Bash o Python.

Las acciones de preinstalación se invocan antes de iniciar cualquier acción de arranque del despliegue del clúster, como la configuración de NAT Amazon Elastic Block Store EBS (Amazon) o el programador. La modificación del almacenamiento o añadir usuarios o paquetes adicionales son acciones de preinstalación típicas.

Las acciones posteriores a la instalación se ejecutan una vez finalizados los procesos de arranque del clúster. Las acciones posteriores a la instalación son las últimas acciones que se llevan a cabo antes de que una instancia se considere completamente configurada y completa. Algunas acciones incluyen la modificación de la configuración del programador, del almacenamiento o de los paquetes.

Puede pasar argumentos a los scripts especificándolos durante la configuración. Estos se transfieren con comillas dobles a las acciones previas o posteriores a la instalación.

Si una acción previa o posterior a la instalación produce un error, el arranque de instancia también falla. El éxito aparece con un código de salida de 0. Cualquier otro código de salida indica que se ha producido un error en el arranque de la instancia.

Puede diferenciar entre el nodo principal en ejecución y el nodo de cómputo. Obtenga el archivo `/etc/parallelcluster/cfnconfig` y evalúe la variable de entorno `cfn_node_type`, cuyos valores posibles son "MasterServer" y "ComputeFleet" para el nodo principal y de computación, respectivamente.

```
#!/bin/bash

. "/etc/parallelcluster/cfnconfig"

case "${cfn_node_type}" in
    MasterServer)
        echo "I am the head node" >> /tmp/head.txt
        ;;
    ComputeFleet)
        echo "I am a compute node" >> /tmp/compute.txt
        ;;
    *)
        ;;
esac
```

Configuración

Se utilizan las siguientes opciones de configuración para definir acciones de preinstalación o posinstalación y argumentos.

```
# URL to a preinstall script. This is run before any of the boot_as_* scripts are run
# (no default)
pre_install = https://<bucket-name>.s3.amazonaws.com/my-pre-install-script.sh
# Arguments to be passed to preinstall script
# (no default)
pre_install_args = argument-1 argument-2
# URL to a postinstall script. This is run after any of the boot_as_* scripts are run
# (no default)
post_install = https://<bucket-name>.s3.amazonaws.com/my-post-install-script.sh
# Arguments to be passed to postinstall script
# (no default)
post_install_args = argument-3 argument-4
```

Argumentos

Los dos primeros argumentos, \$0 y \$1, se reservan para el nombre del script y la dirección url.

```
$0 => the script name
$1 => s3 url
$n => args set by pre/post_install_args
```

Ejemplo

Los siguientes pasos crean un script de posinstalación sencillo que instala los paquetes R en un clúster.

1. Cree un script.

```
#!/bin/bash

echo "post-install script has $# arguments"
for arg in "$@"
do
    echo "arg: ${arg}"
done
```

```
yum -y install "${@:2}"
```

2. Cargue el script con los permisos correctos en Amazon S3. Si los permisos de lectura públicos no son adecuados para usted, utilice uno de [s3_read_resource](#) los dos [s3_read_write_resource](#) parámetros para conceder el acceso. Para obtener más información, consulte [Uso de Amazon S3](#).

```
$ aws s3 cp --acl public-read /path/to/myscript.sh s3://<bucket-name>/myscript.sh
```

Important

Si el script se editó en Windows, los finales de línea deben CRLF cambiarse de LF antes de cargar el script en Amazon S3.

3. Actualice la AWS ParallelCluster configuración para incluir la nueva acción posterior a la instalación.

```
[cluster default]
...
post_install = https://<bucket-name>.s3.amazonaws.com/myscript.sh
post_install_args = 'R curl wget'
```

Si el bucket no tiene permiso de lectura pública, utilícelo s3 como protocolo. URL

```
[cluster default]
...
post_install = s3://<bucket-name>/myscript.sh
post_install_args = 'R curl wget'
```

4. Lance el clúster.

```
$ pcluster create mycluster
```

5. Verifique el resultado.

```
$ less /var/log/cfn-init.log
2019-04-11 10:43:54,588 [DEBUG] Command runpostinstall output: post-install script
has 4 arguments
arg: s3://<bucket-name>/test.sh
```

```
arg: R
arg: curl
arg: wget
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

Uso de Amazon S3

Para conceder permiso a los recursos del clúster para acceder a los buckets de Amazon S3, especifique el bucket ARNs en los [s3_read_write_resource](#) parámetros [s3_read_resource](#) y de la AWS ParallelCluster configuración. Para obtener más información sobre cómo controlar el acceso con AWS ParallelCluster, consulte [AWS Identity and Access Management funciones en AWS ParallelCluster](#).

```
# Specify Amazon S3 resource which AWS ParallelCluster nodes will be granted read-only
access
# (no default)
s3_read_resource = arn:aws:s3:::my_corporate_bucket*
# Specify Amazon S3 resource which AWS ParallelCluster nodes will be granted read-write
access
# (no default)
s3_read_write_resource = arn:aws:s3:::my_corporate_bucket/*
```

Ambos parámetros aceptan Amazon S3 * o uno válidoARN. Para obtener información sobre cómo especificar Amazon S3ARNs, consulte el [ARNformato Amazon S3](#) en Referencia general de AWS.

Ejemplos

En el siguiente ejemplo, se le ofrece acceso de lectura a cualquier objeto del bucket de Amazon S3 my_corporate_bucket.

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket/*
```

En el siguiente ejemplo, se le ofrece acceso de lectura al bucket, pero no se le permite leer elementos del bucket.

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket
```

Este último ejemplo le ofrece acceso de lectura al bucket y a los elementos almacenados en el bucket.

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket*
```

Uso de instancias de spot

AWS ParallelCluster usa instancias puntuales si la configuración del clúster ha establecido `cluster_type = spot`. Las instancias de spot son más rentables que las instancias bajo demanda, pero pueden interrumpirse. El efecto de la interrupción varía según el programador utilizado. Puede ser útil aprovechar los avisos de interrupción de las instancias puntuales, que proporcionan un aviso de dos minutos antes de que Amazon EC2 deba detener o cancelar su instancia puntual. Para obtener más información, consulte [Interrupciones de instancias puntuales](#) en la Guía EC2 del usuario de Amazon. En las secciones siguientes se describen tres escenarios en los que las instancias de spot pueden interrumpirse.

Note

El uso de instancias de spot requiere que el rol de `AWSServiceRoleForEC2Spot` vinculado al servicio esté en su cuenta. Para crear este rol en su cuenta mediante el AWS CLI, ejecute el siguiente comando:

```
aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Para obtener más información, consulte [Función vinculada a servicios para solicitudes de instancias puntuales](#) en la Guía EC2 del usuario de Amazon.

Escenario 1: Se interrumpe una instancia de spot sin trabajos en ejecución

Cuando se produce esta interrupción, AWS ParallelCluster intenta reemplazar la instancia si la cola del programador tiene trabajos pendientes que requieren instancias adicionales o si el número de instancias activas es inferior al establecido. `initial_queue_size` Si AWS ParallelCluster no puede aprovisionar nuevas instancias, se repite periódicamente una solicitud de nuevas instancias.

Escenario 2: Se interrumpe una instancia de spot que ejecuta trabajos de un solo nodo

El comportamiento de esta interrupción depende del programador que se esté utilizando.

Slurm

El trabajo falla con un código de `NODE_FAIL` estado igual a `y` y se vuelve a poner en cola (a menos que `--no-requeue` se especifique al enviar el trabajo). Si el nodo es estático, se reemplaza. Si el nodo es un nodo dinámico, se termina y se restablece. Para obtener más información sobre cómo incluir el `--no-requeue` parámetros `batch`, consulte [sbatch](#) en la documentación de Slurm.

Note

Este comportamiento cambió en la AWS ParallelCluster versión 2.9.0. Las versiones anteriores finalizaban el trabajo con un código de estado de `NODE_FAIL` y el nodo se eliminaba de la cola del programador.

SGE

Note

Esto solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

Se termina el trabajo. Si el trabajo ha habilitado el indicador "rerun" (mediante `qsub -r yes` o `qalter -r yes`) o la cola tiene el valor de `rerun` establecido en `TRUE`, el trabajo se reprograma. La instancia de computación se elimina de la cola del programador. Este comportamiento proviene de los siguientes parámetros de SGE configuración:

- `reschedule_unknown 00:00:30`
- `ENABLE_FORCED_QDEL_IF_UNKNOWN`
- `ENABLE_RESCHEDULE_KILL=1`

Torque

Note

Esto solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

El trabajo se elimina del sistema y el nodo se elimina del programador. El trabajo no se vuelve a ejecutar. Si se están ejecutando varios trabajos en la instancia cuando se interrumpe, se puede agotar el tiempo de espera de Torque durante la eliminación del nodo. Puede aparecer un error en el archivo de registro [sqswatcher](#). Esto no afecta a la lógica de escalado, y los posteriores reintentos realizan una limpieza adecuada.

Escenario 3: Se interrumpe una instancia de spot que ejecuta trabajos de varios nodos

El comportamiento de esta interrupción depende del programador que se esté utilizando.

Slurm

El trabajo falla con un código de `NODE_FAIL` estado igual a y se vuelve a poner en cola (a menos que `--no-requeue` se especifique al enviar el trabajo). Si el nodo es estático, se reemplaza. Si el nodo es un nodo dinámico, se termina y se restablece. Otros nodos que estaban ejecutando los trabajos terminados se pueden asignar a otros trabajos pendientes o reducirse verticalmente una vez transcurrido el tiempo de [scaledown_idletime](#) configurado .

Note

Este comportamiento cambió en la AWS ParallelCluster versión 2.9.0. Las versiones anteriores finalizaban el trabajo con un código de estado de `NODE_FAIL` y el nodo se eliminaba de la cola del programador. Otros nodos que estaban ejecutando los trabajos terminados pueden reducirse verticalmente una vez transcurrido el tiempo de [scaledown_idletime](#) configurado.

SGE

Note

Esto solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

El trabajo no se termina y continúa ejecutándose en los nodos restantes. El nodo de computación se elimina de la cola del programador, pero aparecerá en la lista de hosts como un nodo huérfano y no disponible.

El usuario debe eliminar el trabajo cuando esto ocurra (`qdel <jobid>`). El nodo aún se muestra en la lista de hosts (`qhost`), aunque esto no afecta a AWS ParallelCluster. Para quitar el host de la lista, puede ejecutar el siguiente comando después de reemplazar la instancia.

```
sudo -- bash -c 'source /etc/profile.d/sge.sh; qconf -dattr hostgroup  
hostlist <hostname> @allhosts; qconf -de <hostname>'
```

Torque

Note

Esto solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

El trabajo se elimina del sistema y el nodo se elimina del programador. El trabajo no se vuelve a ejecutar. Si se están ejecutando varios trabajos en la instancia cuando se interrumpe, se puede agotar el tiempo de espera de Torque durante la eliminación del nodo. Puede aparecer un error en el archivo de registro [sqswatcher](#). Esto no afecta a la lógica de escalado, y los posteriores reintentos realizan una limpieza adecuada.

Para obtener más información sobre las instancias puntuales, consulte [Instancias puntuales](#) en la Guía del EC2 usuario de Amazon.

AWS Identity and Access Management funciones en AWS ParallelCluster

AWS ParallelCluster usa roles AWS Identity and Access Management (IAM) para EC2 que Amazon permita a las instancias acceder a AWS los servicios para el despliegue y el funcionamiento de un clúster. De forma predeterminada, el IAM rol de Amazon EC2 se crea cuando se crea el clúster. Esto significa que el usuario que crea el clúster debe tener el nivel de permisos adecuado, tal como se describe en las siguientes secciones.

AWS ParallelCluster utiliza varios AWS servicios para implementar y operar un clúster. Consulte la lista completa en la sección [AWS Servicios de AWS que se utilizan en AWS ParallelCluster](#).

Puede realizar un seguimiento de los cambios en las políticas de ejemplo de [AWS ParallelCluster la documentación sobre GitHub](#).

Temas

- [Configuración predeterminada para la creación de clústeres](#)
- [Uso de un IAM rol existente para Amazon EC2](#)
- [AWS ParallelCluster ejemplos de políticas de instancia y usuario](#)

Configuración predeterminada para la creación de clústeres

Cuando utilizas la configuración predeterminada para la creación de clústeres, el clúster crea EC2 un IAM rol predeterminado para Amazon. El usuario que crea el clúster debe tener el nivel de permisos adecuado para crear todos los recursos necesarios para lanzar el clúster, incluido un rol de para . Esto incluye la creación de un IAM rol para AmazonEC2. Normalmente, el usuario debe tener los permisos de una política AdministratorAccess gestionada cuando utiliza la configuración predeterminada. Para obtener información sobre las políticas administradas, consulte [las políticas AWS administradas](#) en la Guía del IAM usuario.

Uso de un IAM rol existente para Amazon EC2

En lugar de la configuración predeterminada, puede utilizar una configuración existente [ec2_iam_role](#) para crear un clúster, pero debe definir la IAM política y la función antes de intentar lanzar el clúster. Normalmente, eliges un IAM rol existente para Amazon EC2 a fin de minimizar los permisos que se conceden a los usuarios cuando lanzan clústeres. [AWS ParallelCluster ejemplos](#)

[de políticas de instancia y usuario](#) incluyen los permisos mínimos requeridos AWS ParallelCluster y sus funciones. Debe crear políticas y funciones como políticas individuales IAM y, a continuación, adjuntar las funciones y políticas a los recursos correspondientes. Algunas de las políticas de roles pueden aumentar de tamaño y provocar errores de cuota. Para obtener más información, consulte [Solución de problemas IAM de tamaño de la política](#). En las políticas, sustituya `<REGION>`, `<AWS ACCOUNT ID>`, y cadenas similares con los valores adecuados.

Si su intención es añadir políticas adicionales a la configuración predeterminada de los nodos del clúster, le recomendamos que incorpore las IAM políticas personalizadas adicionales con la [additional_iam_policies](#) configuración en lugar de [ec2_iam_role](#) utilizarla.

AWS ParallelCluster ejemplos de políticas de instancia y usuario

Las siguientes políticas de ejemplo incluyen Amazon Resource Names (ARNs) para los recursos. Si está trabajando en las particiones AWS GovCloud (US) o en las de AWS China, ARNs debe cambiarlas. En concreto, deben cambiarse de «arn:aws» a «arn:aws-us-gov» para la AWS GovCloud (US) partición o «arn:aws-cn» para la partición de China. AWS Para obtener más información, consulte [Nombres de recursos de Amazon \(ARNs\) en AWS GovCloud \(US\) las regiones](#) en la Guía del AWS GovCloud (US) usuario y, [ARNs para AWS los servicios en China](#), en Introducción a AWS los servicios en China.

Estas políticas incluyen los permisos mínimos que actualmente requieren sus funciones y recursos. AWS ParallelCluster Algunas de las políticas de roles pueden aumentar de tamaño y provocar errores de cuota. Para obtener más información, consulte [Solución de problemas IAM de tamaño de la política](#).

Temas

- [ParallelClusterInstancePolicy usando SGE, Slurm, o Torque](#)
- [ParallelClusterInstancePolicy con awsbatch](#)
- [ParallelClusterUserPolicy usando Slurm](#)
- [ParallelClusterUserPolicy usando SGE o Torque](#)
- [ParallelClusterUserPolicy con awsbatch](#)
- [ParallelClusterLambdaPolicy usando SGE, Slurm, o Torque](#)
- [ParallelClusterLambdaPolicy con awsbatch](#)
- [ParallelClusterUserPolicy para usuarios](#)

ParallelClusterInstancePolicy usando SGE, Slurm, o Torque

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores. Puede seguir utilizándolos en las versiones anteriores a la 2.11.4 (inclusive), pero no son aptas para recibir actualizaciones futuras ni asistencia para la solución de problemas por parte de los equipos de AWS servicio y AWS soporte.

Temas

- [ParallelClusterInstancePolicy usando Slurm](#)
- [ParallelClusterInstancePolicy usando SGE o Torque](#)

ParallelClusterInstancePolicy usando Slurm

El siguiente ejemplo establece el ParallelClusterInstancePolicy uso Slurm como programador.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:AttachVolume",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstances",
        "ec2:DescribeRegions",
        "ec2:TerminateInstances",
        "ec2:DescribeLaunchTemplates",
        "ec2:CreateTags"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "EC2"
    }
  ]
}
```

```

    },
    {
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:subnet/<COMPUTE SUBNET ID>",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:network-interface/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*",
        "arn:aws:ec2:<REGION>::image/<IMAGE ID>",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:key-pair/<KEY NAME>",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:security-group/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:launch-template/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:placement-group*"
      ],
      "Effect": "Allow",
      "Sid": "EC2RunInstances"
    },
    {
      "Action": [
        "dynamodb>ListTables"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "DynamoDBList"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackResource",
        "cloudformation:SignalResource"
      ],
      "Resource": [
        "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/parallelcluster-*/*"
      ],
      "Effect": "Allow",
      "Sid": "CloudFormation"
    },
    {
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:Query",

```

```

        "dynamodb:GetItem",
        "dynamodb:BatchWriteItem",
        "dynamodb>DeleteItem",
        "dynamodb:DescribeTable"
    ],
    "Resource": [
        "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "DynamoDBTable"
},
{
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "S3GetObject"
},
{
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow",
    "Sid": "IAMPassRole",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "ec2.amazonaws.com"
            ]
        }
    }
},
{
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::dcv-license.<REGION>/*"
    ]
}

```

```

    ],
    "Effect": "Allow",
    "Sid": "DcvLicense"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "GetClusterConfig"
  },
  {
    "Action": [
      "fsx:DescribeFileSystems"
    ],
    "Resource": [
      "*"
    ],
    "Effect": "Allow",
    "Sid": "FSx"
  },
  {
    "Action": [
      "logs:CreateLogStream",
      "logs:PutLogEvents"
    ],
    "Resource": [
      "*"
    ],
    "Effect": "Allow",
    "Sid": "CWLogs"
  },
  {
    "Action": [
      "route53:ChangeResourceRecordSets"
    ],
    "Resource": [
      "arn:aws:route53:::hostedzone/*"
    ],
    "Effect": "Allow",

```

```

        "Sid": "Route53"
    }
]
}

```

ParallelClusterInstancePolicy usando SGE o Torque

El siguiente ejemplo establece el ParallelClusterInstancePolicy uso SGE o Torque como programador.

Note

Esta política solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:AttachVolume",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstances",
        "ec2:DescribeRegions",
        "ec2:TerminateInstances",
        "ec2:DescribeLaunchTemplates",
        "ec2:CreateTags"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "EC2"
    },
    {
      "Action": "ec2:RunInstances",
      "Resource": [

```

```

        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:subnet/<COMPUTE SUBNET ID>",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:network-interface/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*",
        "arn:aws:ec2:<REGION>::image/<IMAGE ID>",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:key-pair/<KEY NAME>",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:security-group/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:launch-template/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:placement-group*"
    ],
    "Effect": "Allow",
    "Sid": "EC2RunInstances"
},
{
    "Action": [
        "dynamodb:ListTables"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow",
    "Sid": "DynamoDBList"
},
{
    "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility",
        "sqs>DeleteMessage",
        "sqs:GetQueueUrl"
    ],
    "Resource": [
        "arn:aws:sqs:<REGION>:<AWS ACCOUNT ID>:parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "SQSQueue"
},
{
    "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
        "autoscaling:SetDesiredCapacity",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:DescribeTags",

```

```

        "autoscaling:SetInstanceHealth"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow",
    "Sid": "Autoscaling"
},
{
    "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackResource",
        "cloudformation:SignalResource"
    ],
    "Resource": [
        "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormation"
},
{
    "Action": [
        "dynamodb:PutItem",
        "dynamodb:Query",
        "dynamodb:GetItem",
        "dynamodb:BatchWriteItem",
        "dynamodb>DeleteItem",
        "dynamodb:DescribeTable"
    ],
    "Resource": [
        "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "DynamoDBTable"
},
{
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
    ],
    "Effect": "Allow",

```

```

        "Sid": "S3GetObj"
    },
    {
        "Action": [
            "sqs:ListQueues"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow",
        "Sid": "SQSList"
    },
    {
        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow",
        "Sid": "IAMPassRole",
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    },
    {
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::dcv-license.<REGION>/*"
        ],
        "Effect": "Allow",
        "Sid": "DcvLicense"
    },
    {
        "Action": [
            "s3:GetObject",
            "s3:GetObjectVersion"
        ],

```

```
    "Resource": [
      "arn:aws:s3:::parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "GetClusterConfig"
  },
  {
    "Action": [
      "fsx:DescribeFileSystems"
    ],
    "Resource": [
      "*"
    ],
    "Effect": "Allow",
    "Sid": "FSx"
  },
  {
    "Action": [
      "logs:CreateLogStream",
      "logs:PutLogEvents"
    ],
    "Resource": [
      "*"
    ],
    "Effect": "Allow",
    "Sid": "CWLogs"
  },
  {
    "Action": [
      "route53:ChangeResourceRecordSets"
    ],
    "Resource": [
      "arn:aws:route53:::hostedzone/*"
    ],
    "Effect": "Allow",
    "Sid": "Route53"
  }
]
```

ParallelClusterInstancePolicy con awsbatch

En el ejemplo siguiente se establece ParallelClusterInstancePolicy, utilizando awsbatch como programador. Debe incluir las mismas políticas que se asignan a las BatchUserRole definidas en la pila AWS Batch AWS CloudFormation anidada. BatchUserRoleARNSe proporciona como salida de pila. En este ejemplo, `<RESOURCES S3 BUCKET>` es el valor de la `cluster_resource_bucket` configuración; si no `cluster_resource_bucket` se especifica, entonces `<RESOURCES S3 BUCKET>` es `parallelcluster-*`. En el siguiente ejemplo se muestra un resumen de los permisos necesarios:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "batch:RegisterJobDefinition",
        "logs:GetLogEvents"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "batch:SubmitJob",
        "cloudformation:DescribeStacks",
        "ecs:ListContainerInstances",
        "ecs:DescribeContainerInstances",
        "logs:FilterLogEvents",
        "s3:PutObject",
        "s3:Get*",
        "s3:DeleteObject",
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-
definition/<AWS_BATCH_STACK - JOB_DEFINITION_SERIAL_NAME>:1",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-
definition/<AWS_BATCH_STACK - JOB_DEFINITION_MNP_NAME>*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/<AWS_BATCH_STACK -
JOB_QUEUE_NAME>"
      ]
    }
  ]
}
```

```

        "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/<STACK NAME>/
*",
        "arn:aws:s3:::<RESOURCES S3 BUCKET>/batch/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/<AWS_BATCH_STACK - JOB_ROLE>",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/<ECS COMPUTE
ENVIRONMENT>",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/*",
        "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-
stream:*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "s3:List*"
    ],
    "Resource": [
        "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "batch:DescribeJobQueues",
        "batch:TerminateJob",
        "batch:DescribeJobs",
        "batch:CancelJob",
        "batch:DescribeJobDefinitions",
        "batch:ListJobs",
        "batch:DescribeComputeEnvironments"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:DescribeInstances",
        "ec2:AttachVolume",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2"
}

```

```

    },
    {
      "Action": [
        "cloudformation:DescribeStackResource",
        "cloudformation:SignalResource"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    },
    {
      "Action": [
        "fsx:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "FSx"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs:CreateLogStream"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "CWLogs"
    }
  ]
}

```

ParallelClusterUserPolicy usando Slurm

El siguiente ejemplo establece el `ParallelClusterUserPolicy`, mediante Slurm como programador. En este ejemplo, «**<RESOURCES S3 BUCKET>**» es el valor de la [cluster_resource_bucket](#) configuración; si no [cluster_resource_bucket](#) se especifica, entonces «**<RESOURCES S3 BUCKET>**» es «parallelcluster-*».

Note

Si usa un rol personalizado `ec2_iam_role = <role_name>`, debe cambiar el IAM recurso para incluir el nombre de ese rol de:

```
"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*
```

Para:

```
"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/<role_name>"
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeKeyPairs",
        "ec2:DescribeRegions",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribePlacementGroups",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeSnapshots",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeAddresses",
        "ec2:CreateTags",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Describe"
    },
    {
      "Action": [
        "ec2:CreateVpc",
        "ec2:ModifyVpcAttribute",
        "ec2:DescribeNatGateways",
```

```

        "ec2:CreateNatGateway",
        "ec2:DescribeInternetGateways",
        "ec2:CreateInternetGateway",
        "ec2:AttachInternetGateway",
        "ec2:DescribeRouteTables",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:AssociateRouteTable",
        "ec2:CreateSubnet",
        "ec2:ModifySubnetAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "NetworkingEasyConfig"
},
{
    "Action": [
        "ec2:CreateVolume",
        "ec2:RunInstances",
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateNetworkInterface",
        "ec2:CreateSecurityGroup",
        "ec2:ModifyVolumeAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2>DeleteNetworkInterface",
        "ec2>DeleteVolume",
        "ec2:TerminateInstances",
        "ec2>DeleteSecurityGroup",
        "ec2:DisassociateAddress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:ReleaseAddress",
        "ec2:CreatePlacementGroup",
        "ec2>DeletePlacementGroup"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Modify"
},
{

```

```
    "Action": [
      "autoscaling:CreateAutoScalingGroup",
      "ec2:CreateLaunchTemplate",
      "ec2:CreateLaunchTemplateVersion",
      "ec2:ModifyLaunchTemplate",
      "ec2>DeleteLaunchTemplate",
      "ec2:DescribeLaunchTemplates",
      "ec2:DescribeLaunchTemplateVersions"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ScalingModify"
  },
  {
    "Action": [
      "dynamodb:DescribeTable",
      "dynamodb:ListTagsOfResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "DynamoDBDescribe"
  },
  {
    "Action": [
      "dynamodb:CreateTable",
      "dynamodb>DeleteTable",
      "dynamodb:GetItem",
      "dynamodb:PutItem",
      "dynamodb:Query",
      "dynamodb:TagResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "DynamoDBModify"
  },
  {
    "Action": [
      "route53:ChangeResourceRecordSets",
      "route53:ChangeTagsForResource",
      "route53:CreateHostedZone",
      "route53>DeleteHostedZone",
      "route53:GetChange",
      "route53:GetHostedZone",
      "route53:ListResourceRecordSets",
```

```

        "route53:ListQueryLoggingConfigs"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
},
{
    "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStacks",
        "cloudformation:GetTemplate"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudFormationDescribe"
},
{
    "Action": [
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:UpdateStack"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Sid": "CloudFormationModify"
},
{
    "Action": [
        "s3:*"
    ],
    "Resource": [
        "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
},
{
    "Action": [
        "s3:Get*",
        "s3:List*"
    ],

```

```

    "Resource": [
      "arn:aws:s3:::<REGION>-aws-parallelcluster*"
    ],
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
  },
  {
    "Action": [
      "s3:DeleteBucket",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow",
    "Sid": "S3Delete"
  },
  {
    "Action": [
      "iam:PassRole",
      "iam:CreateRole",
      "iam>DeleteRole",
      "iam:GetRole",
      "iam:TagRole",
      "iam:SimulatePrincipalPolicy"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/<PARALLELCLUSTER EC2 ROLE NAME>",
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "IAMModify"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": [
          "fsx.amazonaws.com",
          "s3.data-source.lustre.fsx.amazonaws.com"
        ]
      }
    },
    "Action": [

```

```

        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/*",
    "Effect": "Allow",
    "Sid": "IAMServiceLinkedRole"
},
{
    "Action": [
        "iam:CreateInstanceProfile",
        "iam>DeleteInstanceProfile"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
    "Effect": "Allow",
    "Sid": "IAMCreateInstanceProfile"
},
{
    "Action": [
        "iam:AddRoleToInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:GetRolePolicy",
        "iam:GetPolicy",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "IAMInstanceProfile"
},
{
    "Action": [
        "elasticfilesystem:DescribeMountTargets",
        "elasticfilesystem:DescribeMountTargetSecurityGroups",
        "ec2:DescribeNetworkInterfaceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EFSDescribe"
},
{
    "Action": [
        "ssm:GetParametersByPath"
    ],

```

```
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SSMDescribe"
  },
  {
    "Action": [
      "fsx:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "FSx"
  },
  {
    "Action": [
      "elasticfilesystem:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EFS"
  },
  {
    "Action": [
      "logs:DeleteLogGroup",
      "logs:PutRetentionPolicy",
      "logs:DescribeLogGroups",
      "logs:CreateLogGroup",
      "logs:TagResource",
      "logs:UntagResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogs"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda>DeleteFunction",
      "lambda:GetFunctionConfiguration",
      "lambda:GetFunction",
      "lambda:InvokeFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
      "lambda:TagResource",
      "lambda:ListTags",
```

```

        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
        "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Sid": "CloudWatch",
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutDashboard",
        "cloudwatch:ListDashboards",
        "cloudwatch>DeleteDashboards",
        "cloudwatch:GetDashboard"
    ],
    "Resource": "*"
}
]
}

```

ParallelClusterUserPolicy usando SGE o Torque

Note

Esta sección solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

El siguiente ejemplo establece el `ParallelClusterUserPolicy`, mediante SGE o Torque como programador. En este ejemplo, «`<RESOURCES S3 BUCKET>`» es el valor de la `cluster_resource_bucket` configuración; si no `cluster_resource_bucket` se especifica, entonces «`<RESOURCES S3 BUCKET>`» es «`parallelcluster-*`».

Note

Si usa un rol personalizado `ec2_iam_role = <role_name>`, debe cambiar el IAM recurso para incluir el nombre de ese rol de:

```
"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*
```

Para:

```
"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/<role_name>"
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeKeyPairs",
        "ec2:DescribeRegions",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribePlacementGroups",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeSnapshots",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeAddresses",
        "ec2:CreateTags",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Describe"
    },
    {
      "Action": [
        "ec2:CreateVpc",
        "ec2:ModifyVpcAttribute",
        "ec2:DescribeNatGateways",
        "ec2:CreateNatGateway",
        "ec2:DescribeInternetGateways",
        "ec2:CreateInternetGateway",
        "ec2:AttachInternetGateway",

```

```

        "ec2:DescribeRouteTables",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:AssociateRouteTable",
        "ec2:CreateSubnet",
        "ec2:ModifySubnetAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "NetworkingEasyConfig"
},
{
    "Action": [
        "ec2:CreateVolume",
        "ec2:RunInstances",
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateNetworkInterface",
        "ec2:CreateSecurityGroup",
        "ec2:ModifyVolumeAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2>DeleteNetworkInterface",
        "ec2>DeleteVolume",
        "ec2:TerminateInstances",
        "ec2>DeleteSecurityGroup",
        "ec2:DisassociateAddress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:ReleaseAddress",
        "ec2:CreatePlacementGroup",
        "ec2>DeletePlacementGroup"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Modify"
},
{
    "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeAutoScalingInstances"
    ],

```

```

    "Resource": "*",
    "Effect": "Allow",
    "Sid": "AutoScalingDescribe"
  },
  {
    "Action": [
      "autoscaling:CreateAutoScalingGroup",
      "ec2:CreateLaunchTemplate",
      "ec2:CreateLaunchTemplateVersion",
      "ec2:ModifyLaunchTemplate",
      "ec2>DeleteLaunchTemplate",
      "ec2:DescribeLaunchTemplates",
      "ec2:DescribeLaunchTemplateVersions",
      "autoscaling:PutNotificationConfiguration",
      "autoscaling:UpdateAutoScalingGroup",
      "autoscaling:PutScalingPolicy",
      "autoscaling:DescribeScalingActivities",
      "autoscaling>DeleteAutoScalingGroup",
      "autoscaling>DeletePolicy",
      "autoscaling:DisableMetricsCollection",
      "autoscaling:EnableMetricsCollection"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "AutoScalingModify"
  },
  {
    "Action": [
      "dynamodb:DescribeTable",
      "dynamodb:ListTagsOfResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "DynamoDBDescribe"
  },
  {
    "Action": [
      "dynamodb:CreateTable",
      "dynamodb>DeleteTable",
      "dynamodb:GetItem",
      "dynamodb:PutItem",
      "dynamodb:Query",
      "dynamodb:TagResource"
    ],

```

```
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "DynamoDBModify"
  },
  {
    "Action": [
      "sqs:GetQueueAttributes"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SQSDescribe"
  },
  {
    "Action": [
      "sqs:CreateQueue",
      "sqs:SetQueueAttributes",
      "sqs>DeleteQueue",
      "sqs:TagQueue"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SQSModify"
  },
  {
    "Action": [
      "sns:ListTopics",
      "sns:GetTopicAttributes"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SNSDescribe"
  },
  {
    "Action": [
      "sns:CreateTopic",
      "sns:Subscribe",
      "sns:Unsubscribe",
      "sns>DeleteTopic"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SNSModify"
  },
  {
```

```

    "Action": [
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStackResources",
      "cloudformation:DescribeStacks",
      "cloudformation:ListStacks",
      "cloudformation:GetTemplate"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudFormationDescribe"
  },
  {
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:UpdateStack"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Sid": "CloudFormationModify"
  },
  {
    "Action": [
      "s3:*"
    ],
    "Resource": [
      "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
  },
  {
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": [
      "arn:aws:s3:::<REGION>-aws-parallelcluster*"
    ],
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
  },
  {

```

```

    "Action": [
      "s3:DeleteBucket",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow",
    "Sid": "S3Delete"
  },
  {
    "Action": [
      "iam:PassRole",
      "iam:CreateRole",
      "iam>DeleteRole",
      "iam:GetRole",
      "iam:TagRole",
      "iam:SimulatePrincipalPolicy"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/<PARALLELCLUSTER EC2 ROLE NAME>",
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "IAMModify"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": [
          "fsx.amazonaws.com",
          "s3.data-source.lustre.fsx.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/*",
    "Effect": "Allow",
    "Sid": "IAMServiceLinkedRole"
  },
  {

```

```

    "Action": [
      "iam:CreateInstanceProfile",
      "iam>DeleteInstanceProfile"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
    "Effect": "Allow",
    "Sid": "IAMCreateInstanceProfile"
  },
  {
    "Action": [
      "iam:AddRoleToInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile",
      "iam:GetRolePolicy",
      "iam:GetPolicy",
      "iam:AttachRolePolicy",
      "iam:DetachRolePolicy",
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "IAMInstanceProfile"
  },
  {
    "Action": [
      "elasticfilesystem:DescribeMountTargets",
      "elasticfilesystem:DescribeMountTargetSecurityGroups",
      "ec2:DescribeNetworkInterfaceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EFSDescribe"
  },
  {
    "Action": [
      "ssm:GetParametersByPath"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SSMDescribe"
  },
  {
    "Action": [
      "fsx:*"
    ]
  }

```

```

    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "FSx"
  },
  {
    "Action": [
      "elasticfilesystem:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EFS"
  },
  {
    "Action": [
      "logs:DeleteLogGroup",
      "logs:PutRetentionPolicy",
      "logs:DescribeLogGroups",
      "logs:CreateLogGroup",
      "logs:TagResource",
      "logs:UntagResource"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogs"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda>DeleteFunction",
      "lambda:GetFunctionConfiguration",
      "lambda:GetFunction",
      "lambda:InvokeFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
      "lambda:TagResource",
      "lambda:ListTags",
      "lambda:UntagResource"
    ],
    "Resource": [
      "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
      "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",

```

```

        "Sid": "Lambda"
    },
    {
        "Sid": "CloudWatch",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:PutDashboard",
            "cloudwatch:ListDashboards",
            "cloudwatch>DeleteDashboards",
            "cloudwatch:GetDashboard"
        ],
        "Resource": "*"
    }
]
}

```

ParallelClusterUserPolicy con awsbatch

En el ejemplo siguiente se establece ParallelClusterUserPolicy, utilizando awsbatch como programador. En este ejemplo, «*<RESOURCES S3 BUCKET>*» es el valor de la [cluster_resource_bucket](#) configuración; si no [cluster_resource_bucket](#) se especifica, entonces «*<RESOURCES S3 BUCKET>*» es «parallelcluster-*».

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeKeyPairs",
        "ec2:DescribeRegions",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribePlacementGroups",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeSnapshots",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeAddresses",

```

```
        "ec2:CreateTags",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Describe"
},
{
    "Action": [
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:ModifyLaunchTemplate",
        "ec2>DeleteLaunchTemplate",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2LaunchTemplate"
},
{
    "Action": [
        "ec2:CreateVpc",
        "ec2:ModifyVpcAttribute",
        "ec2:DescribeNatGateways",
        "ec2:CreateNatGateway",
        "ec2:DescribeInternetGateways",
        "ec2:CreateInternetGateway",
        "ec2:AttachInternetGateway",
        "ec2:DescribeRouteTables",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:AssociateRouteTable",
        "ec2:CreateSubnet",
        "ec2:ModifySubnetAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "NetworkingEasyConfig"
},
{
    "Action": [
        "ec2:CreateVolume",
```

```

        "ec2:RunInstances",
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateNetworkInterface",
        "ec2:CreateSecurityGroup",
        "ec2:ModifyVolumeAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2>DeleteNetworkInterface",
        "ec2>DeleteVolume",
        "ec2:TerminateInstances",
        "ec2>DeleteSecurityGroup",
        "ec2:DisassociateAddress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:ReleaseAddress",
        "ec2:CreatePlacementGroup",
        "ec2>DeletePlacementGroup"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Modify"
},
{
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:CreateTable",
        "dynamodb>DeleteTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Query",
        "dynamodb:TagResource"
    ],
    "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/
parallelcluster-*",
    "Effect": "Allow",
    "Sid": "DynamoDB"
},
{
    "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",

```

```

        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStacks",
        "cloudformation:GetTemplate",
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:UpdateStack"
    ],
    "Resource": "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*",
    "Effect": "Allow",
    "Sid": "CloudFormation"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:CreateHostedZone",
        "route53>DeleteHostedZone",
        "route53:GetChange",
        "route53:GetHostedZone",
        "route53:ListResourceRecordSets"
    ],
    "Resource": "arn:aws:route53:::hostedzone/*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
},
{
    "Action": [
        "sqs:GetQueueAttributes",
        "sqs:CreateQueue",
        "sqs:SetQueueAttributes",
        "sqs>DeleteQueue",
        "sqs:TagQueue"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SQS"
},
{
    "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility",

```

```

        "sqs:DeleteMessage",
        "sqs:GetQueueUrl"
    ],
    "Resource": "arn:aws:sqs:<REGION>:<AWS ACCOUNT ID>;parallelcluster-*",
    "Effect": "Allow",
    "Sid": "SQSQueue"
},
{
    "Action": [
        "sns:ListTopics",
        "sns:GetTopicAttributes",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Unsubscribe",
        "sns>DeleteTopic"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "SNS"
},
{
    "Action": [
        "iam:PassRole",
        "iam:CreateRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam:SimulatePrincipalPolicy"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>;role/parallelcluster-*",
        "arn:aws:iam::<AWS ACCOUNT ID>;role/<PARALLELCLUSTER EC2 ROLE NAME>"
    ],
    "Effect": "Allow",
    "Sid": "IAMRole"
},
{
    "Action": [
        "iam>CreateInstanceProfile",
        "iam>DeleteInstanceProfile",
        "iam:GetInstanceProfile",
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>;instance-profile/*",

```

```

    "Effect": "Allow",
    "Sid": "IAMInstanceProfile"
  },
  {
    "Action": [
      "iam:AddRoleToInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile",
      "iam:GetRolePolicy",
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy",
      "iam:GetPolicy",
      "iam:AttachRolePolicy",
      "iam:DetachRolePolicy"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "IAM"
  },
  {
    "Action": [
      "s3:*"
    ],
    "Resource": [
      "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
  },
  {
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": [
      "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
  },
  {
    "Action": [
      "s3:DeleteBucket",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion"
    ]
  }

```

```

    ],
    "Resource": [
        "arn:aws:s3:::<RESOURCES S3 BUCKET>"
    ],
    "Effect": "Allow",
    "Sid": "S3Delete"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda>DeleteFunction",
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:InvokeFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:TagResource",
        "lambda:ListTags",
        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
        "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "logs:*"
    ],
    "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:*",
    "Effect": "Allow",
    "Sid": "Logs"
},
{
    "Action": [
        "codebuild:*"
    ],
    "Resource": "arn:aws:codebuild:<REGION>:<AWS ACCOUNT ID>:project/parallelcluster-*",
    "Effect": "Allow",
    "Sid": "CodeBuild"
},

```

```
{
  "Action": [
    "ecr:*"
  ],
  "Resource": "*",
  "Effect": "Allow",
  "Sid": "ECR"
},
{
  "Action": [
    "batch:*"
  ],
  "Resource": "*",
  "Effect": "Allow",
  "Sid": "Batch"
},
{
  "Action": [
    "events:*"
  ],
  "Effect": "Allow",
  "Resource": "*",
  "Sid": "AmazonCloudWatchEvents"
},
{
  "Action": [
    "ecs:DescribeContainerInstances",
    "ecs:ListContainerInstances"
  ],
  "Resource": "*",
  "Effect": "Allow",
  "Sid": "ECS"
},
{
  "Action": [
    "elasticfilesystem:CreateFileSystem",
    "elasticfilesystem:CreateMountTarget",
    "elasticfilesystem>DeleteFileSystem",
    "elasticfilesystem>DeleteMountTarget",
    "elasticfilesystem:DescribeFileSystems",
    "elasticfilesystem:DescribeMountTargets"
  ],
  "Resource": "*",
  "Effect": "Allow",
```

```

        "Sid": "EFS"
    },
    {
        "Action": [
            "fsx:*"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "FSx"
    },
    {
        "Sid": "CloudWatch",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:PutDashboard",
            "cloudwatch:ListDashboards",
            "cloudwatch>DeleteDashboards",
            "cloudwatch:GetDashboard"
        ],
        "Resource": "*"
    }
}

```

ParallelClusterLambdaPolicy usando SGE, Slurm, o Torque

El siguiente ejemplo establece el `ParallelClusterLambdaPolicy`, mediante SGE, Slurm, o Torque como planificador.

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    }
  ]
}

```

```

    ],
    "Resource": "arn:aws:logs:*:*:*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogsPolicy"
  },
  {
    "Action": [
      "s3:DeleteBucket",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion",
      "s3:ListBucket",
      "s3:ListBucketVersions"
    ],
    "Resource": [
      "arn:aws:s3::*:*"
    ],
    "Effect": "Allow",
    "Sid": "S3BucketPolicy"
  },
  {
    "Action": [
      "ec2:DescribeInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "DescribeInstances"
  },
  {
    "Action": [
      "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "FleetTerminatePolicy"
  },
  {
    "Action": [
      "dynamodb:GetItem",
      "dynamodb:PutItem"
    ],
    "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
    "Effect": "Allow",
    "Sid": "DynamoDBTable"
  },

```

```

{
  "Action": [
    "route53:ListResourceRecordSets",
    "route53:ChangeResourceRecordSets"
  ],
  "Resource": [
    "arn:aws:route53::hostedzone/*"
  ],
  "Effect": "Allow",
  "Sid": "Route53DeletePolicy"
}
]
}

```

ParallelClusterLambdaPolicy con awsbatch

En el ejemplo siguiente se establece ParallelClusterLambdaPolicy, utilizando awsbatch como programador.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:*:*:*",
      "Sid": "CloudWatchLogsPolicy"
    },
    {
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:ListImages"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "ECRPolicy"
    },
    {
      "Action": [
        "codebuild:BatchGetBuilds",

```

```
    "codebuild:StartBuild"
  ],
  "Effect": "Allow",
  "Resource": "*",
  "Sid": "CodeBuildPolicy"
},
{
  "Action": [
    "s3:DeleteBucket",
    "s3:DeleteObject",
    "s3:DeleteObjectVersion",
    "s3:ListBucket",
    "s3:ListBucketVersions"
  ],
  "Effect": "Allow",
  "Resource": "*",
  "Sid": "S3BucketPolicy"
}
]
```

ParallelClusterUserPolicy para usuarios

El siguiente ejemplo establece la `ParallelClusterUserPolicy` para los usuarios que no necesitan crear o actualizar clústeres. Se admiten los siguientes comandos:

- [pcluster dcv](#)
- [pcluster instances](#)
- [pcluster list](#)
- [pcluster ssh](#)
- [pcluster start](#)
- [pcluster status](#)
- [pcluster stop](#)
- [pcluster version](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "MinimumModify",
    "Action": [
      "autoscaling:UpdateAutoScalingGroup",
      "batch:UpdateComputeEnvironment",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResources",
      "cloudformation:GetTemplate",
      "dynamodb:GetItem",
      "dynamodb:PutItem"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:autoscaling:<REGION>:<AWS ACCOUNT ID>:autoScalingGroup/*:autoScalingGroupName/parallelcluster-*",
      "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:compute-environment/*",
      "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/<CLUSTERNAME>/
*",
      "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/<CLUSTERNAME>"
    ]
  },
  {
    "Sid": "Describe",
    "Action": [
      "cloudformation:DescribeStacks",
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
]
}

```

Planificadores compatibles con AWS ParallelCluster

AWS ParallelCluster admite varios programadores, configurados mediante la configuración.

[scheduler](#)

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores. Puede seguir utilizándolos en las versiones anteriores a la 2.11.4 (inclusive),

pero no son aptas para recibir actualizaciones futuras ni asistencia para la solución de problemas por parte de los equipos de AWS servicio y AWS soporte.

Temas

- [Son of Grid Engine \(sge\)](#)
- [Slurm Workload Manager \(slurm\)](#)
- [Torque Resource Manager \(torque\)](#)
- [AWS Batch \(awsbatch\)](#)

Son of Grid Engine (**sge**)

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores. Puede seguir utilizándolos en las versiones anteriores a la 2.11.4 (inclusive), pero no son aptas para recibir actualizaciones futuras ni asistencia para la solución de problemas por parte de los equipos de AWS servicio y AWS soporte.

AWS ParallelCluster se utilizan las versiones 2.11.4 y anteriores Son of Grid Engine 8.1.9.

Slurm Workload Manager (**slurm**)

AWS ParallelCluster la versión 2.11.9 usa Slurm 20.11.9. Para obtener más información Slurm, consulte <https://slurm.schedmd.com/>. Para obtener descargas, consulte <https://github.com/SchedMD/slurm/tags>. Para obtener el código fuente, consulte <https://github.com/SchedMD/slurm>.

Important

AWS ParallelCluster se prueba con Slurm parámetros de configuración, que se proporcionan de forma predeterminada. Cualquier cambio que realice en estos Slurm Los parámetros de configuración se realizan bajo su propia responsabilidad. Solo se admiten en la medida de lo posible.

| AWS ParallelCluster versión (s) | Compatible Slurm versión |
|---------------------------------|--------------------------|
| 2.11.7, 2.11.8, 2.11.9 | 20,11,9 |
| De 2.11.4 a 2.11.6 | 20,11,8 |
| De 2.11.0 a 2.11.3 | 20,11,7 |
| 2.10.4 | 20,02.7 |
| De 2.9.0 a 2.10.3 | 20,02.4 |
| De 2.6 a 2.8.1 | 19,05.5 |
| 2.5.0, 2.5.1 | 19,05,3-2 |
| De 2.3.1 a 2.4.1 | 18,08,6-2 |
| Antes de la versión 2.3.1 | 16,05,3-1 |

Modo de Cola múltiple

AWS ParallelCluster la versión 2.9.0 introdujo el modo de cola múltiple. Se admite el modo de cola múltiple cuando [scheduler](#) se establece en `slurm` y se define la [queue_settings](#) configuración. Este modo permite que coexistan diferentes tipos de instancias en los nodos de cómputo. Los recursos informáticos que contienen los distintos tipos de instancias pueden escalarse o reducirse verticalmente según sea necesario. [En el modo de cola, se admiten hasta cinco \(5\) colas y cada \[queue\]sección puede hacer referencia a un máximo de tres \(3\) \[compute_resource\] secciones.](#) Cada una de estas [\[queue\]secciones](#) es una partición en Slurm Workload Manager. Para obtener más información, consulte [Slurm guía para el modo de cola múltiple](#) y [Tutorial sobre el modo de cola múltiple](#).

Cada [\[compute_resource\]sección](#) de una cola debe tener un tipo de instancia diferente y cada una de ellas `[compute_resource]` se divide a su vez en nodos estáticos y dinámicos. Los nodos estáticos de cada uno `[compute_resource]` se numeran del 1 al valor de [min_count](#). Los nodos dinámicos de cada uno `[compute_resource]` se numeran del uno (1) al [\(max_count-min_count\)](#). Por ejemplo, si `min_count` es 2 y `max_count` es 10, los nodos dinámicos `[compute_resource]` se numeran del uno (1) al ocho (8). En cualquier momento, puede haber entre cero (0) y el número máximo de nodos dinámicos en a `[compute_resource]`.

Las instancias que se lanzan a la flota de computación se asignan de forma dinámica. Para ayudar a gestionar esto, se generan nombres de host para cada nodo. El formato del nombre de host es el siguiente:

```
$HOSTNAME=$QUEUE-$STATDYN-$INSTANCE_TYPE-$NODENUM
```

- \$QUEUE es el nombre de la cola. Por ejemplo, si la sección comienza [queue *queue-name*], entonces «\$QUEUE» es «*queue-name*».
- \$STATDYN es st para nodos estáticos o dy para dinámicos.
- \$INSTANCE_TYPE es el tipo de instancia para [compute_resource], de la [instance_type](#) configuración.
- \$NODENUM es el número del nodo. \$NODENUM está entre uno (1) y el valor de [min_count](#) para los nodos estáticos y entre uno (1) y [max_count](#)-[min_count](#) para los dinámicos.

Tanto los nombres de host como los nombres de dominio completos (FQDN) se crean mediante las zonas alojadas de Amazon Route 53. El FQDN es \$HOSTNAME.\$CLUSTERNAME.pcluster, donde \$CLUSTERNAME es el nombre de la [\[cluster\]sección](#) utilizada para el clúster.

Para convertir la configuración a un modo de cola, utilice el [pcluster-config convert](#) comando. Escribe una configuración actualizada con una sola [\[queue\]sección](#) denominada [queue compute]. Esa cola contiene una sola [\[compute_resource\]sección](#) que recibe el nombre [compute_resource default]. El [queue compute] y [compute_resource default] tiene la configuración migrada desde la [\[cluster\]sección](#) especificada.

Slurm guía para el modo de cola múltiple

AWS ParallelCluster la versión 2.9.0 introdujo el modo de cola múltiple y una nueva arquitectura de escalado para Slurm Workload Manager (Slurm).

En las siguientes secciones se proporciona una visión general sobre el uso de un Slurm agrupar con la arquitectura de escalado recientemente introducida.

Información general

La nueva arquitectura de escalado se basa en Slurm de la [guía de programación en la nube](#) y el complemento de ahorro de energía. Para obtener más información sobre el complemento de ahorro de energía, consulta [Slurm Guía de ahorro de energía](#). En la nueva arquitectura, los recursos que podrían estar disponibles para un clúster suelen estar predefinidos en la Slurm configuración como nodos de nube.

Ciclo de vida de los nodos de la nube

A lo largo de su ciclo de vida, los nodos de la nube entran en varios de los siguientes estados (o en todos): POWER_SAVING, POWER_UP (pow_up), ALLOCATED (alloc) y POWER_DOWN (pow_dn). En algunos casos, un nodo de la nube puede entrar en el estado OFFLINE. La siguiente lista detalla varios aspectos de estos estados en el ciclo de vida de los nodos de la nube.

- Un nodo en un estado POWER_SAVING aparece con un sufijo ~ (por ejemplo idle~) en `sinfo`. En este estado, no hay ninguna EC2 instancia que respalde al nodo. Sin embargo, Slurm aún puede asignar trabajos al nodo.
- Un nodo en transición a un estado POWER_UP aparece con un sufijo # (por ejemplo idle#) en `sinfo`.
- Cuando Slurm asigna la tarea a un nodo en un POWER_SAVING estado, el nodo la transfiere automáticamente a un POWER_UP estado. De lo contrario, los nodos se pueden colocar en el POWER_UP estado manualmente mediante el `scontrol update nodename=nodename state=pow_up` comando. En esta etapa, `ResumeProgram` se invoca y las EC2 instancias se lanzan y configuran para respaldar un POWER_UP nodo.
- Un nodo que está actualmente disponible para su uso aparece sin sufijo (por ejemplo) en `sinfo`. Una vez que el nodo se haya configurado y se haya unido al clúster, estará disponible para ejecutar trabajos. En esta etapa, el nodo está correctamente configurado y listo para su uso. Como regla general, recomendamos que el número de instancias EC2 sea el mismo que el número de nodos disponibles. En la mayoría de los casos, los nodos estáticos están disponibles una vez creado el clúster.
- Un nodo que está en transición a un estado POWER_DOWN aparece con un sufijo % (por ejemplo idle%) en `sinfo`. Los nodos dinámicos entran automáticamente en el estado POWER_DOWN después de [scaledown_idletime](#). Por el contrario, los nodos estáticos no están apagados en la mayoría de los casos. Sin embargo, los nodos se pueden colocar en el POWER_DOWN estado manualmente mediante el `scontrol update nodename=nodename state=powering_down` comando. En este estado, se finalizan las instancias asociadas a un nodo, y el nodo vuelve a ese estado POWER_SAVING y está disponible para su uso después de [scaledown_idletime](#). La `scaledown-idletime` configuración se guarda en Slurm configuración como `SuspendTimeout` parámetro.
- Un nodo que esté desconectado aparece con un sufijo * (por ejemplo down*) en `sinfo`. Un nodo se desconecta si Slurm el controlador no puede contactar con el nodo o si los nodos estáticos están deshabilitados y las instancias de respaldo están terminadas.

Observe los estados de los nodos que se muestran en el siguiente ejemplo de `sinfo`.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4    idle~ efa-dy-c5n18xlarge-[1-4]
efa        up    infinite   1    idle  efa-st-c5n18xlarge-1
gpu        up    infinite   1    idle% gpu-dy-g38xlarge-1
gpu        up    infinite   9    idle~ gpu-dy-g38xlarge-[2-10]
ondemand  up    infinite   2    mix#  ondemand-dy-c52xlarge-[1-2]
ondemand  up    infinite  18    idle~ ondemand-dy-c52xlarge-[3-10],ondemand-dy-
t2xlarge-[1-10]
spot*     up    infinite  13    idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2large-[1-3]
spot*     up    infinite   2    idle  spot-st-t2large-[1-2]
```

Los nodos `spot-st-t2large-[1-2]` y `efa-st-c5n18xlarge-1` ya tienen instancias de respaldo configuradas y están disponibles para su uso. Los nodos `ondemand-dy-c52xlarge-[1-2]` se encuentran en el estado `POWER_UP` y deberían estar disponibles en unos minutos. El nodo `gpu-dy-g38xlarge-1` se encuentra en el estado `POWER_DOWN` y pasa al estado `POWER_SAVING` después de [scaledown_idletime](#) (el valor predeterminado es 10 minutos).

Todos los demás nodos están en `POWER_SAVING` estado y no hay EC2 instancias que los respalden.

Trabajo con un nodo disponible

Un nodo disponible está respaldado por una EC2 instancia. De forma predeterminada, el nombre del nodo se puede usar para SSH entrar directamente en la instancia (por ejemplo `ssh efa-st-c5n18xlarge-1`). La dirección IP privada de la instancia se puede recuperar usando el comando `scontrol show nodes nodename` y comprobando el campo `NodeAddr`: En el caso de los nodos que no están disponibles, el `NodeAddr` campo no debe apuntar a una EC2 instancia en ejecución. Más bien, debe ser el mismo que el nombre del nodo.

Estados y envío de los trabajos

En la mayoría de los casos, los trabajos enviados se asignan inmediatamente a los nodos del sistema o se dejan pendientes si se asignan todos los nodos.

Si los nodos asignados a un trabajo incluyen algún nodo en un estado `POWER_SAVING`, el trabajo comienza con un estado `CF` o `CONFIGURING`. En este momento, el trabajo espera a que los nodos del estado `POWER_SAVING` pasen al estado `POWER_UP` y estén disponibles.

Una vez que todos los nodos asignados a un trabajo estén disponibles, el trabajo pasa al estado `RUNNING (R)`.

De forma predeterminada, todos los trabajos se envían a la cola predeterminada (conocida como partición) en Slurm). Esto se indica con un * sufijo después del nombre de la cola. Puede seleccionar una cola mediante la opción de envío de trabajos -p.

Todos los nodos están configurados con las siguientes características, que se pueden utilizar en los comandos de envío de trabajos:

- Un tipo de instancia (por ejemplo c5.xlarge)
- Un tipo de nodo (puede ser `dynamic` o `static`)

Puede ver todas las funciones disponibles para un nodo concreto utilizando el `scontrol show nodes nodename` comando y consultando la `AvailableFeatures` lista.

Otra consideración son los puestos de trabajo. Tenga en cuenta el estado inicial del clúster, que puede ver ejecutando el comando `sinfo`.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4     idle~ efa-dy-c5n18xlarge-[1-4]
efa        up    infinite   1     idle  efa-st-c5n18xlarge-1
gpu        up    infinite  10     idle~ gpu-dy-g38xlarge-[1-10]
ondemand  up    infinite  20     idle~ ondemand-dy-c52xlarge-[1-10],ondemand-dy-
t2xlarge-[1-10]
spot*      up    infinite  13     idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2large-[1-3]
spot*      up    infinite   2     idle  spot-st-t2large-[1-2]
```

Tenga en cuenta que la cola predeterminada es `spot`. Se indica mediante el sufijo `*`.

Envíe un trabajo a un nodo estático de la cola predeterminada (`spot`).

```
$ sbatch --wrap "sleep 300" -N 1 -C static
```

Envíe un trabajo a un nodo dinámico de la cola EFA.

```
$ sbatch --wrap "sleep 300" -p efa -C dynamic
```

Envíe un trabajo a ocho (8) nodos `c5.2xlarge` y dos (2) nodos `t2.xlarge` de la cola `ondemand`.

```
$ sbatch --wrap "sleep 300" -p ondemand -N 10 -C "[c5.2xlarge*8&t2.xlarge*2]"
```

Envíe un trabajo a un GPU nodo de la cola. gpu

```
$ sbatch --wrap "sleep 300" -p gpu -G 1
```

Tenga en cuenta el estado de los trabajos mediante el comando squeue.

```
$ squeue
      JOBID PARTITION    NAME    USER ST       TIME  NODES NODELIST(REASON)
      12  ondemand    wrap    ubuntu CF        0:36    10  ondemand-dy-
c52xlarge-[1-8],ondemand-dy-t2xlarge-[1-2]
      13      gpu    wrap    ubuntu CF        0:05     1  gpu-dy-g38xlarge-1
      7      spot    wrap    ubuntu R        2:48     1  spot-st-t2large-1
      8      efa    wrap    ubuntu R        0:39     1  efa-dy-
c5n18xlarge-1
```

Los trabajos 7 y 8 (en las colas spot y efa) ya se están ejecutando (R). Los trabajos 12 y 13 aún se están configurando (CF), probablemente esperando a que las instancias estén disponibles.

```
# Nodes states corresponds to state of running jobs
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa       up    infinite   3  idle~ efa-dy-c5n18xlarge-[2-4]
efa       up    infinite   1  mix  efa-dy-c5n18xlarge-1
efa       up    infinite   1  idle  efa-st-c5n18xlarge-1
gpu       up    infinite   1  mix~  gpu-dy-g38xlarge-1
gpu       up    infinite   9  idle~  gpu-dy-g38xlarge-[2-10]
ondemand  up    infinite  10  mix#  ondemand-dy-c52xlarge-[1-8],ondemand-dy-
t2xlarge-[1-2]
ondemand  up    infinite  10  idle~  ondemand-dy-c52xlarge-[9-10],ondemand-dy-
t2xlarge-[3-10]
spot*     up    infinite  13  idle~  spot-dy-c5xlarge-[1-10],spot-dy-t2large-[1-3]
spot*     up    infinite   1  mix  spot-st-t2large-1
spot*     up    infinite   1  idle  spot-st-t2large-2
```

Estado y características de los nodos

En la mayoría de los casos, los estados de los nodos se administran completamente de AWS ParallelCluster acuerdo con los procesos específicos del ciclo de vida de los nodos de la nube descritos anteriormente en este tema.

Sin embargo, AWS ParallelCluster también reemplaza o termina los nodos en mal estado y los DRAINED estados DOWN y nodos que tienen instancias de respaldo en mal estado. Para obtener más información, consulte [clustermgtd](#).

Estados de partición

AWS ParallelCluster admite los siguientes estados de partición. A Slurm la partición es una cola en AWS ParallelCluster.

- UP: indica que la partición se encuentra en estado activo. Es el valor predeterminado de una partición. En este estado, todos los nodos de la partición están activos y disponibles para su uso.
- INACTIVE: indica que la partición se encuentra en estado inactivo. En este estado, se finalizan todas las instancias que respaldan a los nodos de una partición inactiva. No se lanzan nuevas instancias para los nodos de una partición inactiva.

pcluster: iniciar y detener

Cuando [pcluster stop](#) se ejecuta, todas las particiones se colocan en ese INACTIVE estado y los AWS ParallelCluster procesos mantienen las particiones en ese INACTIVE estado.

Cuando [pcluster start](#) se ejecuta, todas las particiones se colocan inicialmente en el UP estado. Sin embargo, AWS ParallelCluster los procesos no mantienen la partición en un UP estado. Debe cambiar los estados de las particiones manualmente. Todos los nodos estáticos están disponibles al cabo de unos minutos. Tenga en cuenta que establecer una partición en UP no activa ninguna capacidad dinámica. Si [initial_count](#) es mayor que [max_count](#), es [initial_count](#) posible que no se satisfaga cuando el estado de la partición cambie al UP estado.

Cuando se ejecuta [pcluster start](#) y [pcluster stop](#), puede consultar el estado del clúster ejecutando el comando [pcluster status](#) y comprobando el ComputeFleetStatus. A continuación, se indican los estados posibles:

- STOP_REQUESTED: La [pcluster stop](#) solicitud se envía al clúster.
- STOPPING: el proceso de `pcluster` está iniciando actualmente el clúster.
- STOPPED: el proceso de `pcluster` ha finalizado el proceso de detención, todas las particiones están en estado INACTIVE y todas las instancias de procesamiento han finalizado.
- START_REQUESTED: La [pcluster start](#) solicitud se envía al clúster.
- STARTING: el proceso de `pcluster` está iniciando actualmente el clúster.

- **RUNNING:** el proceso de `pcluster` ha finalizado el proceso de inicio, todas las particiones están en estado UP y los nodos estáticos están disponibles después de unos minutos.

Control manual de las colas

En algunos casos, es posible que desee tener cierto control manual sobre los nodos o la cola (conocida como partición) en Slurm) en un clúster. Puede administrar los nodos de un clúster mediante los siguientes procedimientos comunes usando el comando .

- Encienda los nodos dinámicos en `POWER_SAVING` estado: ejecute el `scontrol update nodename=nodename state=power_up` comando o envíe un marcador de `sleep 1` posición solicitando un número determinado de nodos y confíe en Slurm para encender la cantidad requerida de nodos.
- Apague los nodos dinámicos antes `scaledown_idletime`: configure los nodos dinámicos en `DOWN` con el `scontrol update nodename=nodename state=down` comando. AWS ParallelCluster termina y restablece automáticamente los nodos dinámicos desactivados. En general, no es recomendable establecer nodos en `POWER_DOWN` directamente con el comando `scontrol update nodename=nodename state=power_down`. Esto se debe a que AWS ParallelCluster administra automáticamente el proceso de apagado. No es necesario realizar una intervención manual. Por lo tanto, se recomienda intentar configurar nodos en `DOWN`
- Desactive una cola (partición) o detenga todos los nodos estáticos de una partición específica: defina la cola en un valor específico `INACTIVE` con el `scontrol update partition=queue name state=inactive` comando. De este modo, se finalizan todas las instancias que respaldan a los nodos de la partición.
- Habilitar una cola (partición): defina una cola específica `INACTIVE` con el comando. `scontrol update partition=queue name state=up`

Comportamiento y ajustes del escalado

A continuación, se muestra un ejemplo del flujo de trabajo del escalado normal:

- El programador recibe un trabajo que requiere dos nodos.
- El programador pasa dos nodos a un estado `POWER_UP` y llama a `ResumeProgram` con los nombres de los nodos (por ejemplo `queue1-dy-c5xlarge-[1-2]`).

- ResumeProgram lanza dos EC2 instancias y asigna las direcciones IP privadas y los nombres de host de los nodos `queue1-dy-c5xlarge-[1-2]`, esperando ResumeTimeout (el período predeterminado es de 60 minutos (1 hora)) antes de restablecer los nodos.
- Las instancias se configuran y se unen al clúster. Un trabajo comienza a ejecutarse en las instancias.
- Job está hecho.
- Una vez transcurrido el SuspendTime configurado (que está establecido en [scaledown_idletime](#)), el programador establece las instancias en el estado POWER_SAVING. El programador lo coloca `queue1-dy-c5xlarge-[1-2]` en POWER_DOWN estado y llama SuspendProgram con los nombres de los nodos.
- Se llama a SuspendProgram para dos nodos. Los nodos permanecen en el estado POWER_DOWN, por ejemplo, permaneciendo idle% durante un SuspendTimeout (el periodo predeterminado es de 120 segundos, es decir, 2 minutos). Cuando clustermgtd detecta que los nodos se están apagando, finaliza las instancias de respaldo. Luego, se configura `queue1-dy-c5xlarge-[1-2]` en estado inactivo y restablece la dirección IP privada y el nombre de host para que puedan volver a encenderse para futuros trabajos.

Si algo sale mal y no se puede lanzar una instancia para un nodo concreto por algún motivo, ocurre lo siguiente:

- El programador recibe un trabajo que requiere dos nodos.
- El programador pasa dos nodos de ampliación en la nube al estado POWER_UP y llama a ResumeProgram con los nombres de los nodos (por ejemplo `queue1-dy-c5xlarge-[1-2]`).
- ResumeProgram lanza solo una (1) EC2 instancia y la configura `queue1-dy-c5xlarge-1`, pero no pudo lanzar una instancia para `queue1-dy-c5xlarge-2`
- `queue1-dy-c5xlarge-1` no se verá afectada y se pondrá en línea cuando alcance POWER_UP el estado.
- `queue1-dy-c5xlarge-2` se coloca en POWER_DOWN estado y el trabajo se vuelve a poner en cola automáticamente porque Slurm detecta un fallo en el nodo.
- `queue1-dy-c5xlarge-2` pasa a estar disponible después de SuspendTimeout (el valor predeterminado es 120 segundos, es decir, 2 minutos). Mientras tanto, el trabajo se vuelve a poner en cola y puede empezar a ejecutarse en otro nodo.
- El proceso anterior se repite hasta que el trabajo se pueda ejecutar en un nodo disponible sin que se produzca ningún error.

Hay dos parámetros de temporización que se pueden ajustar si es necesario:

- `ResumeTimeout`(el valor predeterminado es 60 minutos (1 hora)): `ResumeTimeout` controla la hora Slurm espera antes de poner el nodo en estado inactivo.
 - Podría ser útil ampliar el si el proceso previo o posterior a la instalación tiene una duración similar.
 - Este es también el tiempo máximo de AWS ParallelCluster espera antes de reemplazar o restablecer un nodo si hay algún problema. Los nodos de computación se autofinalizan si se produce algún error durante el inicio o la configuración. A continuación, el AWS ParallelCluster proceso también reemplaza al nodo cuando ve que la instancia ha terminado.
- `SuspendTimeout` (el valor predeterminado es 120 segundos, es decir, 2 minutos): controla la rapidez con la que los nodos se vuelven a colocar en el sistema y están listos para volver a usarse.
 - Cuanto más corto `SuspendTimeout` sea, los nodos se restablecerán más rápidamente, y Slurm puede intentar lanzar instancias con más frecuencia.
 - Un valor más alto de `SuspendTimeout` significa que los nodos que han fallado se restablecen más lento. Mientras tanto, Slurm neumáticos para usar otros nodos. Si `SuspendTimeout` son más de unos pocos minutos, Slurm intenta recorrer todos los nodos del sistema. Un tiempo más prolongado `SuspendTimeout` podría ser beneficioso para los sistemas a gran escala (más de 1000 nodos) para reducir el stress en Slurm volviendo a poner en cola con frecuencia los trabajos que no funcionan.
 - Tenga en cuenta que `SuspendTimeout` esto no se refiere al tiempo de AWS ParallelCluster espera para finalizar una instancia de respaldo para un nodo. Las instancias de respaldo de los nodos `power down` finalizan inmediatamente. El proceso de finalización por lo general se completa en unos minutos. Sin embargo, durante este tiempo, el nodo permanece en el estado y no está disponible para que lo utilice el programador.

Registros para la arquitectura

La siguiente lista contiene los registros clave de la arquitectura de colas múltiples. El nombre del flujo de registro utilizado con Amazon CloudWatch Logs tiene el formato `{hostname}.{instance_id}.{logIdentifier}` siguiente: `logIdentifier` sigue los nombres de los registros. Para obtener más información, consulte [Integración con Amazon CloudWatch Logs](#).

- `ResumeProgram`:

`/var/log/parallelcluster/slurm_resume.log (slurm_resume)`

- SuspendProgram:

`/var/log/parallelcluster/slurm_suspend.log (slurm_suspend)`

- clustermgtd:

`/var/log/parallelcluster/clustermgtd.log (clustermgtd)`

- computemgtd:

`/var/log/parallelcluster/computemgtd.log (computemgtd)`

- slurmctld:

`/var/log/slurmctld.log (slurmctld)`

- slurmd:

`/var/log/slurmd.log (slurmd)`

Problemas frecuentes y cómo depurarlos:

Nodos que no se iniciaron o encendieron o que no se unieron al clúster

- Nodos dinámicos:
 - Compruebe el registro `ResumeProgram` para ver si se ha llamado a `ResumeProgram` con el nodo. Si no es así, compruebe el `slurmctld` registro para determinar si Slurm alguna vez intentó llamar `ResumeProgram` con el nodo. Tenga en cuenta que los permisos incorrectos activados en `ResumeProgram` pueden provocar un error silencioso.
 - Si se llama a `ResumeProgram`, compruebe si se ha lanzado una instancia para el nodo. Si la instancia no se ha lanzado, debería mostrarse un mensaje de error claro que explique por qué no se ha podido lanzar la instancia.
 - Si se ha lanzado una instancia, es posible que se haya producido algún problema durante el proceso de arranque. Busque la dirección IP privada y el ID de instancia correspondientes en el `ResumeProgram` registro y consulte los registros de arranque correspondientes a la instancia específica en CloudWatch los registros.
- Nodos estáticos:

- Compruebe el registro `clustermgtd` para ver si se han lanzado instancias para el nodo. Si no se han lanzado instancias, deberían mostrarse mensajes de error claros que expliquen por qué no se han podido lanzar las instancias.
- Si se ha lanzado una instancia, se ha producido algún problema en el proceso de arranque. Busca la IP privada y el ID de instancia correspondientes en el `clustermgtd` registro y busca los registros de arranque correspondientes a la instancia específica en CloudWatch Logs.

Los nodos se han sustituido o finalizado de forma inesperada y han fallado

- Nodos sustituidos o finalizados de forma inesperada:
 - En la mayoría de los casos, `clustermgtd` administra todas las acciones de mantenimiento de los nodos. Para comprobar si `clustermgtd` ha sustituido o finalizado un nodo, compruebe el registro `clustermgtd`.
 - Si `clustermgtd` sustituye o finaliza el nodo, debería mostrarse un mensaje que indique el motivo de la acción. Si el motivo está relacionado con el programador (por ejemplo, el nodo estaba DOWN), consulte el registro `slurmctld` para obtener más información. Si el motivo está EC2 relacionado, usa herramientas para comprobar el estado o los registros de esa instancia. Por ejemplo, puedes comprobar si la instancia tenía eventos programados o no pasó las comprobaciones de EC2 estado.
 - Si `clustermgtd` no ha terminado el nodo, compruebe si `computemgtd` ha terminado el nodo o si EC2 ha terminado la instancia para recuperar una instancia puntual.
- Fallos de nodo:
 - En la mayoría de los casos, los trabajos se vuelven a poner en cola automáticamente si se produce un error en un nodo. Consulte el registro `slurmctld` para ver por qué ha fallado un trabajo o un nodo y evalúe la situación a partir de ahí.

Fallo al sustituir o finalizar instancias, error al apagar los nodos

- En general, `clustermgtd` administra todas las acciones de finalización de instancias esperadas. Consulte el registro `clustermgtd` para ver por qué no se ha podido sustituir o finalizar un nodo.
- En el caso de los nodos dinámicos que no superan el [scaledown_idletime](#), consulte el registro de `SuspendProgram` para ver si los procesos de `slurmctld` realizaron llamadas con el nodo específico como argumento. Tenga en cuenta que `SuspendProgram` no realiza ninguna acción específica. Más bien, solo se encarga de registrar cuando se le llama. La finalización y el

NodeAdd: restablecimiento de todas las instancias se completan antes de `clustermgtd`. Slurm coloca los nodos IDLE después `SuspendTimeout`.

Otros problemas.

- AWS ParallelCluster no toma decisiones de asignación de puestos o escalamiento. Simplemente intenta lanzar, finalizar y mantener los recursos de acuerdo con Slurmsus instrucciones.

Si tiene problemas relacionados con la asignación de trabajos, la asignación de nodos y la decisión de escalado, consulte el registro `slurmctld` para ver si hay errores.

Torque Resource Manager (**torque**)

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores. Puede seguir utilizándolos en las versiones anteriores a la 2.11.4 (inclusive), pero no son aptas para recibir actualizaciones futuras ni asistencia para la solución de problemas por parte de los equipos de AWS servicio y AWS soporte.

AWS ParallelCluster se utilizan las versiones 2.11.4 y anteriores Torque Resource Manager 6.1.2. Para obtener más información acerca de Torque Resource Manager 6.1.2, véase <http://docs.adaptivecomputing.com/torque/6-1-2/releaseNotes/torquerelnote.htm> Para obtener la documentación, consulte <http://docs.adaptivecomputing.com/torque/6-1-2/adminGuide/torque.htm>. Para obtener el código fuente, consulte <https://github.com/adaptivecomputing/torque/tree/6.1.2>.

AWS ParallelCluster versiones 2.4.0 y versiones anteriores Torque Resource Manager 6.0.2. Para ver las notas de la versión, consulte <http://docs.adaptivecomputing.com/torque/6-0-2/releaseNotes/torqueReleaseNotes6.0.2.pdf>. Para obtener la documentación, consulte <http://docs.adaptivecomputing.com/torque/6-0-2/adminGuide/help.htm>. Para obtener el código fuente, consulte <https://github.com/adaptivecomputing/torque/tree/6.0.2>.

AWS Batch (**awsbatch**)

Para obtener información al respecto AWS Batch, consulte [AWS Batch](#). Para ver la documentación, consulte la [Guía del usuario de AWS Batch](#).

AWS ParallelCluster CLI comandos para AWS Batch

Al utilizar el `awsbatch` planificador, los AWS ParallelCluster CLI comandos de AWS Batch se instalan automáticamente en el nodo AWS ParallelCluster principal. CLI utiliza AWS Batch API operaciones y permite las siguientes operaciones:

- Enviar y administrar trabajos
- Monitorear trabajos, colas y hosts
- Crear un reflejo de los comandos del programador tradicionales

Important

AWS ParallelCluster no admite GPU trabajos para AWS Batch. Para obtener más información, consulte [GPUempleos](#).

Temas

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)
- [awsbqueues](#)
- [awsbhosts](#)

awsbsub

Envía trabajos a la cola de trabajos del clúster.

```
awsbsub [-h] [-jn JOB_NAME] [-c CLUSTER] [-cf] [-w WORKING_DIR]  
        [-pw PARENT_WORKING_DIR] [-if INPUT_FILE] [-p VCPUS] [-m MEMORY]  
        [-e ENV] [-eb ENV_DENYLIST] [-r RETRY_ATTEMPTS] [-t TIMEOUT]  
        [-n NODES] [-a ARRAY_SIZE] [-d DEPENDS_ON]  
        [command] [arguments [arguments ...]]
```

⚠ Important

AWS ParallelCluster no admite GPU puestos de trabajo para AWS Batch. Para obtener más información, consulte [GPUempleos](#).

Argumentos de posición

command

Envía el trabajo (el comando especificado debe estar disponible en las instancias de computación) o el nombre de archivo que se va a transferir. Véase también `--command-file`.

arguments

(Opcional) Especifica argumentos para el comando o el archivo de comandos.

Argumentos con nombre

-jn *JOB_NAME*, --job-name *JOB_NAME*

Asigna un nombre al trabajo. El primer carácter debe ser una letra o un número. El nombre del trabajo puede contener letras (mayúsculas y minúsculas), números, guiones (-) y guiones bajos (_), y puede tener una longitud máxima de 128 caracteres.

-c *CLUSTER*, --cluster *CLUSTER*

Especifica el clúster que se va a utilizar.

-cf, --command-file

Indica que el comando es un archivo que se va a transferir a las instancias de informática.

Valor predeterminado: False

-w *WORKING_DIR*, --working-dir *WORKING_DIR*

Especifica la carpeta que se va a usar como directorio de trabajo del trabajo. Si no se especifica un directorio de trabajo, el trabajo se ejecuta en la subcarpeta `job-<AWS_BATCH_JOB_ID>` del directorio de inicio del usuario. Puede utilizar este parámetro o el parámetro `--parent-working-dir`.

-pw *PARENT_WORKING_DIR*, --parent-working-dir *PARENT_WORKING_DIR*

Especifica la carpeta principal del directorio de trabajo del trabajo. Si no se especifica un directorio de trabajo principal, el valor predeterminado es el directorio de inicio del usuario. Una subcarpeta llamada `job-<AWS_BATCH_JOB_ID>` se crea en el directorio de trabajo principal. Puede utilizar este parámetro o el parámetro `--working-dir`.

-if *INPUT_FILE*, --input-file *INPUT_FILE*

Especifica el archivo que se va a transferir a las instancias de computación, en el directorio de trabajo de la tarea. Puede especificar varios parámetros de archivo de entrada.

-p *VCPUS*, --vcpus *VCPUS*

Especifica el número de vCPUs que se va a reservar para el contenedor. Cuando se usa junto con `-nodes`, identifica el número de vCPUs para cada nodo.

Valor predeterminado: 1

-m *MEMORY*, --memory *MEMORY*

Especifica el límite máximo de memoria (en MiB) que se va a proporcionar para el trabajo. Si su trabajo intenta superar el límite de memoria especificado aquí, se finaliza el trabajo.

Valor predeterminado: 128

-e *ENV*, --env *ENV*

Especifica una lista de nombres de variable de entorno separados por comas que se van a exportar al entorno de trabajo. Para exportar todas las variables de entorno, especifique "todas". Tenga en cuenta que una lista de "todas" las variables de entorno no incluyen aquellas que figuren en el parámetro `-env-blacklist` ni tampoco variables que empiecen por el prefijo `PCLUSTER_*` o `AWS_*`.

-eb *ENV_DENYLIST*, --env-blacklist *ENV_DENYLIST*

Especifica una lista de nombres de variable de entorno separados por comas que no se van a exportar al entorno de trabajo. De forma predeterminada, no se exportan `HOME`, `PWD`, `USER`, `PATH`, `LD_LIBRARY_PATH`, `TERM` ni `TERMCAP`.

-r *RETRY_ATTEMPTS*, --retry-attempts *RETRY_ATTEMPTS*

Especifica la cantidad de veces que toma pasar un trabajo al estado `RUNNABLE`. Puede especificar entre 1 y 10 intentos. Si el valor de los intentos es superior a 1, el trabajo se reintenta

si produce un error, hasta que haya pasado al estado `RUNNABLE`, que especificaba la cantidad de veces.

Valor predeterminado: 1

-t *TIMEOUT*, --timeout *TIMEOUT*

Especifica el tiempo en segundos (medido a partir de la `startedAt` marca de tiempo del intento de trabajo) tras el cual AWS Batch finaliza el trabajo si no lo ha hecho. El valor del tiempo de inactividad debe ser al menos 60 segundos.

-n *NODES*, --nodes *NODES*

Especifica el número de nodos que quiere reservar para el trabajo. Especifique un valor para que este parámetro habilite el envío paralelo de varios nodos.

 Note

Cuando el parámetro `cluster_type` está establecido en `parallel`, no se admiten los trabajos paralelos de varios nodos.

-a *ARRAY_SIZE*, --array-size *ARRAY_SIZE*

Indica el tamaño de la matriz. Puede especificar un valor comprendido entre 2 y 10 000. Si especifica las propiedades de matriz para un trabajo, pasa a ser un trabajo de matriz.

-d *DEPENDS_ON*, --depends-on *DEPENDS_ON*

Especifica una lista de dependencias separadas por puntos y coma para un trabajo. Un trabajo puede depender de un máximo de 20 trabajos. Puede especificar una dependencia de tipo `SEQUENTIAL` sin especificar un ID de trabajo para los trabajos de matriz. Una dependencia `SEQUENTIAL` permite que cada trabajo de matriz secundario se complete de forma secuencial, a partir del índice 0. También puede especificar una dependencia de tipo `N_TO_N` con un ID de trabajo para los trabajos de matriz. Una dependencia `N_TO_N` conlleva que cada índice secundario de este trabajo deba esperar a que se complete el índice secundario correspondiente de cada dependencia antes de comenzar. La sintaxis de este parámetro es `"= jobld<string>, tipo=<string>;..."`.

awsbstat

Muestra los trabajos que se envían en la cola de trabajos del clúster.

```
awsbstat [-h] [-c CLUSTER] [-s STATUS] [-e] [-d] [job_ids [job_ids ...]]
```

Argumentos de posición

job_ids

Especifica la lista de tareas separadas por espacios IDs que se mostrará en la salida. Si el trabajo es una matriz de trabajo, se muestran todos los trabajos secundarios. Si se solicita un solo trabajo, se muestra en una versión detallada.

Argumentos con nombre

-c *CLUSTER*, --cluster *CLUSTER*

Indica el clúster que se va a utilizar.

-s *STATUS*, --status *STATUS*

Especifica una lista de estados de los trabajos separados por comas que se van a incluir. El estado del trabajo predeterminado es "activo". Los valores aceptados son: SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING, SUCCEEDED, FAILED y ALL.

Valor predeterminado: "SUBMITTED,PENDING,RUNNABLE,STARTING y RUNNING"

-e, --expand-children

Amplía los trabajos con elementos secundarios (tanto los de matriz como los paralelos de varios nodos).

Valor predeterminado: False

-d, --details

Muestra detalles de los trabajos.

Valor predeterminado: False

awsbout

Muestra la salida de un trabajo especificado.

```
awsbout [ - h ] [ - c CLUSTER ] [ - hd HEAD ] [ - t TAIL ] [ - s ] [ - sp STREAM_PERIOD ] job_id
```

Argumentos de posición

job_id

Especifica el ID de trabajo.

Argumentos con nombre

-c *CLUSTER*, --cluster *CLUSTER*

Indica el clúster que se va a utilizar.

-hd *HEAD*, --head *HEAD*

Obtiene el primero *HEAD* líneas del resultado del trabajo.

-t *TAIL*, --tail *TAIL*

Obtiene las últimas líneas <tail> de la salida del trabajo.

-s, --stream

Obtiene la salida del trabajo y, a continuación, espera a que se produzca una salida adicional. Este argumento puede usarse junto a `-tail` para comenzar desde las últimas líneas <tail> de la salida del trabajo.

Valor predeterminado: False

-sp *STREAM_PERIOD*, --stream-period *STREAM_PERIOD*

Establece el periodo de streaming.

Predeterminado: 5

awsbkill

Cancela o termina trabajos enviados en el clúster.

```
awsbkill [ - h ] [ - c CLUSTER ] [ - r REASON ] job_ids [ job_ids ... ]
```

Argumentos de posición

job_ids

Especifica la lista de tareas separadas por espacios que se van IDs a cancelar o terminar.

Argumentos con nombre

-c CLUSTER, --cluster CLUSTER

Indica el nombre del clúster que se va a utilizar.

-r REASON, --reason REASON

Indica el mensaje que se asociará a un trabajo, en el cual se explica el motivo de su cancelación.

Predeterminado:»Terminated by the user”

awsbqueues

Muestra la cola de trabajos que se asocia al clúster.

```
awsbqueues [ - h ] [ - c CLUSTER ] [ - d ] [ job_queues [ job_queues ... ]]
```

Argumentos de posición

job_queues

Especifica la lista de nombres de colas separados por espacios que se van a mostrar. Si se solicita una sola cola, se muestra en una versión detallada.

Argumentos con nombre

-c CLUSTER, --cluster CLUSTER

Especifica el nombre del clúster que se va a utilizar.

-d, --details

Indica si se deben mostrar los detalles de las colas.

Valor predeterminado: False

awsbhosts

Muestra los hosts que pertenecen al entorno informático del clúster.

```
awsbhosts [ - h ] [ - c CLUSTER ] [ - d ] [ instance_ids [ instance_ids ... ] ]
```

Argumentos de posición

instance_ids

Especifica una lista de instancias separadas por espacios. IDs Si se solicita una sola instancia, se muestra en una versión detallada.

Argumentos con nombre

-c *CLUSTER*, --cluster *CLUSTER*

Especifica el nombre del clúster que se va a utilizar.

-d, --details

Indica si se deben mostrar los detalles de los hosts.

Valor predeterminado: False

AWS ParallelCluster recursos y etiquetado

Con AWS ParallelCluster ella puede crear etiquetas para rastrear y administrar sus AWS ParallelCluster recursos. Defina las etiquetas que desea crear y propagar AWS CloudFormation a todos los recursos del clúster en la [tags](#) sección del archivo de configuración del clúster. También puede usar etiquetas que AWS ParallelCluster se generan automáticamente para rastrear y administrar sus recursos.

Al crear un clúster, el clúster y sus recursos se etiquetan con las etiquetas AWS ParallelCluster y AWS sistemas definidas en esta sección.

AWS ParallelCluster aplica etiquetas a las instancias, los volúmenes y los recursos del clúster. Para identificar la pila de clústeres, AWS CloudFormation aplica etiquetas de AWS sistema a las

instancias del clúster. Para identificar las plantillas de EC2 lanzamiento del clúster, EC2 aplica etiquetas de sistema a las instancias. Puede usar estas etiquetas para ver y administrar sus AWS ParallelCluster recursos.

No puede modificar las etiquetas AWS del sistema. Para evitar que la AWS ParallelCluster funcionalidad se vea afectada, no modifique las AWS ParallelCluster etiquetas.

El siguiente es un ejemplo de una etiqueta de AWS sistema para un AWS ParallelCluster recurso. No puede modificarla.

```
"aws:cloudformation:stack-name"="parallelcluster-clustername-  
MasterServerSubstack-ABCD1234EFGH"
```

Los siguientes son ejemplos de AWS ParallelCluster etiquetas aplicadas a un recurso. No la modifique.

```
"aws-parallelcluster-node-type"="Master"
```

```
"Name"="Master"
```

```
"Version"="2.11.9"
```

Puede ver estas etiquetas en la EC2 sección de AWS Management Console.

Ver etiquetas

1. Navegue por la EC2 consola en <https://console.aws.amazon.com/ec2/>.
2. Para ver todas las etiquetas del clúster, elija Etiquetas en el panel de navegación.
3. Para ver las etiquetas de clúster por instancia, elija Instancias en el panel de navegación.
4. Seleccione una instancia de clúster.
5. Seleccione la pestaña Administrar etiquetas en los detalles de la instancia y vea las etiquetas.
6. Elija la pestaña Almacenamiento en los detalles de la instancia.
7. Seleccione el ID de volumen.
8. En Volúmenes, seleccione el volumen.
9. Seleccione la pestaña Etiquetas en los detalles del volumen y vea las etiquetas.

AWS ParallelCluster etiquetas de instancia del nodo principal

| Clave | Valor de etiqueta |
|-------------------------------|---|
| ClusterName | <i>clustername</i> |
| Name | Master |
| Application | parallelcluster- <i>clustername</i> |
| aws:ec2launchtemplate:id | <i>lt-1234567890abcdef0</i> |
| aws:ec2launchtemplate:version | <i>1</i> |
| aws-parallelcluster-node-type | Master |
| aws:cloudformation:stack-name | parallelcluster- <i>clustername</i> - MasterServerSubstack- <i>ABCD1234E FGH</i> |
| aws:cloudformation:logical-id | MasterServer |
| aws:cloudformation:stack-id | arn:aws:cloudformation: <i>region- id</i> : <i>ACCOUNTID</i> :stack/parallelclu ster- <i>clustername</i> -MasterSe rverSubstack- <i>ABCD1234E FGH</i> / <i>1234abcd-12ab-12ab-12ab-123 4567890abcdef0</i> |
| Version | <i>2.11.9</i> |

AWS ParallelCluster etiquetas de volumen raíz del nodo principal

| Clave de etiqueta | Valor de etiqueta |
|-------------------------------|-------------------------------------|
| ClusterName | <i>clustername</i> |
| Application | parallelcluster- <i>clustername</i> |
| aws-parallelcluster-node-type | Master |

AWS ParallelCluster etiquetas de instancia de nodo de cómputo

| Clave | Valor de etiqueta |
|-------------------------------|-----------------------------|
| ClusterName | <i>clustername</i> |
| aws-parallelcluster-node-type | Compute |
| aws:ec2launchtemplate:id | <i>lt-1234567890abcdef0</i> |
| aws:ec2launchtemplate:version | <i>1</i> |
| QueueName | <i>queue-name</i> |
| Version | <i>2.11.9</i> |

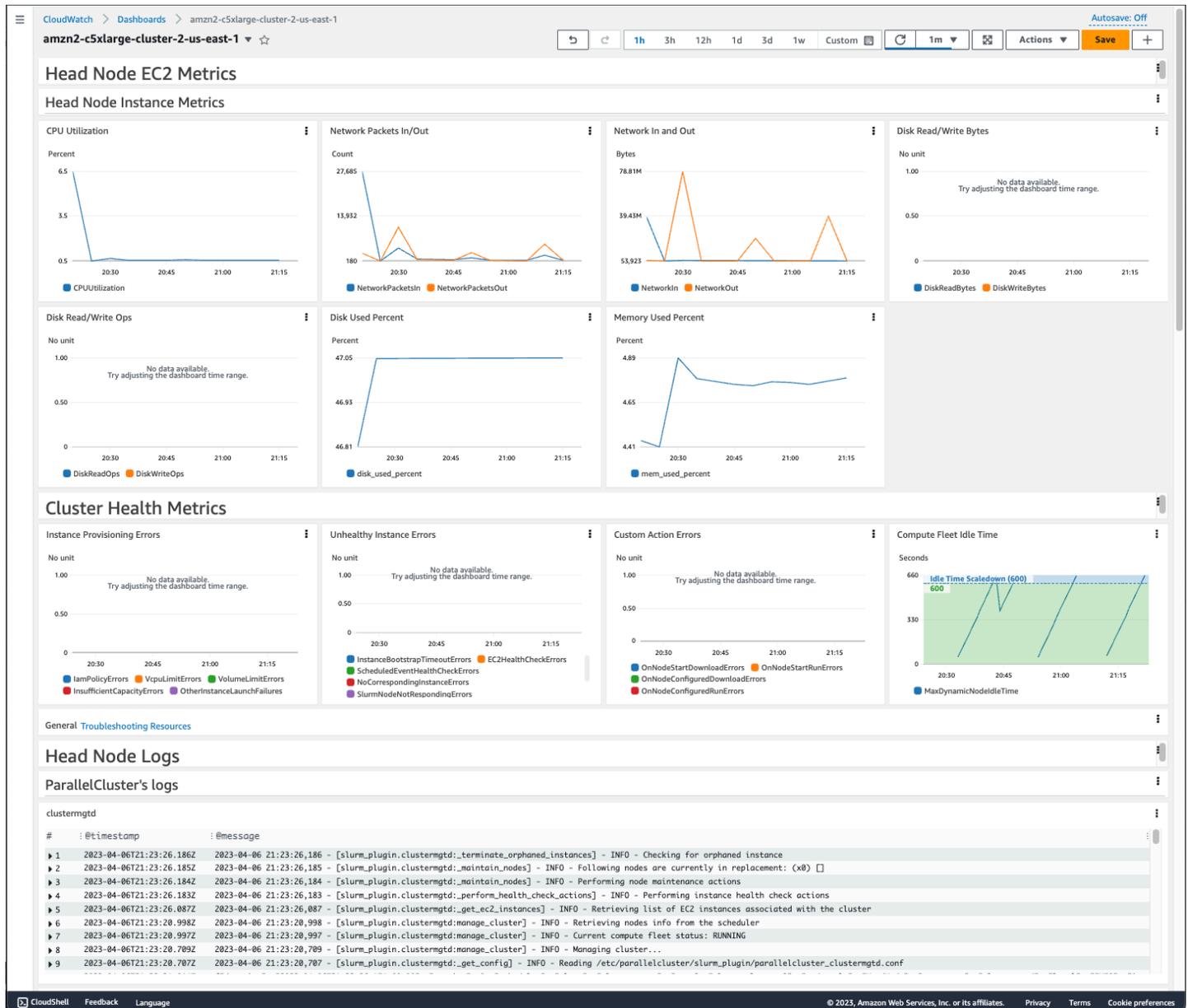
AWS ParallelCluster etiquetas de volumen raíz del nodo de cálculo

| Clave de etiqueta | Valor de etiqueta |
|-------------------------------|-------------------------------------|
| ClusterName | <i>clustername</i> |
| Application | parallelcluster- <i>clustername</i> |
| aws-parallelcluster-node-type | Compute |
| QueueName | <i>queue-name</i> |
| Version | <i>2.11.9</i> |

CloudWatch Panel de control de Amazon

A partir de AWS ParallelCluster la versión 2.10.0, se crea un CloudWatch panel de Amazon al crear el clúster. Esto facilita la supervisión de los nodos del clúster y la visualización de los registros almacenados en Amazon CloudWatch Logs. El nombre del panel es `parallelcluster-ClusterName-Region`. *ClusterName* es el nombre de su clúster y *Region* es el Región de AWS del clúster. Puede acceder al panel de control desde la consola o abriendo `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=parallelcluster-ClusterName`.

La siguiente imagen muestra un ejemplo de CloudWatch panel para un clúster.



La primera sección del panel muestra gráficos de las EC2 métricas del nodo principal. Si su clúster tiene almacenamiento compartido, en la siguiente sección se muestran las métricas de almacenamiento compartido. La sección final muestra los registros ParallelCluster del nodo principal agrupados por registros, registros del programador, registros de NICE DCV integración y registros del sistema.

Para obtener más información sobre los CloudWatch paneles de Amazon, consulte [Uso de los CloudWatch paneles de Amazon](#) en la Guía CloudWatch del usuario de Amazon.

Si no quieres crear el CloudWatch panel de Amazon, debes completar estos pasos: primero, añade una [\[dashboard\]sección](#) a tu archivo de configuración y, a continuación, añade el nombre de esa sección como el valor de la [dashboard_settings](#) configuración de tu [\[cluster\]sección](#). En su [\[dashboard\]sección](#), `enable = false` defina.

Por ejemplo, si se nombra [\[dashboard\]la sección](#) myDashboard y se nombra [\[cluster\]la sección](#) myCluster, los cambios se parecerán a esto.

```
[cluster MyCluster]
dashboard_settings = MyDashboard
...

[dashboard MyDashboard]
enable = false
```

Integración con Amazon CloudWatch Logs

A partir de AWS ParallelCluster la versión 2.6.0, los registros comunes se almacenan en CloudWatch Logs de forma predeterminada. Para obtener más información sobre CloudWatch los registros, consulte la [Guía del usuario de Amazon CloudWatch Logs](#). Para configurar la integración de CloudWatch Logs, consulte la [\[cw_log\]sección](#) y la [cw_log_settings](#) configuración.

Se crea un grupo de registros para cada clúster con el nombre `/aws/parallelcluster/cluster-name` (por ejemplo, `/aws/parallelcluster/testCluster`). Cada registro (o conjunto de registros si la ruta contiene un `*`) de cada nodo tiene un flujo de registro denominado `{hostname}.{instance_id}.{logIdentifier}`. (Por ejemplo: `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`). El [CloudWatch agente](#) envía los datos CloudWatch de registro y se ejecuta como `root` en todas las instancias del clúster.

A partir de AWS ParallelCluster la versión 2.10.0, se crea un CloudWatch panel de Amazon al crear el clúster. Este panel facilita la revisión de los registros almacenados en CloudWatch Logs. Para obtener más información, consulte [CloudWatch Panel de control de Amazon](#).

Esta lista contiene los `logIdentifier` y la ruta de los flujos de registro disponibles para las plataformas, los programadores y los nodos.

Flujos de registro disponibles para plataformas, programadores y nodos

| Plataformas | Programadores | Nodos | Flujos de registro |
|----------------------------|------------------|-------------------------|--|
| amazon centos ubuntu | awsbatc Slurm | HeadNc | dcv-authenticator: /var/log/parallelcluster/parallelcluster_dcv_authenticator.log dcv-ext-authenticator: /var/log/parallelcluster/parallelcluster_dcv_connect.log dcv-agent: /var/log/dcv/agent.*.log dcv-xsession: /var/log/dcv/dcv-xsession.*.log dcv-server: /var/log/dcv/server.log dcv-session-launcher: /var/log/dcv/sessionlauncher.log Xdcv: /var/log/dcv/Xdcv.*.log cfn-init: /var/log/cfn-init.log chef-client: /var/log/chef-client.log |
| amazon centos ubuntu | awsbatc Slurm | Comput eet HeadNc | cloud-init: /var/log/cloud-init.log supervisord: /var/log/supervisord.log |
| amazon centos ubuntu | Slurm | Comput eet | cloud-init-output: /var/log/cloud-init-output.log computemgtd: /var/log/parallelcluster/computemgtd slurmd: /var/log/slurmd.log |
| amazon centos ubuntu | Slurm | HeadNc | clustermgtd: /var/log/parallelcluster/clustermgtd slurm_resume: /var/log/parallelcluster/slurm_resume.log |

| Plataformas | Programas | Nodos | Flujos de registros |
|---------------|----------------|-------------------|--|
| | | | slurm_suspend: /var/log/parallelcluster/slurm_suspend.log slurmctld: /var/log/slurmctld.log |
| amazon centos | awsbatch Slurm | Compute HeadNodes | system-messages: /var/log/messages |
| ubuntu | awsbatch Slurm | Compute HeadNodes | syslog: /var/log/syslog |

Los trabajos de los clústeres que se utilizan AWS Batch almacenan en los registros la salida de los trabajos que alcanzaron un RUNNING FAILED estado o un determinado estado. SUCCEEDED CloudWatch El grupo de registros es /aws/batch/job, y el formato del nombre del flujo de registro es *jobDefinitionName/default/ecs_task_id*. De forma predeterminada, estos registros están configurados para no caducar, pero se puede modificar el periodo de retención. Para obtener más información, consulta [Cambiar la retención de datos de registro en CloudWatch los registros](#) en la Guía del usuario de Amazon CloudWatch Logs.

Note

chef-client, cloud-init-output, clustermgtd, computemgtd, slurm_resume, y slurm_suspend se agregaron en la AWS ParallelCluster versión 2.9.0. En la AWS ParallelCluster versión 2.6.0, /var/log/cfn-init-cmd.log (cfn-init-cmd) y /var/log/cfn-wire.log (cfn-wire) también se almacenaban en CloudWatch los registros.

Elastic Fabric Adapter

El Elastic Fabric Adapter (EFA) es un dispositivo de red que cuenta con capacidades de derivación del sistema operativo para comunicaciones de red de baja latencia con otras instancias de la misma

subred. EFA se expone mediante Libfabric y las aplicaciones pueden usarlo mediante la interfaz de paso de mensajes (). MPI

Para usar EFA con AWS ParallelCluster, añada la línea `enable_efa = true` a la [\[queue\]sección](#).

Para ver la lista de EC2 instancias compatibles EFA, consulta los [tipos de instancias compatibles](#) en la Guía del EC2 usuario de Amazon para instancias de Linux.

Para obtener más información, consulte la configuración de `enable_efa` en la sección .

Se debe utilizar un grupo de ubicación del clúster para minimizar las latencias entre instancias. Para obtener más información, consulte [placement](#) y [placement_group](#).

Para obtener más información, consulte [Elastic Fabric Adapter](#) en la Guía del EC2 usuario de Amazon y [escale HPC las cargas de trabajo con un adaptador de tela elástica y AWS ParallelCluster](#) en el blog de código AWS abierto.

Note

De forma predeterminada, Ubuntu las distribuciones habilitan `ptrace` protección (rastreo del proceso). A partir de AWS ParallelCluster 2.6.0, `ptrace` la protección está desactivada para que Libfabric funcione correctamente. Para obtener más información, consulta [Cómo deshabilitar la protección ptrace](#) en la Guía del EC2 usuario de Amazon.

Note

La compatibilidad con EFA las instancias Graviton2 basadas en ARM se agregó en AWS ParallelCluster la versión 2.10.1.

Soluciones Intel Select

AWS ParallelCluster está disponible como una solución Intel Select para simulación y modelado. Se verifica que las configuraciones cumplen con los estándares establecidos en la [especificación de HPC plataforma Intel](#), utilizan tipos de instancias Intel específicos y están configuradas para usar la interfaz de red [Elastic Fabric Adapter](#) (EFA). AWS ParallelCluster es la primera solución en la nube que cumple los requisitos del programa Intel Select Solutions. Los tipos admitidos son

c5n.18xlarge, m5n.24xlarge y r5n.24xlarge. A continuación se proporciona un ejemplo de configuración compatible con el estándar Intel Select Solutions.

Example Configuración de Intel Select Solutions

```
[global]
update_check = true
sanity_check = true
cluster_template = intel-select-solutions

[aws]
aws_region_name = <Your Región de AWS>

[scaling demo]
scaledown_idletime = 5

[cluster intel-select-solutions]
key_name = <Your SSH key name>
base_os = centos7
scheduler = slurm
enable_intel_hpc_platform = true
master_instance_type = c5.xlarge
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = c5n,m5n,r5n
master_root_volume_size = 200
compute_root_volume_size = 80

[queue c5n]
compute_resource_settings = c5n_i1
enable_efa = true
placement_group = DYNAMIC

[compute_resource c5n_i1]
instance_type = c5n.18xlarge
max_count = 5

[queue m5n]
compute_resource_settings = m5n_i1
enable_efa = true
placement_group = DYNAMIC

[compute_resource m5n_i1]
instance_type = m5n.24xlarge
```

```

max_count = 5

[queue r5n]
compute_resource_settings = r5n_i1
enable_efa = true
placement_group = DYNAMIC

[compute_resource r5n_i1]
instance_type = r5n.24xlarge
max_count = 5

```

Para obtener más información sobre AWS ParallelCluster las especificaciones de la HPC plataforma Intel, consulte [Especificación de plataforma Intel HPC](#).

Habilite Intel MPI

Intel MPI está disponible en. AWS ParallelCluster AMIs Para utilizar IntelMPI, debe reconocer y aceptar los términos de la [licencia de software simplificada de Intel](#). De forma predeterminada, la opción Abrir MPI aparece en la ruta. Para activar Intel MPI en lugar de OpenMPI, primero debe cargar el MPI módulo Intel. A continuación, debe instalar la versión más reciente utilizando `module load intelmpi`. El nombre exacto del módulo cambia con cada actualización. Para saber qué módulos están disponibles, ejecute `module avail`. El resultado es el siguiente.

```

$ module avail

----- /usr/share/Modules/modulefiles
-----
dot                libfabric-aws/1.8.1amzn1.3 module-info          null
                  use.own
module-git         modules                openmpi/4.0.2

----- /etc/modulefiles
-----

----- /opt/intel/impi/2019.7.217/intel64/modulefiles
-----
intelmpi

```

```
$ module load intelmpi
```

Para saber qué módulos están cargados, ejecute `module list`.

```
$ module list
Currently Loaded Modulefiles:
 1) intelmpi
```

Para comprobar que Intel MPI está activado, ejecute `mpirun --version`.

```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2019 Update 7 Build 20200312 (id:
5dc2dd3e9)
Copyright 2003-2020, Intel Corporation.
```

Una vez cargado el MPI módulo Intel, se cambian varias rutas para utilizar las MPI herramientas Intel. Para ejecutar el código compilado por las MPI herramientas de Intel, cargue primero el MPI módulo Intel.

Note

Intel MPI no es compatible con las instancias AWS basadas en Graviton.

Note

Antes de AWS ParallelCluster la versión 2.5.0, Intel no MPI estaba disponible AWS ParallelCluster AMIs en las regiones de China (Pekín) y China (Ningxia).

Especificación de plataforma Intel HPC

AWS ParallelCluster cumple con la especificación de HPC plataforma Intel. La especificación de HPC plataforma Intel proporciona un conjunto de requisitos de computación, estructura, memoria, almacenamiento y software para ayudar a alcanzar un alto nivel de calidad y compatibilidad con HPC las cargas de trabajo. Para obtener más información, consulte la [especificación de la HPC plataforma Intel](#) y [las aplicaciones cuya compatibilidad ha sido verificada con la especificación de la HPC plataforma Intel](#).

Para cumplir con la especificación de HPC plataforma Intel, se deben cumplir los siguientes requisitos:

- El sistema operativo debe ser CentOS (7 [base_os](#) = centos7).

- El tipo de instancia de los nodos de cómputo debe tener un procesador Intel CPU y 64 GB de memoria como mínimo. Para la familia de tipos de instancias c5, esto significa que el tipo de instancia debe ser al menos c5.9xlarge ([compute_instance_type](#) = c5.9xlarge).
- El nodo principal debe tener al menos 200 GB de almacenamiento.
- Debe aceptarse el acuerdo de licencia de usuario final para Intel Parallel Studio ([enable_intel_hpc_platform](#) = true).
- Cada nodo de computación debe tener al menos 80 GB de almacenamiento ([compute_root_volume_size](#) = 80).

El almacenamiento puede ser local o en una red (NFS compartido desde el nodo principal, Amazon EBS o FSx Lustre) y se puede compartir.

Bibliotecas de rendimiento de Arm

A partir de AWS ParallelCluster la versión 2.10.1, las bibliotecas de rendimiento de Arm están disponibles en los ubuntu2004 valores AWS ParallelCluster AMIs for alinux2centos8,ubuntu1804, y para esta configuración. [base_os](#) Las bibliotecas de rendimiento de Arm proporcionan bibliotecas matemáticas básicas estándar optimizadas para aplicaciones informáticas de alto rendimiento en procesadores Arm. Para utilizar las bibliotecas Arm Performance, debe conocer y aceptar los términos del [contrato de licencia para el usuario final de las bibliotecas Arm Performance Libraries \(versión gratuita\)](#). Para obtener más información sobre las bibliotecas de rendimiento de Arm, consulte [Bibliotecas de rendimiento de Arm gratuitas](#).

Para habilitar las bibliotecas de rendimiento de Arm, primero debe cargar el módulo de bibliotecas de rendimiento de Arm. `armpl-21.0.0` necesita GCC -9,3 como requisito, cuando cargue el `armpl/21.0.0` módulo, el `gcc/9.3` módulo también se cargará. El nombre exacto del módulo cambia con cada actualización. Para saber qué módulos están disponibles, ejecute `module avail`. A continuación, debe instalar la versión más reciente utilizando `module load armpl`.

```
$ module avail
----- /usr/share/Modules/modulefiles
-----
armpl/21.0.0      dot          libfabric-aws/1.11.1amzn1.0
module-git
module-info      modules     null          openmpi/4.1.0
use.own
```

Para cargar un módulo, ejecute `module load modulename`. Puede añadir esto al script utilizado para ejecutar `mpirun`.

```
$ module load armpl
```

```
Use of the free of charge version of Arm Performance Libraries is subject to the terms
and
conditions of the Arm Performance Libraries (free version) - End User License
Agreement
(EULA). A copy of the EULA can be found in the
'/opt/arm/armpl/21.0.0/arm-performance-libraries_21.0_gcc-9.3/license_terms' folder
```

Para saber qué módulos están cargados, ejecute `module list`.

```
$ module list
```

```
Currently Loaded Modulefiles:
```

- 1) /opt/arm/armpl/21.0.0/modulefiles/armpl/gcc-9.3
- 2) /opt/arm/armpl/21.0.0/modulefiles/armpl/21.0.0_gcc-9.3
- 3) armpl/21.0.0

Para verificar que las bibliotecas de rendimiento de Arm estén habilitadas, ejecute pruebas de ejemplo.

```
$ sudo chmod 777 /opt/arm/armpl/21.0.0/armpl_21.0_gcc-9.3/examples
$ cd /opt/arm/armpl/21.0.0/armpl_21.0_gcc-9.3/examples
$ make
...
Testing: no example difference files were generated.
Test passed OK
```

Una vez cargado el módulo Arm Performance Libraries, se cambian varias rutas para utilizar las herramientas de Arm Performance Libraries. Para ejecutar el código compilado por las herramientas de las bibliotecas de rendimiento de Arm, cargue primero el módulo de bibliotecas de rendimiento de Arm.

Note

AWS ParallelCluster utilizan versiones entre 2.10.1 y 2.10.4. `armpl/20.2.1`

Conéctese al nodo principal a través de Amazon DCV

Amazon DCV es una tecnología de visualización remota que permite a los usuarios conectarse de forma segura a aplicaciones 3D con uso intensivo de gráficos alojadas en un servidor remoto de alto rendimiento. Para obtener más información, consulta [Amazon DCV](#).

El DCV software de Amazon se instala automáticamente en el nodo principal cuando se utiliza `base_os = alinux2`, `base_os = centos7`, `base_os = ubuntu1804` o `base_os = ubuntu2004`.

Si el nodo principal es una ARM instancia, el DCV software de Amazon se instala automáticamente en él cuando se usa `base_os = alinux2`, `base_os = centos7`, o `base_os = ubuntu1804`.

Para habilitar Amazon DCV en el nodo principal, `dcv_settings` debe contener el nombre de una `[dcv]` sección que tenga `enable = master` y `base_os` esté configurada en `alinux2centos7`, `ubuntu1804`, o `ubuntu2004`. Si el nodo principal es una ARM instancia, `base_os` debe estar configurado en `alinux2`, `centos7`, o `ubuntu1804`. De esta forma, AWS ParallelCluster establece el parámetro `shared_dir` de configuración del clúster en la [carpeta de almacenamiento DCV del servidor](#).

```
[cluster custom-cluster]
...
dcv_settings = custom-dcv
...
[dcv custom-dcv]
enable = master
```

Para obtener más información sobre los parámetros DCV de configuración de Amazon, consulte [dcv_settings](#). Para conectarse a la DCV sesión de Amazon, utilice el `pcluster dcv` comando.

Note

La compatibilidad con Amazon DCV on centos8 se eliminó en la AWS ParallelCluster versión 2.10.4. El soporte para Amazon DCV on centos8 se agregó en la AWS ParallelCluster versión 2.10.0. La compatibilidad con Amazon DCV en instancias AWS basadas en Graviton se agregó en la AWS ParallelCluster versión 2.9.0. Support para Amazon DCV está activado a `linux2` y `ubuntu1804` se añadió en la AWS

ParallelCluster versión 2.6.0. El soporte para Amazon DCV on centos7 se añadió en la AWS ParallelCluster versión 2.5.0.

 Note

Amazon no DCV es compatible con las instancias AWS basadas en Graviton en AWS ParallelCluster las versiones 2.8.0 y 2.8.1.

DCVHTTPSCertificado de Amazon

Amazon genera DCV automáticamente un certificado autofirmado para proteger el tráfico entre el DCV cliente de Amazon y el DCV servidor de Amazon.

Para sustituir el DCV certificado autofirmado de Amazon predeterminado por otro certificado, conéctese primero al nodo principal. A continuación, copie el certificado y la clave en la carpeta /etc/dcv antes de ejecutar el comando [pcluster dcv](#).

Para obtener más información, consulta Cómo [cambiar el TLS certificado](#) en la Guía del DCV administrador de Amazon.

Licenciamiento Amazon DCV

El DCV servidor de Amazon no requiere un servidor de licencias cuando se ejecuta en EC2 instancias de Amazon. Sin embargo, el DCV servidor de Amazon debe conectarse periódicamente a un bucket de Amazon S3 para determinar si hay una licencia válida disponible.

AWS ParallelCluster añade automáticamente los permisos necesarios alParallelClusterInstancePolicy. Cuando utilices una política de IAM instancias personalizada, utiliza los permisos descritos en [Amazon DCV on Amazon EC2](#) en la Guía del DCV administrador de Amazon.

Para obtener sugerencias acerca de la solución de problemas, consulte [Solución de problemas en Amazon DCV](#).

Uso de `pcluster update`

A partir de la AWS ParallelCluster versión 2.8.0, [pcluster update](#) analiza los ajustes utilizados para crear el clúster actual y los ajustes del archivo de configuración para detectar problemas. Si se descubre algún problema, se informa al respecto y se muestran los pasos a seguir para solucionarlo. Por ejemplo, si se cambia la configuración de [compute_instance_type](#), se debe detener la flota de computación para que se pueda proceder con la actualización. Este problema se notifica cuando se descubre. Si no se informa de ningún problema de bloqueo, se le preguntará si desea aplicar los cambios.

La documentación de cada configuración define la política de actualización de esa configuración.

Política de actualización: esta configuración se puede cambiar durante una actualización., Política de actualización: esta configuración se puede cambiar durante una actualización.

Tras cambiar esta configuración, el clúster se puede actualizar mediante [pcluster update](#).

Política de actualización: si se cambia esta configuración, no se permite la actualización.

Esta configuración no se puede cambiar si no se ha eliminado el clúster existente. Se debe revertir el cambio o se debe eliminar el clúster (utilizando [pcluster delete](#)) y, a continuación, crear un clúster nuevo (utilizando [pcluster create](#)) el clúster anterior.

Política de actualización: esta configuración no se analiza durante una actualización.

Tras cambiar esta configuración, el clúster se puede actualizar mediante [pcluster update](#).

Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.

Estos ajustes no se pueden cambiar mientras exista la flota de computación. O bien se debe revertir el cambio o bien se debe detener (utilizando [pcluster stop](#)), actualizar (utilizando [pcluster update](#)) y, a continuación, crear una nueva flota de computación (utilizando [pcluster start](#)).

Política de actualización: esta configuración no se puede reducir durante una actualización.

Estos ajustes se pueden cambiar, pero no se pueden reducir. Si se debe reducir esta configuración, es necesario eliminar el clúster (usando [pcluster delete](#)) y crear un nuevo clúster (usando [pcluster create](#)).

Política de actualización: para reducir el tamaño de una cola por debajo del número actual de nodos, es necesario detener primero la flota de computación.

Estos ajustes se pueden cambiar, pero si el cambio pretende reducir el tamaño de la cola por debajo del tamaño actual, la flota de computación se debe detener (utilizando [pcluster stop](#)), actualizar (utilizando [pcluster update](#)) y, a continuación, crear una nueva flota de computación (utilizando [pcluster start](#)).

Política de actualización: para reducir el número de nodos estáticos de una cola, es necesario detener primero la flota de computación.

Estos ajustes se pueden cambiar, pero si el cambio quiere reducir el número de nodos estáticos de la cola por debajo del tamaño actual, la flota de computación se debe detener (utilizando [pcluster stop](#)), actualizar (utilizando [pcluster update](#)) y, a continuación, crear una nueva flota de computación (utilizando [pcluster start](#)).

Política de actualización: si se cambia esta configuración, no se permite la actualización. No se puede forzar la actualización de esta configuración.

Esta configuración no se puede cambiar si no se ha eliminado el clúster existente. Se debe revertir el cambio o se debe eliminar el clúster (utilizando [pcluster delete](#)) y, a continuación, crear un clúster nuevo (utilizando [pcluster create](#)) el clúster anterior.

Política de actualización: si los sistemas de archivos AWS ParallelCluster gestionados de Amazon FSx for Lustre no se especifican en la configuración, esta configuración se puede cambiar durante una actualización.

Esta configuración se puede cambiar si [\[cluster\]fsx_settings](#) no se especifica o si [\[fsx fs\]](#) se especifican ambas [fsx_settings](#) [fsx-fs-id](#) opciones para montar un sistema de archivos externo existente FSx para Lustre.

En este ejemplo se muestra una actualización con algunos cambios que bloquean la actualización.

```
$ pcluster update
Validating configuration file /home/username/.parallelcluster/config...
Retrieving configuration from CloudFormation for cluster test-1...
Found Changes:

# section/parameter      old value      new value
-- -----
[cluster default]
```

```

01* compute_instance_type    t2.micro                    c4.xlarge
02* ebs_settings             ebs2                        -

    [vpc default]
03  additional_sg            sg-0cd61884c4ad16341       sg-0cd61884c4ad11234

    [ebs ebs2]
04* shared_dir               shared                       my/very/very/long/sha...

```

Validating configuration update...

The requested update cannot be performed. Line numbers with an asterisk indicate updates requiring additional actions. Please look at the details below:

#01

Compute fleet must be empty to update "compute_instance_type"

How to fix:

Make sure that there are no jobs running, then run the following command:

```
pcluster stop -c $CONFIG_FILE $CLUSTER_NAME
```

#02

Cannot add/remove EBS Sections

How to fix:

Revert "ebs_settings" value to "ebs2"

#04

Cannot change the mount dir of an existing EBS volume

How to fix:

Revert "my/very/very/long/shared/dir" to "shared"

In case you want to override these checks and proceed with the update please use the `--force` flag. Note that the cluster could end up in an unrecoverable state.

Update aborted.

AMI parcheo y EC2 reemplazo de instancias

Para garantizar que todos los nodos de cómputo del clúster lanzados dinámicamente se comporten de forma coherente, AWS ParallelCluster desactiva las actualizaciones automáticas del sistema operativo de las instancias del clúster. Además, AWS ParallelCluster AMIs se crea un conjunto específico de versiones para cada versión AWS ParallelCluster y las versiones asociadas CLI. Este conjunto específico AMIs permanece sin cambios y solo es compatible con la AWS ParallelCluster

versión para la que se crearon. AWS ParallelCluster AMIs para las versiones publicadas no están actualizadas.

Sin embargo, debido a problemas de seguridad emergentes, es posible que los clientes deseen añadirles parches AMIs y, a continuación, actualizar sus clústeres con los parchesAMI. Esto se ajusta al [modelo de responsabilidad compartida de AWS ParallelCluster](#).

Para ver el conjunto específico de versiones AWS ParallelCluster AMIs compatibles con la AWS ParallelCluster CLI versión que está utilizando actualmente, ejecute:

```
$ pcluster version
```

A continuación, consulta el [archivo amis.txt](#) en AWS ParallelCluster el GitHub repositorio.

El nodo AWS ParallelCluster principal es una instancia estática y puedes actualizarlo manualmente. El reinicio y el reinicio del nodo principal son totalmente compatibles a partir de la AWS ParallelCluster versión 2.11, si el tipo de instancia no tiene un almacén de instancias. Para obtener más información, consulta [Tipos de instancias con volúmenes de almacén de instancias](#) en la Guía del EC2 usuario de Amazon para instancias de Linux. No puedes actualizar ningún clúster existente. AMI

El reinicio del nodo principal y el reinicio con AMI actualizaciones de las instancias de cómputo en clúster son totalmente compatibles a partir de la AWS ParallelCluster versión 3.0.0. Considere la posibilidad de actualizar a la versión más reciente para utilizar estas funciones.

Actualización o sustitución de la instancia del nodo principal

En algunas circunstancias, es posible que tenga que reiniciar el nodo principal. Por ejemplo, es necesario cuando actualiza manualmente el sistema operativo o cuando hay una [retirada programada de una instancia de AWS](#) que impone el reinicio de la instancia del nodo principal.

Si la instancia no tiene unidades efímeras, puede detenerla e iniciarla de nuevo en cualquier momento. En el caso de una retirada programada, al iniciar la instancia detenida, se realiza una migración para usar el nuevo hardware.

Del mismo modo, puede detener e iniciar manualmente una instancia que no tenga almacenes de instancias. En este caso y en otros casos de instancias sin volúmenes efímeros, continúe hasta [Detención e inicio del nodo principal de un clúster](#).

Si la instancia tiene unidades efímeras y se ha detenido, se perderán los datos del almacén de instancias. Puede determinar si el tipo de instancia utilizado para el nodo principal tiene almacenes de instancias a partir de la tabla que se encuentra en [Volúmenes de almacenes de instancias](#).

En las siguientes secciones, se describen las limitaciones del uso de instancias con volúmenes de almacenes de instancias.

Limitaciones del almacén de instancias

Las limitaciones del uso de la AWS ParallelCluster versión 2.11 y de los tipos de instancia con un almacén de instancias son las siguientes:

- Cuando las unidades efímeras no están cifradas (el [encrypted_ephemeral](#) parámetro está establecido `false` o no), una AWS ParallelCluster instancia no puede arrancar después de que se detenga. Esto se debe a que la información sobre objetos efímeros antiguos e inexistentes se graba `fstab` y el sistema operativo intenta montar un almacenamiento inexistente.
- Cuando las unidades efímeras están cifradas (el [encrypted_ephemeral](#) parámetro se establece en `true`), se puede iniciar una AWS ParallelCluster instancia tras una parada, pero las nuevas unidades efímeras no están configuradas, montadas ni disponibles.
- Cuando se cifran las unidades efímeras, se puede reiniciar una AWS ParallelCluster instancia, pero no se puede acceder a las unidades efímeras antiguas (que sobreviven al reinicio de la instancia) porque la clave de cifrado se crea en la memoria que se pierde con el reinicio.

El único caso admitido es el reinicio de la instancia, cuando las unidades efímeras no están cifradas. Esto se debe a que la unidad sobrevive al reinicio y se vuelve a montar debido a la entrada escrita en `fstab`.

Soluciones alternativas para las limitaciones del almacén de instancias

En primer lugar, guarde los datos. Para comprobar si tiene datos que deba conservar, consulte el contenido de la carpeta [ephemeral_dir/](#) (de forma predeterminada). Puede transferir los datos al volumen raíz o a los sistemas de almacenamiento compartido adjuntos al clúster, como Amazon FSxEFS, Amazon o AmazonEBS. Tenga en cuenta que la transferencia de datos al almacenamiento remoto puede conllevar costes adicionales.

La causa principal de las limitaciones está en la lógica que se AWS ParallelCluster utiliza para formatear y montar los volúmenes de almacenes de instancias. La lógica agrega una entrada con `/etc/fstab` el siguiente formato:

```
$ /dev/vg.01/lv_ephemeral ${ephemeral_dir} ext4 noatime,nodiratime 0 0
```

`${ephemeral_dir}` es el valor del parámetro [ephemeral_dir](#) del archivo de configuración `pcluster` (el valor predeterminado es `/scratch`).

Esta línea se agrega para que, si se reinicia un nodo o cuando se reinicia, los volúmenes del almacén de instancias se vuelvan a montar automáticamente. Esto es conveniente porque los datos de las unidades efímeras persisten durante el reinicio. Sin embargo, los datos de las unidades efímeras no persisten durante un ciclo de inicio o parada. Esto significa que están formateadas y montadas sin datos.

El único caso admitido es el reinicio de la instancia cuando las unidades efímeras no están cifradas. Esto se debe a que la unidad sobrevive al reinicio y se vuelve a montar porque está escrita. `fstab`

Para conservar los datos en todos los demás casos, debe eliminar la entrada del volumen lógico antes de detener la instancia. Por ejemplo, elimine `/dev/vg.01/lv_ephemeral` la instancia de `/etc/fstab` antes de detenerla. Una vez hecho esto, se inicia la instancia sin montar los volúmenes efímeros. Sin embargo, el montaje de nuevo en el almacén de instancias no estará disponible después de detener o iniciar la instancia.

Después de guardar los datos y eliminar la `fstab` entrada, continúe con la siguiente sección.

Detención e inicio del nodo principal de un clúster

Note

A partir de AWS ParallelCluster la versión 2.11, solo se admiten la función de parada e inicio del nodo principal si el tipo de instancia no tiene un almacén de instancias.

1. Compruebe que no haya ningún trabajo en ejecución en el clúster.

Cuando se usa un Slurm planificador:

- Si no se especifica la opción `--no-requeue` de `sbatch`, los trabajos en ejecución se vuelven a poner en cola.
- Si se especifica la opción `--no-requeue`, se produce un error en los trabajos en ejecución.

2. Solicite la detención de la flota de computación de un clúster:

```
$ pcluster stop cluster-name
Compute fleet status is: RUNNING. Submitting status change request.
Request submitted successfully. It might take a while for the transition to
complete.
Please run 'pcluster status' if you need to check compute fleet status
```

3. Espere hasta que el estado de la flota de computación sea STOPPED:

```
$ pcluster status cluster-name
...
ComputeFleetStatus: STOP_REQUESTED
$ pcluster status cluster-name
...
ComputeFleetStatus: STOPPED
```

4. Para las actualizaciones manuales con un reinicio del sistema operativo o un reinicio de la instancia, puede usar la tecla AWS Management Console o AWS CLI. A continuación, se muestra un ejemplo de cómo se utiliza AWS CLI.

```
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
  "StoppingInstances": [
    {
      "CurrentState": {
        "Name": "stopping"
        ...
      },
      "InstanceId": "i-1234567890abcdef0",
      "PreviousState": {
        "Name": "running"
        ...
      }
    }
  ]
}
$ aws ec2 start-instances --instance-ids 1234567890abcdef0
{
  "StartingInstances": [
    {
      "CurrentState": {
        "Name": "pending"
        ...
      }
    }
  ]
}
```

```
    },  
    "InstanceId": "i-1234567890abcdef0",  
    "PreviousState": {  
      "Name": "stopped"  
      ...  
    }  
  }  
]  
}
```

5. Inicie la flota de computación del clúster:

```
$ pcluster start cluster-name  
Compute fleet status is: STOPPED. Submitting status change request.  
Request submitted successfully. It might take a while for the transition to  
complete.  
Please run 'pcluster status' if you need to check compute fleet status
```

AWS ParallelCluster CLI comandos

`pcluster` y `pcluster-config` son los AWS ParallelCluster CLI comandos. Se utilizan `pcluster` para lanzar y gestionar HPC los clústeres Nube de AWS y `pcluster-config` para actualizar la configuración.

Para usarlo `pcluster`, debe tener un IAM rol con los [permisos](#) necesarios para ejecutarlo.

```
pcluster [ -h ] ( create | update | delete | start | stop | status | list |
                instances | ssh | dcw | createami | configure | version ) ...
pcluster-config [-h] (convert) ...
```

Temas

- [pcluster](#)
- [pcluster-config](#)

pcluster

`pcluster` es el AWS ParallelCluster CLI comando principal. Se utiliza `pcluster` para lanzar y administrar HPC clústeres en Nube de AWS.

```
pcluster [ -h ] ( create | update | delete | start | stop | status | list |
                instances | ssh | dcw | createami | configure | version ) ...
```

Argumentos

`pcluster` *command*

Posibles opciones: [configure](#), [create](#), [createami](#), [dcw](#), [delete](#), [instances](#), [list](#), [ssh](#), [start](#), [status](#), [stop](#), [update](#), [version](#)

Subcomandos:

Temas

- [pcluster configure](#)
- [pcluster create](#)
- [pcluster createami](#)
- [pcluster dcw](#)
- [pcluster delete](#)
- [pcluster instances](#)
- [pcluster list](#)
- [pcluster ssh](#)
- [pcluster start](#)
- [pcluster status](#)
- [pcluster stop](#)
- [pcluster update](#)
- [pcluster version](#)

pcluster configure

Comienza una AWS ParallelCluster configuración. Para obtener más información, consulte [Configuración AWS ParallelCluster](#).

```
pcluster configure [ -h ] [ -c CONFIG_FILE ] [ -r REGION ]
```

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster configure`.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica la ruta completa del archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

Para obtener más información, consulte [Configuración AWS ParallelCluster](#).

-r REGION, --region REGION

Especifica el Región de AWS que se va a utilizar. Si se especifica, la configuración omite la Región de AWS detección.

Para eliminar los recursos de red delVPC, puede eliminar la pila de CloudFormation redes. El nombre de la pila comienza con»parallelclusternetworking-»y contiene la hora de creación en formatoYYYYMMDDHHMMSS"». Puede enumerar las pilas mediante el comando [list-stacks](#).

```
$ aws --region us-east-1 cloudformation list-stacks \
  --stack-status-filter "CREATE_COMPLETE" \
  --query "StackSummaries[].StackName" | \
  grep -e "parallelclusternetworking-"
  "parallelclusternetworking-pubpriv-20191029205804"
```

La pila se puede eliminar mediante el comando [delete-stack](#).

```
$ aws --region us-east-1 cloudformation delete-stack \
  --stack-name parallelclusternetworking-pubpriv-20191029205804
```

Lo VPC que [pcluster configure](#) crea para usted no se crea en la pila CloudFormation de redes. Puede eliminarlo VPC manualmente en la consola o mediante el AWS CLI.

```
$ aws --region us-east-1 ec2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

pcluster create

Crea un nuevo clúster.

```
pcluster create [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] [ -nr ]
                [ -u TEMPLATE_URL ] [ -t CLUSTER_TEMPLATE ]
                [ -p EXTRA_PARAMETERS ] [ -g TAGS ]
                cluster_name
```

Argumentos de posición

cluster_name

Define el nombre del clúster. El nombre de la AWS CloudFormation pila esparallelcluster-**cluster_name**.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster create`.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

-r *REGION*, --region *REGION*

Especifica el Región de AWS que se va a utilizar. El orden de prioridad utilizado Región de AWS para seleccionar el clúster nuevo es el siguiente:

1. Parámetro `-r` o `--region` para [pcluster create](#).
2. `AWS_DEFAULT_REGION` variable de entorno
3. `aws_region_name` configuración en `[aws]` la sección del archivo de AWS ParallelCluster configuración (la ubicación predeterminada es `~/.parallelcluster/config`.) Esta es la ubicación actualizada por el comando [pcluster configure](#).
4. `region` configuración en `[default]` la sección del archivo de AWS CLI configuración (`~/.aws/config`.)

-nw, --nowait

Indica que no se deben esperar eventos de pila después de ejecutar un comando de pila.

El valor predeterminado es `False`.

-nr, --norollback

Deshabilita la restauración de pila al producirse un error.

El valor predeterminado es `False`.

-u *TEMPLATE_URL*, --template-url *TEMPLATE_URL*

Especifica a URL para la AWS CloudFormation plantilla personalizada si se usó cuando se creó.

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

Indica la plantilla de clúster que se va a utilizar.

-p *EXTRA_PARAMETERS*, --extra-parameters *EXTRA_PARAMETERS*

Añade parámetros adicionales para crear la pila.

-g *TAGS*, --tags *TAGS*

Especifica etiquetas adicionales que se van a añadir a la pila.

Cuando se llama al comando y comienza a sondear el estado de dicha llamada, es seguro utilizar "Ctrl-C" para salir. Puede volver a visualizar el estado actual llamando a `pcluster status mycluster`.

Ejemplos que utilizan AWS ParallelCluster la versión 2.11.7:

```
$ pcluster create mycluster
  Beginning cluster creation for cluster: mycluster
  Info: There is a newer version 3.1.4 of AWS ParallelCluster available.
  Creating stack named: parallelcluster-mycluster
  Status: ComputeFleetHITSubstack - CREATE_IN_PROGRESS
$ pcluster create mycluster --tags '{ "Key1" : "Value1" , "Key2" : "Value2" }'
```

pcluster createami

(Linux/macOS) Crea una personalización AMI para usarla. AWS ParallelCluster

```
pcluster createami [ -h ] -ai BASE_AMI_ID -os BASE_AMI_OS
  [ -i INSTANCE_TYPE ] [ -ap CUSTOM_AMI_NAME_PREFIX ]
  [ -cc CUSTOM_AMI_COOKBOOK ] [--no-public-ip]
  [ -post-install POST_INSTALL_SCRIPT ]
  [ -c CONFIG_FILE ] [-t CLUSTER_TEMPLATE]
  [--vpc-id VPC_ID] [--subnet-id SUBNET_ID]
  [ -r REGION ]
```

Dependencias requeridas

Además de AWS ParallelCluster CLI, se requiere la siguiente dependencia para ejecutarse:

`pcluster createami`

- Packer: descargue la versión más reciente de <https://developer.hashicorp.com/packer/downloads>.

Note

Antes de AWS ParallelCluster la versión 2.8.0, era necesario usar [Berkshelf](#) (instalado usando `gem install berkshelf`). `pcluster createami`

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster createami`.

-ai *BASE_AMI_ID*, --ami-id *BASE_AMI_ID*

Especifica la base AMI que se utilizará para construir el. AWS ParallelCluster AMI

-os *BASE_AMI_OS*, --os *BASE_AMI_OS*

Especifica el sistema operativo de la baseAMI. Las opciones válidas son: `alinux2`, `ubuntu1804`, `ubuntu2004` y `centos7`.

Note

El sistema operativo admite cambios en diferentes AWS ParallelCluster versiones:

- El soporte para `centos8` se eliminó en la AWS ParallelCluster versión 2.10.4.
- Se ha añadido compatibilidad con `centos8` y se ha eliminado la compatibilidad con `centos6` en AWS ParallelCluster 2.10.0.
- Support para `alinux2` se agregó en la AWS ParallelCluster versión 2.6.0.
- Support para `ubuntu1804` se agregó en AWS ParallelCluster la versión 2.5.0.

-i *INSTANCE_TYPE*, --instance-type *INSTANCE_TYPE*

Especifica el tipo de instancia que se utilizará para crear el. AMI

El valor predeterminado es `t2.xlarge`.

Note

Support for the `--instance-type` argument was added in AWS ParallelCluster version 2.4.1.

-ap *CUSTOM_AMI_NAME_PREFIX*, --ami-name-prefix *CUSTOM_AMI_NAME_PREFIX*

Especifica el nombre del prefijo del resultado. AWS ParallelCluster AMI

El valor predeterminado es `custom-ami-`.

-cc *CUSTOM_AMI_COOKBOOK*, --custom-cookbook *CUSTOM_AMI_COOKBOOK*

Especifica el libro de cocina que se utilizará para crear el. AWS ParallelCluster AMI

--post-install *POST_INSTALL_SCRIPT*

Especifica la ruta al script posterior a la instalación. Las rutas deben usar un `file://` URL esquema `s3://https://`, o. Algunos ejemplos son los siguientes:

- `https://bucket-name.s3.region.amazonaws.com/path/post_install.sh`
- `s3://bucket-name/post_install.sh`
- `file:///opt/project/post_install.sh`

Note

El soporte para el `--post-install` argumento se agregó en la AWS ParallelCluster versión 2.10.0.

--no-public-ip

No asocie una dirección IP pública a la instancia utilizada para crear el. AMI De forma predeterminada, se asocia una dirección IP pública a la instancia.

Note

El soporte para el `--no-public-ip` argumento se agregó en la AWS ParallelCluster versión 2.5.0.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

Especifica la [sección \[cluster\]](#) del *CONFIG_FILE* para usar para recuperar la configuración de la subred VPC y.

 Note

El soporte para el `--cluster-template` argumento se agregó en la AWS ParallelCluster versión 2.4.0.

--vpc-id *VPC_ID*

Especifica el ID del VPC que se va a utilizar para crear el AWS ParallelCluster AMI.

 Note

El soporte para el `--vpc-id` argumento se agregó en la AWS ParallelCluster versión 2.5.0.

--subnet-id *SUBNET_ID*

Especifica el ID de la subred que se va a utilizar para crear el. AWS ParallelCluster AMI

 Note

El soporte para el `--vpc-id` argumento se agregó en la AWS ParallelCluster versión 2.5.0.

-r *REGION*, --region *REGION*

Especifica el que se va Región de AWS a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

pcluster dcv

Interactúa con el DCV servidor de Amazon que se ejecuta en el nodo principal.

```
pcluster dcv [ -h ] ( connect )
```

pcluster dcv *command*

Opciones posibles: [connect](#)

Note

El sistema operativo admite cambios en el `pcluster dcv` comando en diferentes AWS ParallelCluster versiones:

- Se ha agregado compatibilidad para el comando `pcluster dcv` en centos8 en la versión 2.10.0 de AWS ParallelCluster .
- La compatibilidad con el `pcluster dcv` comando en instancias AWS basadas en Graviton se agregó en la AWS ParallelCluster versión 2.9.0.
- Se ha agregado compatibilidad para el comando `pcluster dcv` en ubuntu1804 en la versión 2.6.0 de AWS ParallelCluster .
- Se ha agregado compatibilidad para el comando `pcluster dcv` en centos7 en la versión 2.5.0 de AWS ParallelCluster .

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster dcv`.

Subcomandos

pcluster dcv connect

```
pcluster dcv connect [ -h ] [ -k SSH_KEY_PATH ] [ -r REGION ] cluster_name
```

⚠ Important

URL caduca 30 segundos después de su emisión. Si la conexión no se establece antes de que URL caduque, `pcluster dcv connect` vuelve a ejecutarla para generar una nueva URL.

Argumentos de posición***cluster_name***

Especifica el nombre del clúster al que hay que conectarse.

Argumentos con nombre**-h, --help**

Muestra el texto de ayuda de `pcluster dcv connect`.

-k *SSH_KEY_PATH*, --key-path *SSH_KEY_PATH*

Ruta clave de la SSH clave que se va a utilizar para la conexión.

La clave debe ser la misma que se especificó en el parámetro de configuración [key_name](#) al crear el clúster. Este argumento es opcional, pero si no se especifica, la clave debe estar disponible de forma predeterminada para el SSH cliente. Por ejemplo, agréguela al `ssh-agent` con `ssh-add`.

-r *REGION*, --region *REGION*

Especifica el Región de AWS que se va a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

-s, --show-url

Muestra una única vez URL para conectarse a la DCV sesión de Amazon. El navegador predeterminado no se abre cuando se especifica esta opción.

i Note

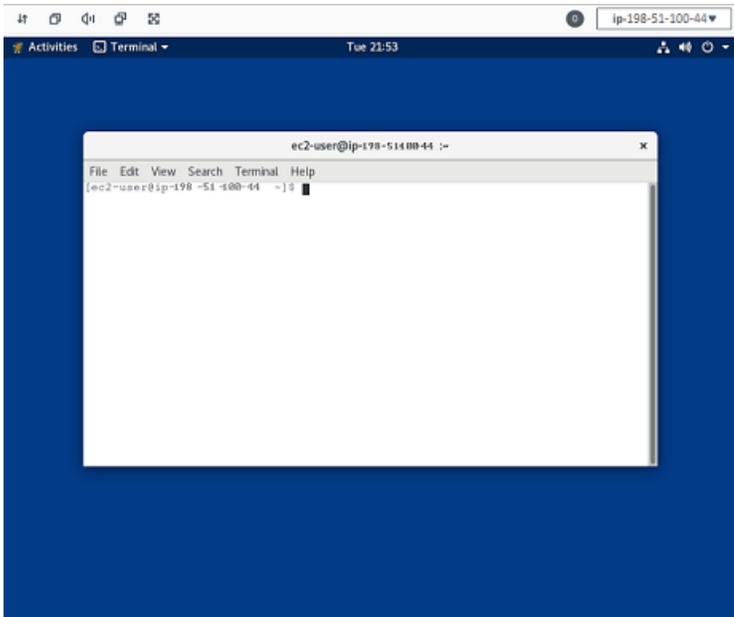
Support for the `--show-url` argument was added in AWS ParallelCluster version 2.5.1.

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

```
$ pcluster dcv connect -k ~/.ssh/id_rsa mycluster
```

Abre el navegador predeterminado para conectarse a la DCV sesión de Amazon que se ejecuta en el nodo principal.

Se crea una nueva DCV sesión de Amazon si aún no se ha iniciado ninguna.



pcluster delete

Elimina un clúster.

```
pcluster delete [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] cluster_name
```

Argumentos de posición

cluster_name

Especifica el nombre del clúster que se va a eliminar.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster delete`.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

--keep-logs

Conserve los datos de los CloudWatch registros después de eliminar el clúster. El grupo de registros permanece hasta que se elimina manualmente, pero los eventos del registro caducarán según el valor de [retention_days](#). El valor predeterminado es 14 días.

Note

Se ha agregado compatibilidad para el argumento **--keep-logs** en la versión 2.6.0 de AWS ParallelCluster .

-r *REGION*, --region *REGION*

Especifica el Región de AWS que se va a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

Cuando se llama al comando y comienza a sondear el estado de dicha llamada, es seguro utilizar "Ctrl-C" para salir. Puede volver a visualizar el estado actual llamando a `pcluster status mycluster`.

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

```
$ pcluster delete -c path/to/config -r us-east-1 mycluster
Deleting: mycluster
Status: RootRole - DELETE_COMPLETE
Cluster deleted successfully.
```

Para eliminar los recursos de red deVPC, puede eliminar la pila de CloudFormation redes. El nombre de la pila comienza con «parallelclusternetworking-» y contiene la hora de creación en formato «YYYYMMDDHHMMSS». Puede enumerar las pilas mediante el comando [list-stacks](#).

```
$ aws --region us-east-1 cloudformation list-stacks \
  --stack-status-filter "CREATE_COMPLETE" \
  --query "StackSummaries[].StackName" | \
  grep -e "parallelclusternetworking-"
  "parallelclusternetworking-pubpriv-20191029205804"
```

La pila se puede eliminar mediante el comando [delete-stack](#).

```
$ aws --region us-east-1 cloudformation delete-stack \
  --stack-name parallelclusternetworking-pubpriv-20191029205804
```

Lo VPC que [pcluster configure](#) crea para usted no se crea en la pila CloudFormation de redes. Puede eliminarlo VPC manualmente en la consola o mediante el AWS CLI.

```
$ aws --region us-east-1 ec2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

pcluster instances

Muestra una lista de todas las instancias de un clúster.

```
pcluster instances [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

Argumentos de posición

nombre_clúster

Muestra las instancias para el clúster con el nombre proporcionado.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster instances`.

-c CONFIG_FILE, --config CONFIG_FILE

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/parallelcluster/config`.

-r REGION, --region REGION

Especifica el que se va a utilizar Región de AWS . El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

```
$ pcluster instances -c path/to/config -r us-east-1 mycluster
MasterServer      i-1234567890abcdef0
ComputeFleet      i-abcdef01234567890
```

pcluster list

Muestra una lista de las pilas a las que están asociadas. AWS ParallelCluster

```
pcluster list [ -h ] [ -c CONFIG_FILE ] [ -r REGION ]
```

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster list`.

--color

Muestra el estado del clúster en color.

El valor predeterminado es `False`.

-c CONFIG_FILE, --config CONFIG_FILE

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `c`.

-r REGION, --region REGION

Especifica el que se va Región de AWS a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

Muestra el nombre de todas las AWS CloudFormation pilas nombradas. `parallelcluster-*`

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

```
$ pcluster list -c path/to/config -r us-east-1
mycluster          CREATE_IN_PROGRESS  2.11.7
myothercluster    CREATE_IN_PROGRESS  2.11.7
```

pcluster ssh

Ejecuta un comando ssh con el nombre de usuario y la dirección IP del clúster completados previamente. Se adjuntan argumentos arbitrarios al final del comando ssh. Este comando se puede personalizar en la sección de alias del archivo de configuración.

```
pcluster ssh [ -h ] [ -d ] [ -r REGION ] cluster_name
```

Argumentos de posición

cluster_name

Especifica el nombre del clúster al que hay que conectarse.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster ssh`.

-d, --dryrun

Muestra el comando que se va a ejecutar y termina.

El valor predeterminado es `False`.

-r *REGION*, --region *REGION*

Especifica el que se va Región de AWS a utilizar. El valor predeterminado es la región especificada mediante el comando [pcluster configure](#).

Ejemplos que utilizan AWS ParallelCluster la versión 2.11.7:

```
$ pcluster ssh -d mycluster -i ~/.ssh/id_rsa
SSH command: ssh ec2-user@1.1.1.1 -i /home/user/.ssh/id_rsa
```

```
$ pcluster ssh mycluster -i ~/.ssh/id_rsa
```

Ejecuta un comando ssh con el nombre de usuario y la dirección IP del clúster completados previamente:

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

El comando ssh se define en el archivo de configuración global en [Sección \[alias\]](#). Se puede personalizar tal y como se indica a continuación.

```
[ alias ]  
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

Variables sustituidas:

CFN_USER

El nombre de usuario para la instancia de [base_os](#) que se ha seleccionado.

MASTER_IP

La dirección IP del nodo maestro.

ARGS

Argumentos opcionales que se pasan al comando ssh.

pcluster start

Comienza la flota de computación de un clúster que se ha detenido.

```
pcluster start [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

Argumentos de posición

cluster_name

Comienza la flota de computación del nombre de clúster proporcionado.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster start`.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

-r *REGION*, --region *REGION*

Especifica el que se va Región de AWS a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

```
$ pcluster start mycluster
```

```
Compute fleet status is: RUNNING. Submitting status change request.
```

```
Request submitted successfully. It might take a while for the transition to complete.
```

```
Please run 'pcluster status' if you need to check compute fleet status
```

Este comando establece los parámetros del grupo de escalado automático en uno de los siguientes valores:

- Los valores de configuración inicial (`max_queue_size` y `initial_queue_size`) de la plantilla que se utilizó para crear el clúster.
- Los valores de configuración que se utilizaron para actualizar el clúster desde que se creó por primera vez.

pcluster status

Extrae el estado actual del clúster.

```
pcluster status [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] cluster_name
```

Argumentos de posición

cluster_name

Muestra el estado del clúster con el nombre proporcionado.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster status`.

-c CONFIG_FILE, --config CONFIG_FILE

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

-r REGION, --region REGION

Especifica el que se va Región de AWS a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

-nw, --nowait

Indica que no se deben esperar eventos de pila después de ejecutar un comando de pila.

El valor predeterminado es `False`.

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

```
$ pcluster status -c path/to/config -r us-east-1 mycluster
Status: ComputeFleetHITSubstack - CREATE_IN_PROGRESS
```

pcluster stop

Detiene la flota de computación y deja el nodo maestro en ejecución.

```
pcluster stop [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

Argumentos de posición

cluster_name

Detiene la flota de computación del nombre de clúster proporcionado.

Ejemplo que utiliza la AWS ParallelCluster versión 2.11.7:

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster stop`.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

-r *REGION*, --region *REGION*

Especifica el que se va Región de AWS a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

```
$ pcluster stop mycluster
```

```
Compute fleet status is: STOPPED. Submitting status change request.  
Request submitted successfully. It might take a while for the transition to complete.  
Please run 'pcluster status' if you need to check compute fleet status
```

Establece los parámetros del grupo de escalado automático en mínimo/máximo/deseado = 0/0/0 y termina la flota de computación. La cabeza permanece en el aire. Para cancelar todos los EC2 recursos y evitar EC2 cargos, considere la posibilidad de eliminar el clúster.

pcluster update

Analiza el archivo de configuración para determinar si el clúster se puede actualizar de forma segura. Si el análisis determina que el clúster se puede actualizar, se le solicitará que confirme el cambio. Si el análisis muestra que el clúster no se puede actualizar, los ajustes de configuración que son el origen de los conflictos se enumeran con detalles. Para obtener más información, consulte [Uso de pcluster update](#).

```
pcluster update [ -h ] [ -c CONFIG_FILE ] [ --force ] [ -r REGION ] [ -nr ]  
                [ -nw ] [ -t CLUSTER_TEMPLATE ] [ -p EXTRA_PARAMETERS ] [ -rd ]  
                [ --yes ] cluster_name
```

Argumentos de posición

cluster_name

Especifica el nombre del clúster que se va a actualizar.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster update`.

-c *CONFIG_FILE*, --config *CONFIG_FILE*

Especifica el archivo de configuración alternativo que se va a utilizar.

El valor predeterminado es `~/.parallelcluster/config`.

--force

Habilita una actualización incluso si una o más configuraciones tienen un cambio bloqueante o si es necesaria una acción pendiente (como detener la flota de computación) antes de que la actualización pueda continuar. Esto no debe combinarse con el argumento `--yes`.

-r *REGION*, --region *REGION*

Especifica el Región de AWS que se va a utilizar. El valor predeterminado es el Región de AWS especificado mediante el [pcluster configure](#) comando.

-nr, --norollback

Desactiva la reversión de la AWS CloudFormation pila en caso de error.

El valor predeterminado es `False`.

-nw, --nowait

Indica que no se deben esperar eventos de pila después de ejecutar un comando de pila.

El valor predeterminado es `False`.

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

Especifica la sección de la plantilla del clúster que se va a utilizar.

-p *EXTRA_PARAMETERS*, --extra-parameters *EXTRA_PARAMETERS*

Añade parámetros adicionales para actualizar una pila.

-rd, --reset-desired

Restablece la capacidad actual de un grupo de Auto Scaling a los valores de configuración inicial.

El valor predeterminado es False.

--yes

Supone automáticamente que la respuesta a todas las peticiones es afirmativa. Esto no debe combinarse con el argumento `--force`.

```
$ pcluster update -c path/to/config mycluster
Retrieving configuration from CloudFormation for cluster mycluster...
Validating configuration file .parallelcluster/config...
Found Configuration Changes:

#      parameter                old value  new value
---  -
      [compute_resource default]
01   min_count                  1          2
02   max_count                  5          12

Validating configuration update...
Congratulations! The new configuration can be safely applied to your cluster.
Do you want to proceed with the update? - Y/N: Y
Updating: mycluster
Calling update_stack
Status: parallelcluster-mycluster - UPDATE_COMPLETE
```

Cuando se llama al comando y comienza a sondear el estado de dicha llamada, es seguro utilizar "Ctrl-C" para salir. Puede volver a visualizar el estado actual llamando a `pcluster status mycluster`.

pcluster version

Muestra la versión. AWS ParallelCluster

```
pcluster version [ -h ]
```

Para obtener información de marcadores específicos de comandos, ejecute: `pcluster [command] --help`.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster version`.

Cuando se llama al comando y comienza a sondear el estado de dicha llamada, es seguro utilizar "Ctrl-C" para salir. Puede volver a visualizar el estado actual llamando a `pcluster status mycluster`.

```
$ pcluster version  
2.11.7
```

pcluster-config

Actualiza el archivo AWS ParallelCluster de configuración.

```
pcluster-config [ -h ] [convert]
```

Para obtener información de marcadores específicos de comandos, ejecute: `pcluster-config [command] -h`.

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster-config`.

Note

El `pcluster-config` comando se agregó en la AWS ParallelCluster versión 2.9.0.

Subcomandos

pcluster-config convert

```
pcluster-config convert [ -h ] [ -c CONFIG_FILE ] [ -t CLUSTER_TEMPLATE ]  
                        [ -o OUTPUT_FILE ]
```

Argumentos con nombre

-h, --help

Muestra el texto de ayuda de `pcluster-config convert`.

-c *CONFIG_FILE*, --config-file *CONFIG_FILE*

Especifica la ruta del archivo de configuración.

El valor predeterminado es `~/.parallelcluster/config`.

Para obtener más información, consulte [Configuración AWS ParallelCluster](#).

-t *CLUSTER_TEMPLATE*, --cluster-template *CLUSTER_TEMPLATE*

Indica el [Sección de \[cluster\]](#) que se va a utilizar. Si no se especifica este argumento, `pcluster-config convert` utilizará la `cluster_template` configuración de [Sección \[global\]](#). Si no se especifica, se utiliza la sección de `[cluster default]`.

-o *OUTPUT_FILE*, --output *OUTPUT_FILE*

Especifica la ruta del archivo de configuración convertido que se va a escribir. De forma predeterminada, la salida se escribe en STDOUT.

Ejemplo:

```
$ pcluster-config convert -t alpha -o ~/.parallelcluster/multiinstance
```

Convierte la configuración del clúster especificada en la sección de `[cluster alpha]` de `~/.parallelcluster/config`, escribiendo el archivo de configuración convertido en `~/.parallelcluster/multiinstance`.

Configuración

De forma predeterminada, AWS ParallelCluster utiliza el archivo `~/.parallelcluster/config` para todos los parámetros de configuración. Puede especificar un archivo de configuración personalizado mediante la opción de la línea de comandos `-c` o `--config`, o la variable de entorno `AWS_PCLUSTER_CONFIG_FILE`.

Encontrará un archivo de configuración de ejemplo instalado con AWS ParallelCluster, en el directorio Python, en `site-packages/aws-parallelcluster/examples/config`. El archivo de configuración de ejemplo también está disponible en GitHub, en <https://github.com/aws/aws-parallelcluster/blob/v2.11.9/cli/src/pcluster/examples/config>.

Versión actual AWS ParallelCluster 2: 2.11.9.

Temas

- [Diseño](#)
- [Sección \[global\]](#)
- [Sección \[aws\]](#)
- [Sección \[alias\]](#)
- [Sección de \[cluster\]](#)
- [Sección \[compute_resource\]](#)
- [Sección \[cw_log\]](#)
- [Sección \[dashboard\]](#)
- [Sección de \[dcv\]](#)
- [Sección de \[ebs\]](#)
- [Sección \[efs\]](#)
- [Sección \[fsx\]](#)
- [Sección de \[queue\]](#)
- [Sección de \[raid\]](#)
- [Sección \[scaling\]](#)
- [Sección de \[vpc\]](#)
- [Ejemplos](#)

Diseño

Una configuración AWS ParallelCluster se define en varias secciones.

Las siguientes secciones son obligatorias: [sección \[global\]](#) y [sección \[aws\]](#).

También debe incluir al menos una [sección \[cluster\]](#) y una [sección \[vpc\]](#).

Las secciones comienzan con el nombre de sección entre corchetes, seguido de los parámetros y la configuración.

```
[global]
cluster_template = default
update_check = true
sanity_check = true
```

Sección [global]

Especifica opciones de configuración globales relacionadas con `pcluster`.

```
[global]
```

Temas

- [cluster_template](#)
- [update_check](#)
- [sanity_check](#)

cluster_template

Define el nombre de la sección `cluster` que se usa para el clúster de forma predeterminada. Para obtener más información sobre las secciones `cluster`, consulte la [sección \[cluster\]](#). El nombre del clúster debe comenzar por una letra, contener no más de 60 caracteres y solo letras, números y guiones (-).

Por ejemplo, la siguiente configuración especifica que la sección que comienza `[cluster default]` se utilice de forma predeterminada.

```
cluster_template = default
```

Política de actualización: esta configuración no se analiza durante una actualización.

update_check

(Opcional) Comprueba si hay actualizaciones de `pcluster`.

El valor predeterminado es `true`.

```
update_check = true
```

Política de actualización: esta configuración no se analiza durante una actualización.

sanity_check

(Opcional) Se usa para validar la configuración de los recursos que se definen en los parámetros de clúster.

El valor predeterminado es `true`.

Warning

Si `sanity_check` se establece en `false`, se omiten comprobaciones importantes. Esto podría provocar que la configuración no funcione según lo previsto.

```
sanity_check = true
```

Note

Antes de AWS ParallelCluster versión 2.5.0, el valor predeterminado de `sanity_check` era `false`.

Política de actualización: esta configuración no se analiza durante una actualización.

Sección [aws]

(Opcional) Se utiliza para seleccionar el Región de AWS.

La creación de clústeres utiliza este orden de prioridad para seleccionar el Región de AWS para un clúster nuevo:

1. Parámetro `-r` o `--region` para [pcluster create](#).
2. `AWS_DEFAULT_REGION` variable de entorno
3. Configuración de `aws_region_name` en la sección `[aws]` del archivo de configuración AWS ParallelCluster (la ubicación predeterminada es `~/.parallelcluster/config`). Esta es la ubicación actualizada por el comando [pcluster configure](#).
4. Configuración de `region` en la sección `[default]` del archivo de configuración AWS CLI (`~/.aws/config`.)

Note

Antes de AWS ParallelCluster versión 2.10.0, estos ajustes eran obligatorios y se aplicaban a todos los clústeres.

Para almacenar las credenciales, puede utilizar el entorno, roles de IAM para Amazon EC2 o la [AWS CLI](#), en lugar de guardar las credenciales en el archivo de configuración de AWS ParallelCluster.

```
[aws]
aws_region_name = Region
```

[Política de actualización: esta configuración no se analiza durante una actualización.](#)

Sección **[aliases]**

Especifica los alias y le permite personalizar el comando `ssh`.

Tenga en cuenta la configuración predeterminada siguiente:

- `CFN_USER` se establece en el nombre de usuario predeterminado para el sistema operativo
- `MASTER_IP` se establece en la dirección IP del nodo principal
- `ARGS` se establece con los argumentos que el usuario proporcione después de `pcluster ssh cluster_name`

```
[aliases]
# This is the aliases section, you can configure
# ssh alias here
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

[Política de actualización: esta configuración no se analiza durante una actualización.](#)

Sección de `[cluster]`

Define una plantilla de clúster que se puede utilizar para crear un clúster. Un archivo de configuración puede contener varias secciones de `[cluster]`.

Se puede usar la misma plantilla de clúster para crear varios clústeres.

El formato es `[cluster cluster-template-name]`. La [sección de `\[cluster\]`](#) nombrada por la configuración de `cluster_template` de la [sección de `\[global\]`](#) se usa de forma predeterminada, pero se puede anular en la línea de comandos de `pcluster`.

cluster-template-name debe empezar por una letra, no contener más de 30 caracteres y contener únicamente letras, números, guiones (-) y guiones bajos (_).

```
[cluster default]
```

Temas

- [additional_cfn_template](#)
- [additional_iam_policies](#)
- [base_os](#)
- [cluster_resource_bucket](#)
- [cluster_type](#)
- [compute_instance_type](#)
- [compute_root_volume_size](#)
- [custom_ami](#)
- [cw_log_settings](#)
- [dashboard_settings](#)
- [dcv_settings](#)

- [desired_vcpus](#)
- [disable_cluster_dns](#)
- [disable_hyperthreading](#)
- [ebs_settings](#)
- [ec2_iam_role](#)
- [efs_settings](#)
- [enable_efa](#)
- [enable_efa_gdr](#)
- [enable_intel_hpc_platform](#)
- [encrypted_ephemeral](#)
- [ephemeral_dir](#)
- [extra_json](#)
- [fsx_settings](#)
- [iam_lambda_role](#)
- [initial_queue_size](#)
- [key_name](#)
- [maintain_initial_size](#)
- [master_instance_type](#)
- [master_root_volume_size](#)
- [max_queue_size](#)
- [max_vcpus](#)
- [min_vcpus](#)
- [placement](#)
- [placement_group](#)
- [post_install](#)
- [post_install_args](#)
- [pre_install](#)
- [pre_install_args](#)
- [proxy_server](#)

- [queue_settings](#)
- [raid_settings](#)
- [s3_read_resource](#)
- [s3_read_write_resource](#)
- [scaling_settings](#)
- [scheduler](#)
- [shared_dir](#)
- [spot_bid_percentage](#)
- [spot_price](#)
- [tags](#)
- [template_url](#)
- [vpc_settings](#)

additional_cfn_template

(Opcional) Define una AWS CloudFormation plantilla adicional para lanzarla junto con el clúster. Esta plantilla adicional se utiliza para la creación de recursos que existen fuera del clúster pero que forman parte del ciclo de vida de este.

El valor debe corresponder HTTP URL a una plantilla pública y debe proporcionarse todos los parámetros.

No hay valor predeterminado.

```
additional_cfn_template = https://<bucket-name>.s3.amazonaws.com/my-cfn-template.yaml
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

additional_iam_policies

(Opcional) Especifica una lista de nombres de recursos de Amazon (ARNs) de IAM políticas para AmazonEC2. Esta lista se asocia al rol raíz utilizado en el clúster, además de los permisos requeridos por AWS ParallelCluster separados por comas. El nombre IAM de una política y el suyo ARN son diferentes. Los nombres no se pueden utilizar como argumento para `additional_iam_policies`.

Si su intención es añadir políticas adicionales a la configuración predeterminada de los nodos del clúster, le recomendamos que incorpore las IAM políticas personalizadas adicionales con la `additional_iam_policies` [ec2_iam_role](#) configuración en lugar de utilizarla para añadir EC2 políticas específicas. Esto se debe a `additional_iam_policies` que se añaden a los permisos predeterminados que se AWS ParallelCluster requieren. Un [ec2_iam_role](#) existente debe incluir todos los permisos necesarios. Sin embargo, dado que los permisos necesarios suelen cambiar de una versión a otra a medida que se añaden características, un [ec2_iam_role](#) existente puede quedar obsoleto.

No hay valor predeterminado.

```
additional_iam_policies = arn:aws:iam::123456789012:policy/CustomEC2Policy
```

Note

Se ha agregado compatibilidad para [additional_iam_policies](#) en la versión 2.5.0 de AWS ParallelCluster .

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

base_os

(Obligatorio) Especifica qué tipo de sistema operativo se utiliza en el clúster.

Las opciones disponibles son:

- `alinux2`
- `centos7`
- `ubuntu1804`
- `ubuntu2004`

Note

Solo para instancias AWS basadas en Graviton `alinux2ubuntu1804`, o `ubuntu2004` son compatibles.

Note

El soporte para centos8 se eliminó en la AWS ParallelCluster versión 2.11.4. Se ha agregado compatibilidad para ubuntu2004 y se ha eliminado la compatibilidad con alinux y ubuntu1604 en la versión 2.11.0 de AWS ParallelCluster . Se agregó soporte para y se centos8 centos6 eliminó el soporte para en la AWS ParallelCluster versión 2.10.0. Se ha agregado compatibilidad para alinux2 en la versión 2.6.0 de AWS ParallelCluster . Se ha agregado compatibilidad para ubuntu1804 y se ha eliminado la compatibilidad con ubuntu1404 en la versión 2.5.0 de AWS ParallelCluster .

Aparte de las específicas Regiones de AWS mencionadas en la siguiente tabla que no son compatibles. centos7 Todas las demás regiones AWS comerciales son compatibles con los siguientes sistemas operativos.

| Partición (Regiones de AWS) | alinux2 | centos7 | ubuntu1804 y ubuntu2004 |
|---|---------|---------|-------------------------|
| Comercial (Regiones de AWS ninguno de ellos se menciona específicamente) | True | True | True |
| AWS GovCloud (Este de EE. UU.) (us-gov-east-1) | True | False | True |
| AWS GovCloud (EEUU-Oeste) () us-gov-west-1 | True | False | True |
| China (Pekín) (cn-north-1) | True | False | True |
| China (Ningxia) (cn-northwest-1) | True | False | True |

Note

Nota: El parámetro [base_os](#) también determina el nombre de usuario que se utiliza para iniciar sesión en el clúster.

- centos7: centos
- ubuntu1804 y ubuntu2004: ubuntu
- alinux2: ec2-user

 Note

Antes de AWS ParallelCluster la versión 2.7.0, el [base_os](#) parámetro era opcional y el predeterminado era. alinux A partir de la versión 2.7.0 de AWS ParallelCluster , el parámetro [base_os](#) es obligatorio.

 Note

Si el parámetro [scheduler](#) es awsbatch, solo se admite alinux2.

```
base_os = alinux2
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

cluster_resource_bucket

(Opcional) Especifica el nombre del bucket de Amazon S3 que se utiliza para alojar los recursos que se generan al crear el clúster. El bucket debe tener habilitado el control de versiones. Para obtener más información, consulte [Uso del control de versiones](#) en la Guía del usuario de Amazon Simple Storage Service. Este bucket se puede utilizar para varios clústeres. El bucket debe estar en la misma región que el clúster.

Si no se especifica este parámetro, se crea un bucket nuevo al crear el clúster. El nuevo bucket tiene el nombre de `parallelcluster-random_string`. En este nombre, *random_string* es una cadena aleatoria de caracteres alfanuméricos. Todos los recursos del clúster se almacenan en este depósito en una ruta con el formulario `bucket_name/resource_directory`. `resource_directory` tiene el formulario `stack_name-random_string`, donde *stack_name* es el nombre de una de las AWS CloudFormation pilas utilizadas por AWS ParallelCluster. El valor de *bucket_name* se encuentra en el ResourcesS3Bucket valor de la salida de la `parallelcluster-clustername` pila. El valor de *resource_directory* se encuentra en el valor de la ArtifactS3RootDirectory salida de la misma pila.

El valor predeterminado es `parallelcluster-random_string`.

```
cluster_resource_bucket = amzn-s3-demo-bucket
```

Note

El soporte para [cluster_resource_bucket](#) se agregó en la AWS ParallelCluster versión 2.10.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización. No se puede forzar la actualización de esta configuración.

cluster_type

(Opcional) Define el tipo de clúster que se va a lanzar. Si se define la configuración de [queue_settings](#), esta configuración debe reemplazarse por la configuración de [compute_type](#) en las [secciones de \[queue\]](#).

Las opciones válidas son: `ondemand` y `spot`.

El valor predeterminado es `ondemand`.

Para obtener más información acerca de las instancias de Spot, consulte [Uso de instancias de spot](#).

Note

El uso de instancias de spot requiere que el rol de `AWSServiceRoleForEC2Spot` vinculado al servicio esté en su cuenta. Para crear este rol en su cuenta mediante el AWS CLI, ejecute el siguiente comando:

```
aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Para obtener más información, consulte [Función vinculada a servicios para solicitudes de instancias puntuales](#) en la Guía EC2 del usuario de Amazon.

```
cluster_type = ondemand
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

compute_instance_type

(Opcional) Define el tipo de EC2 instancia de Amazon que se usa para los nodos de cómputo del clúster. La arquitectura del tipo de instancia debe ser la misma que la arquitectura utilizada para la configuración de [master_instance_type](#). Si se define la configuración de [queue_settings](#), esta configuración debe reemplazarse por la configuración de [instance_type](#) en las [secciones de \[compute_resource\]](#).

Si utilizas el `awsbatch` programador, consulta la sección Creación de entornos de cómputo en la AWS Batch interfaz de usuario para ver una lista de los tipos de instancias compatibles.

El valor predeterminado es `t2.micro`, `optimal` cuando el programador es `awsbatch`.

```
compute_instance_type = t2.micro
```

Note

En la AWS ParallelCluster versión 2.8.0 se agregó soporte para instancias AWS basadas en Graviton (incluidas C6g las instancias A1 Y).

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

compute_root_volume_size

(Opcional) Especifica el tamaño del volumen ComputeFleet raíz en gibibytes (GiB). El AMI soporte imprescindible. `growroot`

El valor predeterminado es 35.

Note

Para AWS ParallelCluster las versiones entre 2.5.0 y 2.10.4, el valor predeterminado era 25. Antes de AWS ParallelCluster la versión 2.5.0, el valor predeterminado era 20.

```
compute_root_volume_size = 35
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

custom_ami

(Opcional) Especifica el ID de una personalización AMI que se va a utilizar en los nodos principal y de cómputo en lugar del que se [publica AMIs](#) de forma predeterminada. Para obtener más información, consulte [Modificar una AMI de](#) o [Creación de una AMI de AWS ParallelCluster personalizada](#).

No hay valor predeterminado.

```
custom_ami = ami-00d4efc81188687a0
```

Si la personalización AMI requiere permisos adicionales para su lanzamiento, estos permisos se deben agregar a las políticas de usuario y del nodo principal.

Por ejemplo, si una personalización AMI tiene asociada una instantánea cifrada, se requieren las siguientes políticas adicionales tanto en las políticas de usuario como en las del nodo principal:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>;key/<AWS_KMS_KEY_ID>"
      ]
    }
  ]
}
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

cw_log_settings

(Opcional) Identifica la [cw_log] sección con la configuración de CloudWatch registros. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección de \[cw_log\]](#), [CloudWatch Panel de control de Amazon](#) y [Integración con Amazon CloudWatch Logs](#).

Por ejemplo, la siguiente configuración especifica que la sección que se inicia [cw_log custom-cw] se usa para la configuración de CloudWatch registros.

```
cw_log_settings = custom-cw
```

Note

El soporte para [cw_log_settings](#) se agregó en la AWS ParallelCluster versión 2.6.0.

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

dashboard_settings

(Opcional) Identifica la [dashboard] sección con la configuración del CloudWatch cuadro de mandos. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[dashboard\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que se inicia [dashboard custom-dashboard] se usa para la configuración del CloudWatch panel.

```
dashboard_settings = custom-dashboard
```

Note

El soporte para [dashboard_settings](#) se agregó en la AWS ParallelCluster versión 2.10.0.

Política de actualización: esta configuración se puede cambiar durante una actualización.

dcv_settings

(Opcional) Identifica la [dcv] sección con la DCV configuración de Amazon. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[dcv\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que comienza [dcv custom-dcv] se usa para la DCV configuración de Amazon.

```
dcv_settings = custom-dcv
```

Note

En las instancias AWS basadas en Graviton, Amazon solo DCV es compatible con. `alinux2`

Note

El soporte para [dcv_settings](#) se agregó en la AWS ParallelCluster versión 2.5.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

desired_vcpus

(Opcional) Especifica el número deseado de vCPUs en el entorno informático. Solo se utiliza si el programador es `awsbatch`.

El valor predeterminado es 4.

```
desired_vcpus = 4
```

Política de actualización: esta configuración no se analiza durante una actualización.

disable_cluster_dns

(Opcional) Especifica si no se deben crear las DNS entradas del clúster. De forma predeterminada, AWS ParallelCluster crea una zona alojada en Route 53. Si `disable_cluster_dns` se establece en `true`, no se crea la zona alojada.

El valor predeterminado es `false`.

```
disable_cluster_dns = true
```

Warning

Se requiere un sistema de resolución de nombres para que el clúster funcione correctamente. Si `disable_cluster_dns` está establecido en `true`, también se debe proporcionar un sistema de resolución de nombres adicional.

Important

[disable_cluster_dns](#) = `true` solo se admite si se especifica la configuración de [queue_settings](#).

Note

El soporte para [disable_cluster_dns](#) se agregó en la AWS ParallelCluster versión 2.9.1.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

disable_hyperthreading

Deshabilita la tecnología Hyper-Threading en los nodos principal y de computación. No todos los tipos de instancias pueden deshabilitar la tecnología Hyper-Threading. Para obtener una lista de los tipos de instancias que admiten la desactivación del hiperproceso, consulta los [CPU núcleos y subprocesos de cada CPU núcleo para cada tipo de instancia](#) en la Guía del usuario de Amazon EC2. Si se define la configuración de [queue_settings](#), se puede definir esta configuración o se puede definir la configuración de [disable_hyperthreading](#) en las [secciones de \[queue\]](#).

El valor predeterminado es `false`.

```
disable_hyperthreading = true
```

Note

[disable_hyperthreading](#) solo afecta al nodo principal cuando [scheduler](#) = `awsbatch`.

Note

Se ha agregado compatibilidad para [disable_hyperthreading](#) en la versión 2.5.0 de AWS ParallelCluster .

Política de actualización: si se cambia esta configuración, no se permite la actualización.

ebs_settings

(Opcional) Identifica las `[ebs]` secciones con los EBS volúmenes de Amazon que están montados en el nodo principal. Si utilizas varios EBS volúmenes de Amazon, introduce estos parámetros en una lista con cada uno separado por una coma. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Se admiten hasta cinco (5) EBS volúmenes de Amazon adicionales.

Para obtener más información, consulte la [sección \[ebs\]](#).

Por ejemplo, la siguiente configuración especifica las secciones que comienzan `[ebs custom1]` y se `[ebs custom2]` utilizan para los EBS volúmenes de Amazon.

```
ebs_settings = custom1, custom2
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

ec2_iam_role

(Opcional) Define el nombre de un IAM rol existente para Amazon EC2 que está adjunto a todas las instancias del clúster. El nombre de un IAM rol y su nombre de recurso de Amazon (ARN) son diferentes. ARNsno se puede usar como argumento para `ec2_iam_role`.

Si se especifica esta opción, se omite el valor [additional_iam_policies](#). Si su intención es añadir políticas adicionales a la configuración predeterminada de los nodos del clúster, le recomendamos que incorpore las IAM políticas personalizadas adicionales con la [additional_iam_policies](#) configuración en lugar de `ec2_iam_role` utilizarla.

Si no se especifica esta opción, se EC2 utiliza el AWS ParallelCluster IAM rol predeterminado de Amazon. Para obtener más información, consulte [AWS Identity and Access Management funciones en AWS ParallelCluster](#).

No hay valor predeterminado.

```
ec2_iam_role = ParallelClusterInstanceRole
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

efs_settings

(Opcional) Especifica la configuración relacionada con el sistema de EFS archivos de Amazon. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[efs\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que comienza `[efs customfs]` se usa para la configuración del sistema de EFS archivos de Amazon.

```
efs_settings = customfs
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

enable_efa

(Opcional) Si está presente, especifica que el adaptador Elastic Fabric (EFA) está habilitado para los nodos de cómputo. Para ver la lista de EC2 instancias compatiblesEFA, consulta los [tipos de instancias compatibles](#) en la Guía del EC2 usuario de Amazon para instancias de Linux.

Para obtener más información, consulte [Elastic Fabric Adapter](#). Si se define la configuración de [queue_settings](#), se puede definir esta configuración o se puede definir la configuración de [enable_efa](#) en la [sección de \[queue\]](#). Se debe utilizar un grupo de ubicación del clúster para minimizar las latencias entre instancias. Para obtener más información, consulte [placement](#) y [placement_group](#).

```
enable_efa = compute
```

Note

EFA En la versión 2.10.1 se agregó soporte para instancias Graviton2 basadas en AWS ParallelCluster ARM.

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

enable_efa_gdr

(Opcional) A partir de la AWS ParallelCluster versión 2.11.3, esta configuración no tiene efecto. La compatibilidad con Elastic Fabric Adapter GPUDirect RDMA (EFA) para (acceso remoto directo a la memoria) siempre está habilitada si es compatible tanto con el tipo de instancia como con el sistema operativo.

Note

AWS ParallelCluster de la versión 2.10.0 a la 2.11.2: Si `compute`, especifica que la compatibilidad con el Elastic Fabric Adapter (EFA) para GPUDirect RDMA (acceso remoto directo a la memoria) está habilitada para los nodos de cómputo. Si se establece esta configuración en `compute`, es necesario que la configuración de [enable_efa](#) esté establecida en `compute`. EFA GPUDirect RDMA La compatibilidad con es compatible con tipos de instancias específicos (`p4d.24xlarge`) en sistemas operativos específicos ([base_os](#) `esalinux2`, `centos7`, `ubuntu1804` o). `ubuntu2004` Si se define la configuración de [queue_settings](#), se puede definir esta configuración o se puede definir la configuración de [enable_efa_gdr](#) en las [secciones de \[queue\]](#). Se debe utilizar un grupo de ubicación del clúster para minimizar las latencias entre instancias. Para obtener más información, consulte [placement](#) y [placement_group](#).

```
enable_efa_gdr = compute
```

Note

El soporte para `enable_efa_gdr` se agregó en la AWS ParallelCluster versión 2.10.0.

Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.

`enable_intel_hpc_platform`

Si existe, indica que se acepta el Acuerdo de licencia de usuario final para Intel Parallel Studio. Esto hace que Intel Parallel Studio se instale en el nodo maestro y se comparta con los nodos de computación. Esto añade varios minutos al tiempo que tarda el nodo principal en arrancar. La `enable_intel_hpc_platform` configuración solo se admite en CentOS (7 `base_os = centos7`).

El valor predeterminado es `false`.

```
enable_intel_hpc_platform = true
```

Note

El `enable_intel_hpc_platform` parámetro no es compatible con las instancias AWS basadas en Graviton.

Note

Se ha agregado compatibilidad para `enable_intel_hpc_platform` en la versión 2.5.0 de AWS ParallelCluster .

Política de actualización: si se cambia esta configuración, no se permite la actualización.

encrypted_ephemeral

(Opcional) Cifra los volúmenes del almacén de instancias efímeras con claves en memoria no recuperables mediante la configuración de claves unificadas de Linux. LUKS

Para obtener más información, consulte <https://gitlab.com/cryptsetup/cryptsetup/blob/master/README.md>.

El valor predeterminado es `false`.

```
encrypted_ephemeral = true
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

ephemeral_dir

Define la ruta donde se montan los volúmenes de almacenes de instancias, si se usan.

El valor predeterminado es `/scratch`.

```
ephemeral_dir = /scratch
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

extra_json

(Opcional) Define el extra que se fusiona en JSON Chef `dna.json`. Para obtener más información, consulte [Creación de una AMI de AWS ParallelCluster personalizada](#).

El valor predeterminado es `{}`.

```
extra_json = {}
```

Note

A partir de AWS ParallelCluster la versión 2.6.1, la mayoría de las recetas de instalación se omiten de forma predeterminada al lanzar los nodos para mejorar los tiempos de inicio. Para ejecutar todas las recetas de instalación para una mejor compatibilidad con versiones anteriores a costa de los tiempos de inicio, añade `"skip_install_recipes" : "no"` a la clave `cluster` en la configuración [extra_json](#). Por ejemplo:

```
extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

fsx_settings

(Opcional) Especifica la sección que define la configuración FSx de Lustre. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[fsx\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que comienza [fsx fs] se usa FSx para la configuración de Lustre.

```
fsx_settings = fs
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

iam_lambda_role

(Opcional) Define el nombre de un rol de AWS Lambda ejecución existente. Este rol está asociado a todas las funciones de Lambda en el clúster. Para obtener más información, consulte [Rol de ejecución de AWS Lambda](#) en la Guía para desarrolladores de AWS Lambda .

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

El nombre de un IAM rol y su nombre de recurso de Amazon (ARN) son diferentes. ARNsno se puede usar como argumento para `iam_lambda_role`. Si ambos `ec2_iam_role` y `iam_lambda_role` están definidos, y el `scheduler` es `sge`, `slurm` o `torque`, entonces no se creará ningún rol. Si `scheduler` se establece en `awsbatch`, se crearán roles durante `pcluster`

[start](#). Para obtener ejemplos de políticas, consulte [ParallelClusterLambdaPolicyusando SGE, Slurm, o Torque](#) y [ParallelClusterLambdaPolicy con awsbatch](#).

No hay valor predeterminado.

```
iam_lambda_role = ParallelClusterLambdaRole
```

Note

El soporte para `iam_lambda_role` se agregó en la AWS ParallelCluster versión 2.10.1.

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

initial_queue_size

(Opcional) Establece el número inicial de EC2 instancias de Amazon que se lanzarán como nodos de cómputo en el clúster. Si se define la configuración de [queue_settings](#), esta configuración debe eliminarse y reemplazarse por la configuración de [initial_count](#) en las [secciones de \[compute_resource\]](#).

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

Esta configuración solo se aplica a los planificadores tradicionales (SGE, Slurm, y Torque). Si la [maintain_initial_size](#) configuración es true, entonces la [initial_queue_size](#) configuración debe ser al menos una (1).

Si el programador es `awsbatch`, utilice [min_vcpus](#) en su lugar.

El valor predeterminado es 2.

```
initial_queue_size = 2
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

key_name

(Opcional) Nombra un par de EC2 claves de Amazon existente con el que permitir el SSH acceso a las instancias.

```
key_name = mykey
```

Note

Antes de AWS ParallelCluster la versión 2.11.0, `key_name` era una configuración obligatoria.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

maintain_initial_size

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

(Opcional) Mantiene el tamaño inicial del grupo Auto Scaling para los programadores tradicionales (SGE, Slurm, y Torque).

Si el programador es `awsbatch`, utilice [desired_vcpus](#) en su lugar.

Esta configuración es un indicador booleano. Si se establece en `true`, el grupo de escalado automático nunca tendrá menos miembros que el valor de [initial_queue_size](#), y el valor de [initial_queue_size](#) debe ser uno (1) o mayor. El clúster también se puede ampliar y escalar hasta el valor de [max_queue_size](#). Si `cluster_type = spot`, el grupo de escalado automático puede tener instancias interrumpidas y el tamaño puede ser inferior a [initial_queue_size](#).

Si se establece en `false`, el grupo de escalado automático se puede reducir a cero (0) miembros, para evitar que los recursos permanezcan inactivos cuando no sean necesarios.

Si se define la configuración de [queue_settings](#), esta configuración debe eliminarse y reemplazarse por la configuración de [initial_count](#) y [min_count](#) en las [secciones de \[compute_resource\]](#).

El valor predeterminado es `false`.

```
maintain_initial_size = false
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

master_instance_type

(Opcional) Define el tipo de EC2 instancia de Amazon que se utiliza para el nodo principal. La arquitectura del tipo de instancia debe ser la misma que la arquitectura utilizada para la configuración de [compute_instance_type](#).

Si Regiones de AWS tiene una capa gratuita, el valor predeterminado es el tipo de instancia de capa gratuita (`t2.micro` o `t3.micro`). Si Regiones de AWS no tiene un nivel gratuito, el valor predeterminado es `t3.micro`. Para obtener más información sobre la capa AWS gratuita, consulte la [capa AWS gratuita. FAQs](#)

```
master_instance_type = t2.micro
```

Note

Antes de AWS ParallelCluster la versión 2.10.1, el valor predeterminado era «`t2.micro` in all». Regiones de AWS En AWS ParallelCluster versión 2.10.0, `p4d.24xlarge` no era compatible con el nodo principal. La compatibilidad con instancias AWS basadas en Graviton (como `A1` y `C6g`) se agregó en la AWS ParallelCluster versión 2.8.0.

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

master_root_volume_size

(Opcional) Especifica el tamaño del volumen raíz del nodo principal en gibibytes (GiB). El AMI soporte imprescindible. `growroot`

El valor predeterminado es `35`.

Note

Para AWS ParallelCluster las versiones entre 2.5.0 y 2.10.4, el valor predeterminado era 25. Antes de AWS ParallelCluster la versión 2.5.0, el valor predeterminado era 20.

```
master_root_volume_size = 35
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

max_queue_size

(Opcional) Establece el número máximo de EC2 instancias de Amazon que se pueden lanzar en el clúster. Si se define la configuración de [queue_settings](#), esta configuración debe eliminarse y reemplazarse por la configuración de [max_count](#) en las [secciones de \[compute_resource\]](#).

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

Esta configuración solo se aplica a los planificadores tradicionales (SGE, Slurm, y Torque).

Si el programador es `awsbatch`, utilice [max_vcpus](#) en su lugar.

El valor predeterminado es 10.

```
max_queue_size = 10
```

Política de actualización: esta configuración se puede cambiar durante una actualización, pero la flota de computación debe detenerse si se reduce el valor. De lo contrario, es posible que se cierren los nodos existentes.

max_vcpus

(Opcional) Especifica el número máximo de vCPUs en el entorno informático. Solo se utiliza si el programador es `awsbatch`.

El valor predeterminado es 20.

```
max_vcpus = 20
```

Política de actualización: esta configuración no se puede reducir durante una actualización.

min_vcpus

(Opcional) Mantiene el tamaño inicial del grupo de escalado automático para el programador de `awsbatch`.

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

Si el planificador es SGE, Slurm, o Torque, utilice [maintain_initial_size](#) en su lugar.

El entorno informático nunca tendrá menos miembros que el valor de [min_vcpus](#).

El valor predeterminado es 0.

```
min_vcpus = 0
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

placement

(Opcional) Define la lógica de los grupos con ubicación en clúster, lo que permite a todo el clúster o solo a las instancias de computación utilizar el grupo con ubicación en clúster.

Si se define la configuración de [queue_settings](#), esta configuración debe eliminarse y reemplazarse por la configuración de [placement_group](#) en cada una de las [secciones de \[queue\]](#). Si se usa el mismo grupo de ubicación para distintos tipos de instancias, es más probable que la solicitud no se realice correctamente debido a un error de capacidad insuficiente. Para obtener más información, consulta [Capacidad de instancia insuficiente](#) en la Guía del EC2 usuario de Amazon. Las colas múltiples solo pueden compartir un grupo de ubicación si se crean con antelación y se configuran en la configuración de [placement_group](#) de cada cola. Si cada [sección de \[queue\]](#) define una configuración de [placement_group](#), el nodo principal no puede estar en el grupo de ubicación de una cola.

Las opciones válidas son `cluster` o `compute`.

Este parámetro no se usa cuando el programador es `awsbatch`.

El valor predeterminado es `compute`.

```
placement = compute
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

placement_group

(Opcional) Define el grupo de ubicación en clúster. Si se define la configuración de [queue_settings](#), esta configuración debe eliminarse y reemplazarse por la configuración de [placement_group](#) en las [secciones de \[queue\]](#).

Las opciones válidas son las siguientes:

- DYNAMIC
- Un nombre de grupo de ubicación de EC2 clústeres de Amazon existente

Cuando se establece en DYNAMIC, se crea y elimina un grupo de ubicación único como parte de la pila del clúster.

Este parámetro no se usa cuando el programador es `awsbatch`.

Para obtener más información sobre los grupos de ubicación, consulte [Grupos de ubicación](#) en la Guía del EC2 usuario de Amazon. Si se usa el mismo grupo de ubicación para distintos tipos de instancias, es más probable que la solicitud no se realice correctamente debido a un error de capacidad insuficiente. Para obtener más información, consulta [Capacidad de instancia insuficiente](#) en la Guía del EC2 usuario de Amazon.

No hay valor predeterminado.

No todos los tipos de instancias admiten grupos de ubicación en clúster. Por ejemplo, el tipo de instancia predeterminado `t3.micro` no admite grupos con ubicación en clúster. Para obtener información sobre la lista de tipos de instancias que admiten grupos de ubicación en clústeres, consulta [Reglas y limitaciones de los grupos de ubicación en clústeres](#) en la Guía del EC2 usuario de Amazon. Consulte [Problemas con los grupos de ubicación y el lanzamiento de instancias](#) para obtener sugerencias al trabajar con grupos de ubicación.

```
placement_group = DYNAMIC
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

post_install

(Opcional) Especifica el URL script posterior a la instalación que se ejecuta una vez que se hayan completado todas las acciones de arranque del nodo. Para obtener más información, consulte [Acciones de arranque personalizadas](#).

Cuando se utiliza `awsbatch` como programador, el script posterior a la instalación solo se ejecuta en el nodo principal.

El formato de los parámetros puede ser `http://hostname/path/to/script.sh` o `s3://bucketname/path/to/script.sh`.

No hay valor predeterminado.

```
post_install = s3://<bucket-name>/my-post-install-script.sh
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

post_install_args

(Opcional) Especifica una lista de argumentos entre comillas que se pasará al script posterior a la instalación.

No hay valor predeterminado.

```
post_install_args = "argument-1 argument-2"
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

pre_install

(Opcional) Especifica el script URL de preinstalación que se ejecuta antes de que se inicie cualquier acción de arranque de despliegue de nodos. Para obtener más información, consulte [Acciones de arranque personalizadas](#).

Cuando se utiliza `awsbatch` como programador, el script de preinstalación solo se ejecuta en el nodo principal.

El formato de los parámetros puede ser `http://hostname/path/to/script.sh` o `s3://bucketname/path/to/script.sh`.

No hay valor predeterminado.

```
pre_install = s3://<bucket-name>/my-pre-install-script.sh
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

pre_install_args

Especifica una lista de argumentos entre comillas que se pasará al script de preinstalación.

No hay valor predeterminado.

```
pre_install_args = "argument-3 argument-4"
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

proxy_server

(Opcional) Define normalmente un servidor HTTPS proxy HTTP o un servidor proxy.

`http://x.x.x.x:8080`

No hay valor predeterminado.

```
proxy_server = http://10.11.12.13:8080
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

queue_settings

(Opcional) Especifica que el clúster usa colas en lugar de una flota de computación homogénea y qué [secciones de \[queue\]](#) se utilizan. La primera [sección de \[queue\]](#) en la lista es la cola del programador predeterminada. Los nombres de la sección de queue deben comenzar por una letra, contener no más de 30 caracteres y solo letras, números y guiones (-).

⚠ Important

Solo se admite cuando el spot se establece en `spot`. Los ajustes [cluster_type](#), [compute_instance_type](#), [initial_queue_size](#), [maintain_initial_size](#), [max_queue_size](#), [placement](#), [placement_group](#) y [spot_price](#) no deben especificarse. La configuración de [disable_hyperthreading](#) y [enable_efa](#) se puede especificar en la [sección de \[cluster\]](#) o en las [secciones de \[queue\]](#), pero no en ambas.

Se admiten hasta cinco (5) [secciones de \[queue\]](#).

Para obtener más información, consulte la [sección \[queue\]](#).

Por ejemplo, la siguiente configuración especifica que se utilicen las secciones que inician `[queue q1]` y `[queue q2]`.

```
queue_settings = q1, q2
```

ℹ Note

El soporte para [queue_settings](#) se agregó en la AWS ParallelCluster versión 2.9.0.

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

raid_settings

(Opcional) Identifica la `[raid]` sección con la RAID configuración de EBS volúmenes de Amazon. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[raid\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que inicia `[raid rs]` se utiliza para la configuración del escalado automático.

```
raid_settings = rs
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

s3_read_resource

(Opcional) Especifica un recurso de Amazon S3 al que se concede acceso de solo lectura a AWS ParallelCluster los nodos.

Por ejemplo, `arn:aws:s3:::my_corporate_bucket*` proporciona acceso de solo lectura a `my_corporate_bucket` cubo y a los objetos del cubo.

Consulte la sección sobre cómo [trabajar con](#) para obtener información detallada sobre el formato.

No hay valor predeterminado.

```
s3_read_resource = arn:aws:s3:::my_corporate_bucket*
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

s3_read_write_resource

Especifica un recurso de al que los nodos tienen concedido acceso de lectura y escritura.

Por ejemplo, `arn:aws:s3:::my_corporate_bucket/Development/*` proporciona acceso de lectura y escritura a todos los objetos de la `Development` carpeta del `my_corporate_bucket` cubo.

Consulte la sección sobre cómo [trabajar con](#) para obtener información detallada sobre el formato.

No hay valor predeterminado.

```
s3_read_write_resource = arn:aws:s3:::my_corporate_bucket/*
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

scaling_settings

Identifica la sección de `[scaling]` con la configuración del escalado automático. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[scaling\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que inicia `[scaling custom]` se utiliza para la configuración del escalado automático.

```
scaling_settings = custom
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

scheduler

(Obligatorio) Define el programador de clúster.

Las opciones válidas son las siguientes:

`awsbatch`

AWS Batch

Para obtener más información sobre el programador de `awsbatch`, consulte la [configuración de red](#) y [AWS Batch \(awsbatch\)](#).

`sge`

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

Son of Grid Engine (SGE)

`slurm`

Slurm Workload Manager (Slurm)

`torque`

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

Torque Resource Manager (Torque)

Note

Antes de AWS ParallelCluster la versión 2.7.0, el `scheduler` parámetro era opcional y el predeterminado era `sge`. A partir de AWS ParallelCluster la versión 2.7.0, el `scheduler` parámetro es obligatorio.

```
scheduler = slurm
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

shared_dir

(Opcional) Define la ruta en la que se monta el EBS volumen compartido de Amazon.

No utilices esta opción con varios EBS volúmenes de Amazon. En su lugar, proporcione los valores [shared_dir](#) en cada sección de .

Consulta la [\[ebs\]sección](#) para obtener más información sobre cómo trabajar con varios EBS volúmenes de Amazon.

El valor predeterminado es `/shared`.

El siguiente ejemplo muestra un EBS volumen compartido de Amazon montado en `/myshared`.

```
shared_dir = myshared
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

spot_bid_percentage

(Opcional) Establece el porcentaje de demanda que se utiliza para calcular el ComputeFleet precio spot máximo para `awsbatch` el programador.

Si no se especifica, se selecciona el precio de mercado de spot actual con el precio bajo demanda como límite.

```
spot_bid_percentage = 85
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

spot_price

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque programadores.

(Opcional) Establece el precio máximo al contado para los programadores ComputeFleet tradicionales (SGE, Slurm, y Torque). Se usa solo cuando el [cluster_type](#) ajuste está establecido en spot. Si no especifica ningún valor, se le cobrará el precio de spot, con el precio bajo demanda como límite máximo. Si se define la configuración de [queue_settings](#), esta configuración debe eliminarse y reemplazarse por la configuración de [spot_price](#) en las [secciones de \[compute_resource\]](#).

Si el programador es awsbatch, utilice [spot_bid_percentage](#) en su lugar.

Para obtener ayuda para encontrar una instancia de spot que se ajuste a sus necesidades, consulte el [Asistente de instancias de spot](#).

```
spot_price = 1.50
```

Note

En AWS ParallelCluster la versión 2.5.0, si `cluster_type = spot` [spot_price](#) no se especifica, la instancia se lanza cuando se produce un ComputeFleet error. Esto se solucionó en la AWS ParallelCluster versión 2.5.1.

Política de actualización: esta configuración se puede cambiar durante una actualización.

tags

(Opcional) Define las etiquetas que utilizará. AWS CloudFormation

Si se especifican etiquetas de línea de comandos a través de `--tags`, se combinan con etiquetas de configuración.

Las etiquetas de la línea de comandos sobrescriben las etiquetas de configuración que tienen la misma clave.

Las etiquetas están JSON formateadas. No utilice comillas fuera de las llaves.

Para obtener más información, consulte [AWS CloudFormation resource tags type](#) en la Guía del usuario de AWS CloudFormation .

```
tags = {"key" : "value", "key2" : "value2"}
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

Note

La política de actualización no permitía cambiar la configuración tags hasta que se detuvo la flota informática desde AWS ParallelCluster versión 2.0.0 hasta la versión 2.9.1.

En el caso de las versiones 2.10.0 a 2.11.7, la política de actualización indicada que permitía cambiar la configuración de tags no es precisa. No se admite una actualización de clúster al modificar esta configuración.

template_url

(Opcional) Define la ruta a la AWS CloudFormation plantilla que se utiliza para crear el clúster.

Actualiza la plantilla que se utilizó originalmente para crear la pila.

El valor predeterminado es `https://aws_region_name-aws-parallelcluster.s3.amazonaws.com/templates/aws-parallelcluster-version.cfn.json`.

Warning

Este es un parámetro avanzado. Cualquier cambio en esta configuración es por su cuenta y riesgo.

```
template_url = https://us-east-1-aws-parallelcluster.s3.amazonaws.com/templates/aws-parallelcluster-2.11.9.cfn.json
```

Política de actualización: esta configuración no se analiza durante una actualización.

vpc_settings

(Obligatorio) Identifica la [vpc] sección con la VPC configuración de Amazon en la que se despliega el clúster. El nombre de la sección debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

Para obtener más información, consulte la [sección \[vpc\]](#).

Por ejemplo, la siguiente configuración especifica que la sección que comienza [vpc public] se usa para la VPC configuración de Amazon.

```
vpc_settings = public
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

Sección [compute_resource]

Define los ajustes de configuración de un recurso informático. [Las secciones \[compute_resource\]](#) son referenciadas por el ajuste [compute_resource_settings](#) en la [sección \[queue\]](#). [Las secciones \[compute_resource\]](#) solo son compatibles cuando [scheduler](#) está configurado a slurm.

El formato es [compute_resource <compute-resource-name>]. *compute-resource-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[compute_resource cr1]
instance_type = c5.xlarge
min_count = 0
initial_count = 2
max_count = 10
spot_price = 0.5
```

Note

La compatibilidad con la [\[compute_resource\] sección](#) se añadió en AWS ParallelCluster versión 2.9.0.

Temas

- [initial_count](#)
- [instance_type](#)
- [max_count](#)
- [min_count](#)
- [spot_price](#)

initial_count

(Opcional) Establece el número inicial de instancias de Amazon EC2 que se van a lanzar para este recurso informático. La creación del clúster no se completa hasta que se hayan lanzado al menos este número de nodos al recurso informático. Si la configuración [compute_type](#) de la cola es spot y no hay suficientes instancias puntuales disponibles, es posible que se agote el tiempo de espera para la creación del clúster y se produzca un error. Cualquier recuento superior a la configuración [min_count](#) es capacidad dinámica sujeta a la configuración [scaledown_idletime](#). Esta configuración reemplaza la configuración [initial_queue_size](#).

El valor predeterminado es 0.

```
initial_count = 2
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

instance_type

(Obligatorio) Define el tipo de instancia de Amazon EC2 que se utiliza para este recurso informático. La arquitectura del tipo de instancia debe ser la misma que la arquitectura utilizada para la configuración [master_instance_type](#). La configuración `instance_type` debe ser única para cada [sección \[compute_resource\]](#) a la que haga referencia una [sección \[queue\]](#). Esta configuración reemplaza la configuración [compute_instance_type](#).

```
instance_type = t2.micro
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

max_count

(Opcional) Establece el número máximo de instancias de Amazon EC2 que se van a lanzar para este recurso informático. Cualquier recuento superior a la configuración [initial_count](#) se inicia en modo de apagado. Esta configuración reemplaza la configuración [max_queue_size](#).

El valor predeterminado es 10.

```
max_count = 10
```

Política de actualización: para reducir el tamaño de una cola por debajo del número actual de nodos, es necesario detener primero la flota de computación.

Note

La política de actualización no permitía cambiar la configuración `max_count` hasta que se detuvo la flota informática desde AWS ParallelCluster versión 2.0.0 hasta la versión 2.9.1.

min_count

(Opcional) Establece el número mínimo de instancias de Amazon EC2 que se van a lanzar para este recurso informático. Todos estos nodos tienen capacidad estática. La creación del clúster no se completa hasta que se hayan lanzado al menos este número de nodos al recurso informático.

El valor predeterminado es 0.

```
min_count = 1
```

Política de actualización: para reducir el número de nodos estáticos de una cola, es necesario detener primero la flota de computación.

Note

La política de actualización no permitía cambiar la configuración `min_count` hasta que se detuvo la flota informática desde AWS ParallelCluster versión 2.0.0 hasta la versión 2.9.1.

spot_price

(Opcional) Establece el precio spot máximo para este recurso informático. Se usa solo cuando la [compute_type](#) configuración de la cola que contiene estos recursos de cómputo está establecida en `spot`. Esta configuración reemplaza la configuración [spot_price](#).

Si no especifica ningún valor, se le cobrará el precio de spot, con el precio bajo demanda como límite máximo.

Para obtener ayuda para encontrar una instancia de spot que se ajuste a sus necesidades, consulte el [Asistente de instancias de spot](#).

```
spot_price = 1.50
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

Sección [cw_log]

Define los ajustes de configuración del panel de CloudWatch.

El formato es [cw_log *cw-log-name*]. *cw-logname* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[cw_log custom-cw-log]  
enable = true  
retention_days = 14
```

Para obtener más información, consulte [Integración con Amazon CloudWatch Logs](#), [CloudWatch Panel de control de Amazon](#) y [Integración con Amazon CloudWatch Logs](#).

Note

Se ha agregado compatibilidad para `cw_log` en AWS ParallelCluster versión 2.6.0.

enable

(Opcional) Indica si CloudWatch Logs está habilitado en el registro.

El valor predeterminado es `true`. Use `false` para deshabilitar CloudWatch Logs.

En el ejemplo siguiente se activa CloudWatch Logs.

```
enable = true
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

retention_days

(Opcional) Indica cuántos días CloudWatch Logs conserva los eventos de registro individuales.

El valor predeterminado es 14. Los valores admitidos son 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827 y 3653.

En el siguiente ejemplo se configura CloudWatch Logs para que conserve los eventos de registro durante 30 días.

```
retention_days = 30
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

Sección [dashboard]

Define los ajustes de configuración del panel de CloudWatch.

El formato es [dashboard *dashboard-name*]. *dashboard-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[dashboard custom-dashboard]  
enable = true
```

Note

Se ha agregado compatibilidad para dashboard en AWS ParallelCluster versión 2.4.0.

enable

(Opcional) Indica si el panel de CloudWatch está habilitado.

El valor predeterminado es `true`. Se usa `false` para deshabilitar el panel de CloudWatch.

El siguiente ejemplo habilita el panel de CloudWatch.

```
enable = true
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

Sección de `[dcv]`

Define los ajustes de configuración para el DCV servidor Amazon que se ejecuta en el nodo principal.

Para crear y configurar un DCV servidor Amazon, especifique el clúster [dcv_settings](#) con el nombre que [enable](#) defina en la `dcv` sección y establézcalo en `master` `alinux2centos7`, y [base_os](#) en `ubuntu1804` o `ubuntu2004`. Si el nodo principal es una ARM instancia, [base_os](#) establézcalo en `alinux2centos7`, o `ubuntu1804`.

El formato es `[dcv dcv-name]`. *dcv-name* debe comenzar con una letra, no contener más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[dcv custom-dcv]  
enable = master  
port = 8443  
access_from = 0.0.0.0/0
```

Para obtener más información, consulte [Conéctese al nodo principal a través de Amazon DCV](#)

Important

De forma predeterminada, el DCV puerto de Amazon configurado por AWS ParallelCluster está abierto a todas IPv4 las direcciones. Sin embargo, solo puedes conectarte a un DCV puerto de Amazon si tienes el puerto URL para la DCV sesión de Amazon y te conectas a la DCV sesión de Amazon en un plazo de 30 segundos desde el momento en que URL se devuelve `cluster dcv connect`. Usa la [access_from](#) configuración para restringir aún más el acceso al DCV puerto de Amazon con un rango CIDR de IP con formato y usa la [port](#) configuración para configurar un puerto no estándar.

Note

La compatibilidad con la [sección de \[dcv\]](#) en centos8 se eliminó en la versión 2.10.4 de AWS ParallelCluster. Support para la [\[dcv\]sección](#) sobre centos8 se agregó en la AWS ParallelCluster versión 2.10.0. El soporte para la [\[dcv\]sección](#) sobre instancias AWS basadas en Graviton se agregó en la AWS ParallelCluster versión 2.9.0. Support para la [\[dcv\]sección](#) sobre alinux2 y ubuntu1804 se agregó en la AWS ParallelCluster versión 2.6.0. Support para la [\[dcv\]sección](#) sobre centos7 se agregó en la AWS ParallelCluster versión 2.5.0.

access_from

(Opcional, recomendado) Especifica el rango CIDR de IP con formato para las conexiones a Amazon DCV. Esta configuración solo se usa cuando se crea el grupo de seguridad de AWS ParallelCluster.

El valor predeterminado es `0.0.0.0/0`, que permite el acceso desde cualquier dirección de Internet.

```
access_from = 0.0.0.0/0
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

enable

(Obligatorio) Indica si Amazon DCV está activado en el nodo principal. Para habilitar Amazon DCV en el nodo principal y configurar la regla de grupo de seguridad requerida, defina la `enable` configuración en `master`.

El siguiente ejemplo habilita Amazon DCV en el nodo principal.

```
enable = master
```

Note

Amazon genera DCV automáticamente un certificado autofirmado que se utiliza para proteger el tráfico entre el DCV cliente de Amazon y el DCV servidor de Amazon.

que se ejecuta en el nodo principal. Para configurar su propio certificado, consulte [DCVHTTPSCertificado de Amazon](#).

Política de actualización: si se cambia esta configuración, no se permite la actualización.

port

(Opcional) Especifica el puerto de AmazonDCV.

El valor predeterminado es 8443.

```
port = 8443
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

Sección de [ebs]

Define la configuración de volumen de Amazon EBS para los volúmenes que están montados en el nodo principal y se comparten a los nodos de computación a través de NFS.

Para obtener información sobre cómo incluir volúmenes Amazon EBS en la definición de su clúster, consulte [Sección de \[cluster\] / ebs_settings](#).

Si desea utilizar un volumen Amazon EBS existente para un almacenamiento permanente a largo plazo que sea independiente del ciclo de vida del clúster, especifique [ebs_volume_id](#).

Si no lo especifica [ebs_volume_id](#), AWS ParallelCluster crea el volumen de EBS a partir de la [ebs] configuración al crear el clúster y elimina el volumen y los datos cuando se elimina el clúster.

Para obtener más información, consulte [Mejores prácticas: trasladar un clúster a un nuevo AWS ParallelCluster versión secundaria o de parche](#).

El formato es [ebs *ebs-name*]. *ebs-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[ebs custom1]
shared_dir = vol1
ebs_snapshot_id = snap-xxxxx
```

```
volume_type = io1
volume_iops = 200
...

[ebs custom2]
shared_dir = vol2
...

...
```

Temas

- [shared_dir](#)
- [ebs_kms_key_id](#)
- [ebs_snapshot_id](#)
- [ebs_volume_id](#)
- [encrypted](#)
- [volume_iops](#)
- [volume_size](#)
- [volume_throughput](#)
- [volume_type](#)

shared_dir

(Obligatorio) Especifica la ruta donde se monta el volumen de Amazon EBS compartido.

Este parámetro es obligatorio cuando se usan varios volúmenes de Amazon EBS.

Cuando se utiliza un volumen de Amazon EBS, esta opción sobrescribe el [shared_dir](#) que se especifica en la [sección \[cluster\]](#). En el siguiente ejemplo, el volumen se monta en /vol1.

```
shared_dir = vol1
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

ebs_kms_key_id

(Opcional) Especifica una AWS KMS clave personalizada para usarla en el cifrado.

Este parámetro se debe usar junto con la opción `encrypted = true`. También debe tener un [ec2_iam_role](#) personalizado.

Para obtener más información, consulte [Cifrado de disco con una clave KMS personalizada](#).

```
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

ebs_snapshot_id

(Opcional) Define el ID de instantánea de Amazon EBS si utiliza una instantánea como origen del volumen.

No hay valor predeterminado.

```
ebs_snapshot_id = snap-xxxxx
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

ebs_volume_id

(Opcional) Define el ID de un volumen de Amazon EBS existente para adjuntarlo al nodo principal.

No hay valor predeterminado.

```
ebs_volume_id = vol-xxxxxx
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

encrypted

(Opcional) Especifica si el volumen de Amazon EBS está cifrado. Nota: No debe usarse con instantáneas.

El valor predeterminado es `false`.

```
encrypted = false
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

volume_iops

(Opcional) Define el número de IOPS para volúmenes de tipo io1, io2 y gp3.

El valor predeterminado, los valores admitidos y la proporción de datos `volume_iops` a `volume_size` varía entre [volume_type](#) y [volume_size](#).

`volume_type = io1`

`volume_iops` predeterminado = 100

Valores admitidos `volume_iops` = 100–64 000 †

Relación máxima `volume_iops` a `volume_size` = 50 IOPS por cada GiB. 5000 IOPS requieren un `volume_size` de al menos 100 GiB.

`volume_type = io2`

`volume_iops` predeterminado = 100

Valores admitidos `volume_iops` = 100–64 000 (256 000 para los volúmenes de io2 Block Express) †

Relación máxima `volume_iops` a `volume_size` = 500 IOPS por cada GiB. 5000 IOPS requieren un `volume_size` de al menos 10 GiB.

`volume_type = gp3`

`volume_iops` predeterminado = 3000

Valores admitidos `volume_iops` = 3000–16 000

Relación máxima `volume_iops` a `volume_size` = 500 IOPS por cada GiB. 5000 IOPS requieren un `volume_size` de al menos 10 GiB.

```
volume_iops = 200
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

† Las IOPS máximas solo se garantizan en [las instancias creadas en el sistema Nitro](#) aprovisionadas con más de 32 000 IOPS. Otras instancias garantizan hasta 32,000 IOPS. Es posible que los volúmenes [más antiguos no alcancen el rendimiento máximo a menos que](#) modifique el

io1, io2 Los volúmenes de Block Express admiten valores de `volume_iops` de hasta 256 000. Para obtener más información, consulte [io2Bloque de volúmenes Express \(versión preliminar\)](#) en la Guía del usuario de Amazon EC2.

volume_size

(Opcional) Especifica el tamaño del volumen que se va a crear, en GiB (si no se utiliza una instantánea).

El valor predeterminado y los valores admitidos varía en [volume_type](#).

`volume_type = standard`

Predeterminado `volume_size = 20 GiB`

Valores admitidos `volume_size = 1–1024 GiB`

`volume_type = gp2, io1, io2 y gp3`

Predeterminado `volume_size = 20 GiB`

Valores admitidos `volume_size = 1–16 384 GiB`

`volume_type = sc1 y st1`

Predeterminado `volume_size = 500 GiB`

Valores admitidos `volume_size = 500–16 384 GiB`

```
volume_size = 20
```

Note

Antes de AWS ParallelCluster la versión 2.10.1, el valor predeterminado para todos los tipos de volumen era de 20 GiB.

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

volume_throughput

(Opcional) Define el rendimiento de los tipos de volumen `gp3`, en MiB/s.

El valor predeterminado es 125.

Valores admitidos `volume_throughput` = 125–1000 MiB/s

La relación entre `volume_throughput` y `volume_iops` no puede ser superior a 0,25. El rendimiento máximo de 1000 MiB/s requiere que la configuración de `volume_iops` sea de al menos 4000.

```
volume_throughput = 1000
```

Note

El soporte para `volume_throughput` se agregó en la AWS ParallelCluster versión 2.10.1.

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

volume_type

(Opcional) Especifica el [tipo de volumen de Amazon EBS](#) del volumen que desea lanzar.

Las opciones de volumen válidas son las siguientes:

`gp2`, `gp3`

General Purpose SSD

`io1`, `io2`

Provisioned IOPS SSD

`st1`

HDD con rendimiento optimizado

`sc1`

HDD en frío

`standard`

Magnéticos de generaciones anteriores

Para obtener más información, consulte [Tipos de volúmenes de Amazon EBS](#) en la Guía del usuario de Amazon EC2.

El valor predeterminado es gp2.

```
volume_type = io2
```

Note

Support gp3 y io2 se agregó en la AWS ParallelCluster versión 2.10.1.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

Sección [efs]

Define los ajustes de configuración del Amazon EFS que se ha montado en los nodos principal y de computación. Para obtener más información, consulte [CreateFileSystem](#) en la Referencia de la API de Amazon FSx.

Para obtener información sobre cómo incluir los sistemas de archivos Amazon EFS en la definición de su clúster, consulte [Sección de \[cluster\] / efs_settings](#).

Si desea utilizar un sistema de archivos Amazon EFS existente para un almacenamiento permanente a largo plazo que sea independiente del ciclo de vida del clúster, especifique [efs_fs_id](#).

Si no lo especifica [efs_fs_id](#), AWS ParallelCluster crea el sistema de archivos Amazon EFS a [efs] partir de la configuración al crear el clúster y elimina el sistema de archivos y los datos cuando se elimina el clúster.

Para obtener más información, consulte [Mejores prácticas: trasladar un clúster a un nuevo AWS ParallelCluster versión secundaria o de parche](#).

El formato es [efs *efs-name*]. *efs-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[efs customfs]  
shared_dir = efs
```

```
encrypted = false
performance_mode = generalPurpose
```

Temas

- [efs_fs_id](#)
- [efs_kms_key_id](#)
- [encrypted](#)
- [performance_mode](#)
- [provisioned_throughput](#)
- [shared_dir](#)
- [throughput_mode](#)

efs_fs_id

(Opcional) Define el ID del sistema de archivos de Amazon EFS para un sistema de archivos ya existente.

Al especificar esta opción, anula las demás opciones de Amazon EFS salvo [shared_dir](#).

Si establece esta opción, solo admite los sistemas de archivos siguientes:

- Sistemas de archivos que no cuentan con un destino de montaje en la zona de disponibilidad de la pila.
- Que cuentan con un destino de montaje ya existente en la zona de disponibilidad de la pila, con tráfico NFS de entrada y salida permitido desde `0.0.0.0/0`.

La comprobación de estado para validar [efs_fs_id](#) requiere que el rol de IAM tenga los siguientes permisos:

- `elasticfilesystem:DescribeMountTargets`
- `elasticfilesystem:DescribeMountTargetSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkInterfaceAttribute`

Para evitar errores, debe añadir los permisos siguientes a su rol de IAM o establecer `sanity_check = false`.

Important

Cuando establece un destino de montaje con tráfico NFS de entrada y salida permitido desde `0.0.0.0/0`, expone el sistema de archivos a solicitudes de montaje de NFS desde cualquier lugar en la zona de disponibilidad del destino de montaje. AWS no recomienda crear un destino de montaje en la zona de disponibilidad de la pila. En su lugar, sigamos con este paso AWS. Si debe tener un destino de montaje en la zona de disponibilidad de la pila, considere la posibilidad de usar un grupo de seguridad personalizado proporcionando una opción `vpc_security_group_id` en la [sección \[vpc\]](#). A continuación, añada dicho grupo de seguridad al destino de montaje y desactive `sanity_check` para crear el clúster.

No hay valor predeterminado.

```
efs_fs_id = fs-12345
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

efs_kms_key_id

(Opcional) Identifica la clave administrada por el cliente de AWS Key Management Service (AWS KMS) que se utilizará para proteger el sistema de archivos cifrados. Si se ha configurado esta opción, el ajuste `encrypted` debe establecerse en `true`. Esto corresponde al parámetro `KmsKeyId` de la Referencia de la API de Amazon EFS.

No hay valor predeterminado.

```
efs_kms_key_id = 1234abcd-12ab-34cd-56ef-1234567890ab
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

encrypted

(Opcional) Indica si el sistema de archivos está cifrado. Esto corresponde al parámetro `Encrypted` de la Referencia de la API de Amazon EFS.

El valor predeterminado es `false`.

```
encrypted = true
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

performance_mode

(Opcional) Define el modo de rendimiento del sistema de archivos. Esto corresponde al parámetro [PerformanceMode](#) de la Referencia de la API de Amazon EFS.

Las opciones válidas son las siguientes:

- `generalPurpose`
- `maxIO`

Ambos valores distinguen entre mayúsculas y minúsculas.

Recomendamos el modo de rendimiento `generalPurpose` para la mayoría de sistemas de archivos.

Los sistemas de archivos que usan el modo de rendimiento `maxIO` pueden escalar a mayores niveles de rendimiento de agregación y operaciones por segundo. Sin embargo, hay una compensación de latencias ligeramente más altas para la mayoría de las operaciones de archivos.

Este parámetro no se puede cambiar después de que se haya creado el sistema de archivos.

El valor predeterminado es `generalPurpose`.

```
performance_mode = generalPurpose
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

provisioned_throughput

(Opcional) Define el rendimiento aprovisionado del sistema de archivos, medido en MiB/s. Esto corresponde al parámetro [ProvisionedThroughputInMibps](#) de la Referencia de la API de Amazon EFS.

Si usa este parámetro, debe establecer [throughput_mode](#) en `provisioned`.

La cuota de rendimiento es de 1024 MiB/s. Para solicitar un aumento de cuota, póngase en contacto con AWS Support.

El valor mínimo es 0.0 MiB/s.

```
provisioned_throughput = 1024
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

shared_dir

(Obligatorio) Define el punto de montaje de Amazon EFS en los nodos principal y de computación.

Este parámetro es obligatorio. La sección de Amazon EFS solo se usa si se especifica [shared_dir](#).

No utilice `NONE` o `/NONE` como directorio compartido.

En el siguiente ejemplo se monta Amazon EFS en `/efs`.

```
shared_dir = efs
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

throughput_mode

(Opcional) Define el modo de rendimiento del sistema de archivos. Esto corresponde al parámetro [ThroughputMode](#) de la Referencia de la API de Amazon EFS.

Las opciones válidas son las siguientes:

- `bursting`
- `provisioned`

El valor predeterminado es `bursting`.

```
throughput_mode = provisioned
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

Sección [fsx]

Define las opciones de configuración de un sistema de archivos FSx para Lustre asociado. Para obtener más información, consulte [Amazon FSx CreateFileSystem](#) en la Referencia de la API de Amazon FSx.

Si el `base_os` es `alinux2`, `centos7`, `ubuntu1804` o `ubuntu2004`, es compatible con FSx para Lustre.

Si se utiliza Amazon Linux, el núcleo debe ser de la versión `4.14.104-78.84.amzn1.x86_64` o posterior. Para obtener instrucciones, consulte [Instalación del cliente Lustre](#) en la Guía del usuario de Amazon FSx para Lustre.

Note

FSx para Lustre no se admite actualmente cuando se usa `awsbatch` como programador.

Note

Se ha eliminado la compatibilidad de FSx for Lustre en `centos8` en AWS ParallelCluster versión 2.2.1. Se ha agregado la compatibilidad de `ubuntu2004` en AWS ParallelCluster versión 2.11.1. Se ha agregado la compatibilidad de `centos8` en AWS ParallelCluster versión 2.10.1. Se ha agregado la compatibilidad de FSx for Lustre en `alinux2`, `ubuntu1604` y `ubuntu1804` en AWS ParallelCluster versión 2.6.0. Se ha agregado la compatibilidad de `centos7` en AWS ParallelCluster versión 2.4.1.

Si se usa un sistema de archivos ya existente, debe asociarse a un grupo de seguridad que permita el tráfico TCP de entrada a través del puerto 988. Establecer el origen en `0.0.0.0/0` en una regla de grupo de seguridad proporciona acceso cliente desde todos los intervalos IP del grupo de seguridad de VPC para el protocolo y el intervalo de puertos de esa regla. Para limitar aún más el acceso a los sistemas de archivos, recomendamos utilizar orígenes más restrictivos para las reglas de grupo de seguridad. Por ejemplo, puede usar rangos de CIDR, direcciones IP o ID de grupos de seguridad más específicos. Esto se realiza automáticamente si no se usa [vpc_security_group_id](#).

Si desea utilizar un sistema de archivos Amazon FSx existente para un almacenamiento permanente a largo plazo que sea independiente del ciclo de vida del clúster, especifique [fsx_fs_id](#).

Si no lo especifica [fsx_fs_id](#), AWS ParallelCluster crea el sistema de archivos FSx for Lustre a [fsx] partir de la configuración al crear el clúster y elimina el sistema de archivos y los datos cuando se elimina el clúster.

Para obtener más información, consulte [Mejores prácticas: trasladar un clúster a un nuevo AWS ParallelCluster versión secundaria o de parche](#).

El formato es [fsx *fsx-name*]. *fsx-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[fsx fs]
shared_dir = /fsx
fsx_fs_id = fs-073c3803dca3e28a6
```

Para crear y configurar un nuevo sistema de archivos, use los siguientes parámetros:

```
[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
import_path = s3://bucket
weekly_maintenance_start_time = 1:00:00
```

Temas

- [auto_import_policy](#)
- [automatic_backup_retention_days](#)
- [copy_tags_to_backups](#)
- [daily_automatic_backup_start_time](#)
- [data_compression_type](#)
- [deployment_type](#)
- [drive_cache_type](#)
- [export_path](#)
- [fsx_backup_id](#)

- [fsx_fs_id](#)
- [fsx_kms_key_id](#)
- [import_path](#)
- [imported_file_chunk_size](#)
- [per_unit_storage_throughput](#)
- [shared_dir](#)
- [storage_capacity](#)
- [storage_type](#)
- [weekly_maintenance_start_time](#)

auto_import_policy

(Opcional) Especifica la política de importación automática para reflejar los cambios en el bucket de S3 utilizado para crear el sistema de archivos FSx para Lustre. Los valores posibles son los siguientes:

NEW

FSx para Lustre importa automáticamente descripciones de directorio de cualquier objeto nuevo agregado al bucket de S3 vinculado que no exista actualmente en el sistema de archivos FSx para Lustre.

NEW_CHANGED

FSx para Lustre importa automáticamente descripciones de archivos y directorios de cualquier objeto nuevo agregado al bucket de S3 y de cualquier objeto existente que se cambie en el bucket de S3.

Esto corresponde a la propiedad [AutoImportPolicy](#). Para obtener más información, consulte [Importar automáticamente actualizaciones desde su bucket de S3](#) en la Guía del usuario de Amazon FSx para Lustre. Cuando se especifica el parámetro [auto_import_policy](#), no se deben especificar los parámetros [automatic_backup_retention_days](#), [copy_tags_to_backups](#), [daily_automatic_backup_start_time](#) y [fsx_backup_id](#).

Si no se especifica la configuración `auto_import_policy`, las importaciones automáticas están deshabilitadas. FSx para Lustre solo actualiza las descripciones de archivos y directorios desde el bucket de S3 vinculado cuando se crea el sistema de archivos.

```
auto_import_policy = NEW_CHANGED
```

Note

Se ha agregado compatibilidad para [auto_import_policy](#) en AWS ParallelCluster versión 2.10.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

automatic_backup_retention_days

(Opcional) Especifica el número de días que se conservan las copias de seguridad automáticas. Solo es válido para su uso con tipos de implementación PERSISTENT_1. Cuando se especifica el parámetro [automatic_backup_retention_days](#), no se deben especificar los parámetros [auto_import_policy](#), [export_path](#), [import_path](#) y [imported_file_chunk_size](#). Esto corresponde a la propiedad [AutomaticBackupRetentionDays](#).

El valor predeterminado es 0. Este ajuste deshabilita las copias de seguridad automáticas. Los valores posibles son números enteros entre 0 y 35, ambos inclusive.

```
automatic_backup_retention_days = 35
```

Note

Se ha agregado compatibilidad para [automatic_backup_retention_days](#) en AWS ParallelCluster versión 2.8.0.

Política de actualización: esta configuración se puede cambiar durante una actualización.

copy_tags_to_backups

(Opcional) Especifica si las etiquetas del sistema de archivos se copian en las copias de seguridad. Solo es válido para su uso con tipos de implementación PERSISTENT_1. Cuando se especifica el [copy_tags_to_backups](#) parámetro, se [automatic_backup_retention_days](#) debe especificar con un valor superior a 0 y no se deben especificar los parámetros [auto_import_policy](#),

[export_path](#), [import_path](#) y [imported_file_chunk_size](#). Esto corresponde a la propiedad [CopyTagsToBackups](#).

El valor predeterminado es `false`.

```
copy_tags_to_backups = true
```

Note

Se ha agregado compatibilidad para [copy_tags_to_backups](#) en AWS ParallelCluster versión 2.8.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

daily_automatic_backup_start_time

(Opcional) Especifica la hora del día (UTC) para iniciar las copias de seguridad automáticas. Solo es válido para su uso con tipos de implementación `PERSISTENT_1`. Cuando se especifica el [daily_automatic_backup_start_time](#) parámetro, se [automatic_backup_retention_days](#) debe especificar con un valor superior a 0 y no se deben especificar [imported_file_chunk_size](#) los parámetros [auto_import_policy](#) [export_path](#)[import_path](#), y. Esto corresponde a la propiedad [DailyAutomaticBackupStartTime](#).

El formato es `HH:MM`, donde `HH` es la hora de relleno cero del día (0-23), y `MM` es el minuto de la hora con relleno cero. Por ejemplo, a las 1:03 a.m. UTC es la siguiente.

```
daily_automatic_backup_start_time = 01:03
```

El valor predeterminado es un tiempo aleatorio entre `00:00` y `23:59`.

Note

Se ha agregado compatibilidad para [daily_automatic_backup_start_time](#) en AWS ParallelCluster versión 2.8.0.

Política de actualización: esta configuración se puede cambiar durante una actualización.

data_compression_type

(Opcional) Especifica el tipo de compresión de datos FSx for Lustre. Esto corresponde a la propiedad [DataCompressionType](#). Para obtener más información, consulte [Compresión de datos de FSx para Lustre](#) en la Guía del usuario de Amazon FSx para Lustre.

El único valor válido es LZ4. Para deshabilitar la compresión de datos, elimine el parámetro. [data_compression_type](#)

```
data_compression_type = LZ4
```

Note

Se ha agregado compatibilidad para [data_compression_type](#) en AWS ParallelCluster versión 211.0.

Política de actualización: esta configuración se puede cambiar durante una actualización.

deployment_type

(Opcional) Especifica el tipo de despliegue de FSx for Lustre. Esto corresponde a la propiedad [DeploymentType](#). Para más información, consulte [Opciones de implementación de FSx para Lustre](#) en la Guía del usuario de Amazon FSx para Lustre. Elija un tipo de implementación temporal para el almacenamiento temporal y el procesamiento de datos a corto plazo. SCRATCH_2 es la última generación de sistemas de archivos temporales. Ofrece mayor rendimiento de ráfagas en comparación con el rendimiento de referencia y cifrado en tránsito de datos.

Los valores válidos son SCRATCH_1, SCRATCH_2 y PERSISTENT_1.

SCRATCH_1

El tipo de implementación predeterminado de FSx para Lustre. Con este tipo de implementación, los valores posibles de [storage_capacity](#) son 1200, 2400 y cualquier múltiplo de 3600. Se ha agregado compatibilidad para SCRATCH_1 en AWS ParallelCluster versión 2.4.0.

SCRATCH_2

La última generación de sistemas de archivos temporales. Soporta hasta seis veces el rendimiento básico para cargas de trabajo con picos de actividad. También admite el cifrado

de datos en tránsito para los tipos de instancias compatibles, si es compatible. Regiones de AWS Para obtener más información, consulte [Cifrado de datos en tránsito](#) en la Guía del usuario de Amazon FSx para Lustre. Con este tipo de implementación, los valores posibles de [storage_capacity](#) son 1200 y cualquier múltiplo de 2400. Se ha agregado compatibilidad para SCRATCH_2 en AWS ParallelCluster versión 2.6.0.

PERSISTENT_1

Diseñado para el almacenamiento a largo plazo. Los servidores de archivos son de alta disponibilidad, y los datos se replican dentro de la misma Zona de Disponibilidad en la que se encuentra el sistema de archivos. Admite el cifrado en tránsito de datos para los tipos de instancia compatibles. Con este tipo de implementación, los valores posibles de [storage_capacity](#) son 1200 y cualquier múltiplo de 2400. Se ha agregado compatibilidad para PERSISTENT_1 en AWS ParallelCluster versión 2.6.0.

El valor predeterminado es SCRATCH_1.

```
deployment_type = SCRATCH_2
```

Note

Se ha agregado compatibilidad para [deployment_type](#) en AWS ParallelCluster versión 2.6.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

drive_cache_type

(Opcional) Especifica que el sistema de archivos tiene una memoria caché de unidad SSD. Esto solo debe configurarse si el ajuste [storage_type](#) se establece a HDD. Esto corresponde a la propiedad [DriveCachetype](#). Para más información, consulte [Opciones de implementación de FSx para Lustre](#) en la Guía del usuario de Amazon FSx para Lustre.

El único valor válido es READ. Para deshabilitar la memoria caché de la unidad SSD, no especifique la configuración. `drive_cache_type`

```
drive_cache_type = READ
```

Note

Se ha agregado compatibilidad para [drive_cache_type](#) en AWS ParallelCluster versión 2.10.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

export_path

(Opcional) Especifica la ruta de Amazon S3 a la que se exporta la raíz del sistema de archivos. Cuando se especifica el [export_path](#) parámetro, no se deben especificar [fsx_backup_id](#) los parámetros [daily_automatic_backup_start_time](#), y [automatic_backup_retention_days](#) [copy_tags_to_backups](#). Esto corresponde a la propiedad [ExportPath](#). Los datos y metadatos del archivo no se exportan automáticamente a [export_path](#). Para obtener información sobre la exportación de datos y metadatos, consulte [Exportación de cambios al repositorio de datos](#) en la Guía del usuario de Amazon FSx for Lustre.

El valor predeterminado es `s3://import-bucket/FSxLustre[creation-timestamp]`, donde *import-bucket* es el bucket que se proporciona en el parámetro [import_path](#).

```
export_path = s3://bucket/folder
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

fsx_backup_id

(Opcional) Especifica el ID de la copia de seguridad que utilizar para restaurar el sistema de archivos a partir de una copia de seguridad existente. Cuando se especifica el [fsx_backup_id](#) parámetro [auto_import_policy](#), [deployment_type](#), [export_path](#), [fsx_kms_key_id](#), [import_path](#), [imported_file_chunk_size](#), [storage_capacity](#), y [per_unit_storage_throughput](#) no deben especificarse. Estos parámetros se leen de la copia de seguridad. Además, no se deben especificar [imported_file_chunk_size](#) los parámetros [import_path](#), y [auto_import_policy](#) [export_path](#).

Esto corresponde a la propiedad [BackUpId](#).

```
fsx_backup_id = backup-fedcba98
```

Note

Se ha agregado compatibilidad para [fsx_backup_id](#) en AWS ParallelCluster versión 2.8.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

fsx_fs_id

(Opcional) Asocia un sistema de archivos de FSx para Lustre ya existente.

Si se especifica esta opción, solo se utilizarán los valores de [shared_dir](#) y [fsx_fs_id](#) de la [sección \[fsx\]](#) y se omitirán todos los demás valores de la [sección \[fsx\]](#).

```
fsx_fs_id = fs-073c3803dca3e28a6
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

fsx_kms_key_id

(Opcional) Especifica el ID de clave de la clave administrada por el cliente de AWS Key Management Service (AWS KMS).

Esta clave se utiliza para cifrar los datos de su sistema de archivos en reposo.

Debe utilizarse con un [ec2_iam_role](#) personalizado. Para obtener más información, consulte [Cifrado de disco con una clave KMS personalizada](#). Esto corresponde al parámetro [KmsKeyId](#) de la Referencia de la API de Amazon FSx.

```
fsx_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Note

Se ha agregado compatibilidad para [fsx_kms_key_id](#) en AWS ParallelCluster versión 2.6.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

import_path

(Opcional) Especifica el bucket de S3 que se utiliza para cargar datos desde el sistema de archivos y sirve como bucket exportador. Para obtener más información, consulte [export_path](#). Si especifica el [import_path](#) parámetro, no se [automatic_backup_retention_days](#) deben especificar [fsx_backup_id](#) los parámetros [copy_tags_to_backupsdaily_automatic_backup_start_time](#), y. Esto corresponde al parámetro [ImportPath](#) de la Referencia de la API de Amazon FSx.

La importación se produce al crear el clúster. Para obtener más información, consulte [Importar datos del repositorio de datos](#) en la Guía del usuario de Amazon FSx for Lustre. Al importar, solo se importan los metadatos del archivo (nombre, propiedad, fecha y permisos). Los datos del archivo no se importan desde el bucket de S3 hasta que se accede al archivo por primera vez. Para obtener información sobre la precarga del contenido de los archivos, consulte [Carga previa de archivos en el sistema de archivos](#) en la Guía del usuario de Amazon FSx for Lustre.

Si no se proporciona un valor, el sistema de archivos está vacío.

```
import_path = s3://bucket
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

imported_file_chunk_size

(Opcional) Determina la cantidad de fragmentos y la cantidad máxima de datos por archivo (en MiB) que se almacenan en un único disco físico, para los archivos que se importan desde un repositorio de datos (con [import_path](#)). El número máximo de discos en los que un único archivo se puede fraccionar está limitado por el número total de los discos que forman el sistema de archivos. Cuando se especifica el [imported_file_chunk_size](#) parámetro, no se deben especificar [fsx_backup_id](#) los parámetros [automatic_backup_retention_days](#) [copy_tags_to_backupsdaily_automatic_backup_start_time](#), y. Esto corresponde a la propiedad [ImportedFileChunkSize](#).

El valor predeterminado de fragmento es 1024 (1 GiB) y puede llegar hasta 512 000 MiB (500 GiB). Los objetos de Amazon S3 tienen un tamaño máximo de 5 TB.

```
imported_file_chunk_size = 1024
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

per_unit_storage_throughput

(Necesario para los tipos de implementación **PERSISTENT_1**) Para el tipo de implementación [deployment_type](#) = PERSISTENT_1, describe el rendimiento de lectura y escritura por cada 1 tebibyte (TiB) de almacenamiento, en MB/s/TiB. La capacidad de rendimiento del sistema de archivos se calcula multiplicando la capacidad de almacenamiento del sistema (TiB) por [per_unit_storage_throughput](#) (MB/s/TiB). Para un sistema de almacenamiento de 2,4 TiB, el aprovisionamiento de 50 MB/s/TiB [per_unit_storage_throughput](#) produce una velocidad del sistema de archivos de 120 MB/s. Usted paga la cantidad de rendimiento aprovisionada. Esto corresponde a la propiedad [perUnitStorageThroughput](#).

Los valores posibles dependen del valor del [storage_type](#) ajuste.

[storage_type](#) = SSD

Los valores posibles son 50, 100 y 200.

[storage_type](#) = HDD

Los valores posibles son 12 o 40.

```
per_unit_storage_throughput = 200
```

Note

Se ha agregado compatibilidad para [per_unit_storage_throughput](#) en AWS ParallelCluster versión 2.6.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

shared_dir

(Obligatorio) Define el punto de montaje para el sistema de archivos FSx para Lustre del nodo principal y los nodos de computación.

No utilice NONE o /NONE como directorio compartido.

En el siguiente ejemplo se monta el sistema de archivos en /fsx.

```
shared_dir = /fsx
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

storage_capacity

(Obligatorio) Especifica la capacidad de almacenamiento del sistema de archivos, en GiB. Esto corresponde a la propiedad [StorageCapacity](#).

Los valores posibles de capacidad de almacenamiento varían en función del valor de [deployment_type](#).

SCRATCH_1

Los valores posibles son 1200, 2400 y cualquier múltiplo de 3600.

SCRATCH_2

Los valores posibles son 1200 y cualquier múltiplo de 2400.

PERSISTENT_1

Los valores posibles varían en función del valor de otros ajustes.

[storage_type](#) = SSD

Los valores posibles son 1200 y cualquier múltiplo de 2400.

[storage_type](#) = HDD

Los valores posibles varían en función del [per_unit_storage_throughput](#) establecimiento.

[per_unit_storage_throughput](#) = 12

Los valores posibles son cualquier múltiplo de 6000.

[per_unit_storage_throughput](#) = 40

Los valores posibles son cualquier múltiplo de 1800.

```
storage_capacity = 7200
```

Note

Para AWS ParallelCluster versión 2.5.0 y 2.5.1, [storage_capacity](#) admite los valores 1200, 2400 y cualquier múltiplo de 3600. Para las versiones anteriores a AWS ParallelCluster versión 2.5.0, [storage_capacity](#) tenía un tamaño mínimo de 3600.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

storage_type

(Opcional) Especifica el tipo de almacenamiento del sistema de archivos. Esto corresponde a la propiedad [StorageType](#). Los valores posibles son SSD y HDD. El valor predeterminado es SSD.

El tipo de almacenamiento cambia los valores posibles de otras configuraciones.

storage_type = SSD

Especifica el tipo de almacenamiento de unidades de estado sólido (SSD).

storage_type = SSD cambia los valores posibles de varios otros ajustes.

[drive_cache_type](#)

Este ajuste no se puede especificar.

[deployment_type](#)

Este ajuste se puede establecer en SCRATCH_1, SCRATCH_2 o PERSISTENT_1.

[per_unit_storage_throughput](#)

Este valor debe especificarse si [deployment_type](#) está establecido en PERSISTENT_1. Los valores posibles son 50, 10 o 200.

[storage_capacity](#)

Este valor debe especificarse. Los valores posibles varían en función del [deployment_type](#).

deployment_type = SCRATCH_1

[storage_capacity](#) puede ser 1200, 2400 o cualquier múltiplo de 3600.

deployment_type = SCRATCH_2 o deployment_type = PERSISTENT_1

[storage_capacity](#) puede ser 1200, 2400 o cualquier múltiplo de 2400.

`storage_type = HDD`

Especifica el tipo de almacenamiento de una unidad de disco duro (HDD).

`storage_type = HDD` cambia los valores posibles de otros ajustes.

[drive_cache_type](#)

Este ajuste se puede especificar.

[deployment_type](#)

Este ajuste debe establecerse en `PERSISTENT_1`.

[per_unit_storage_throughput](#)

Este valor debe especificarse. Los valores posibles son 12 o 40.

[storage_capacity](#)

Este valor debe especificarse. Los valores posibles varían en función del valor de [per_unit_storage_throughput](#).

`storage_capacity = 12`

[storage_capacity](#) puede ser cualquier múltiplo de 6000.

`storage_capacity = 40`

[storage_capacity](#) puede ser cualquier múltiplo de 1800.

```
storage_type = SSD
```

Note

La compatibilidad con la [storage_type](#) sección se añadió en versión 2.10.0.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

weekly_maintenance_start_time

(Opcional) Especifica una hora preferida para realizar el mantenimiento semanal, en la zona horaria UTC. Esto corresponde a la propiedad [WeeklyMaintenanceStartTime](#).

El formato es [día de la semana]: [hora del día]: [minuto]. Por ejemplo, lunes a medianoche es lo siguiente.

```
weekly_maintenance_start_time = 1:00:00
```

[Política de actualización: esta configuración se puede cambiar durante una actualización.](#)

Sección de **[queue]**

Define los ajustes de configuración de una sola cola. [\[queue\]las secciones](#) solo se admiten cuando [scheduler](#) se establece en slurm.

El formato es [queue *<queue-name>*]. *queue-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números y guiones (-).

```
[queue q1]
compute_resource_settings = i1,i2
placement_group = DYNAMIC
enable_efa = true
disable_hyperthreading = false
compute_type = spot
```

Note

El soporte para la [\[queue\]sección](#) se agregó en la AWS ParallelCluster versión 2.9.0.

Temas

- [compute_resource_settings](#)
- [compute_type](#)
- [disable_hyperthreading](#)
- [enable_efa](#)
- [enable_efa_gdr](#)
- [placement_group](#)

compute_resource_settings

(Obligatorio) Identifique las [\[compute_resource\]secciones](#) que contienen las configuraciones de los recursos informáticos de esta cola. Los nombres de sección deben comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

[Se admiten hasta tres \(3\) \[compute_resource\]secciones para cada sección \[queue\]](#)

Por ejemplo, la siguiente configuración especifica que se utilicen las secciones que inician `[compute_resource cr1]` y `[compute_resource cr2]`.

```
compute_resource_settings = cr1, cr2
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

compute_type

(Opcional) Define el tipo de instancias que se van a lanzar para esta cola. Esta configuración reemplaza la configuración [cluster_type](#).

Las opciones válidas son: `ondemand` y `spot`.

El valor predeterminado es `ondemand`.

Para obtener más información acerca de las instancias de Spot, consulte [Uso de instancias de spot](#).

Note

El uso de instancias de spot requiere que el rol de `AWSServiceRoleForEC2Spot` vinculado al servicio esté en su cuenta. Para crear este rol en su cuenta mediante el AWS CLI, ejecute el siguiente comando:

```
aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Para obtener más información, consulte [Función vinculada a servicios para solicitudes de instancias puntuales](#) en la Guía del usuario de Amazon EC2.

En el siguiente ejemplo, se utilizan los nodos `SpotInstances` de cómputo de esta cola.

```
compute_type = spot
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

disable_hyperthreading

(Opcional) Desactive los hipersubprocesos en los nodos de esta cola. No todos los tipos de instancias pueden deshabilitar la tecnología Hyper-Threading. Para obtener una lista de los tipos de instancias que admiten la desactivación del hiperproceso, consulte [Núcleos y subprocesos de CPU para cada núcleo de CPU por tipo de instancia](#) en la Guía del usuario de Amazon EC2. Si la configuración de [disable_hyperthreading](#) de la [\[cluster\]sección](#) está definida, no se puede definir esta configuración.

El valor predeterminado es `false`.

```
disable_hyperthreading = true
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

enable_efa

(Opcional) Si se establece en `true`, especifique que el Elastic Fabric Adapter (EFA) esté habilitado para los nodos de esta cola. Para ver la lista de instancias de EC2 compatibles con EFA, consulte [Tipos de instancias admitidas](#) en la Guía del usuario para instancias de Linux de Amazon EC2. Si la configuración de [enable_efa](#) de la [\[cluster\]sección](#) está definida, no se puede definir esta configuración. Se debe utilizar un grupo de ubicación del clúster para minimizar las latencias entre instancias. Para obtener más información, consulte [placement](#) y [placement_group](#).

```
enable_efa = true
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

enable_efa_gdr

(Opcional) A partir de la AWS ParallelCluster versión 2.11.3, esta configuración no tiene efecto. La compatibilidad con el Elastic Fabric Adapter (EFA) para GPUDirect RDMA (acceso directo a

memoria) está habilitada para los nodos de procesamiento y siempre está habilitada si el tipo de instancia lo admite.

Note

AWS ParallelCluster de la versión 2.10.0 a la 2.11.2: Si `true`, especifica que el Elastic Fabric Adapter (EFA) GPUDirect RDMA (acceso remoto directo a memoria) está habilitado para los nodos de esta cola. Si se configura a `true`, es necesario que la configuración de [enable_efa](#) esté establecida en `true`. EFA GPUDirect RDMA es compatible con los siguientes tipos de instancias (i) en estos sistemas operativos (p4d.24xlarge, alinux2, centos7, ubuntu1804 o ubuntu2004). Si la configuración de [enable_efa_gdr](#) de la [\[cluster\]sección](#) está definida, no se puede definir esta configuración. Se debe utilizar un grupo de ubicación del clúster para minimizar las latencias entre instancias. Para obtener más información, consulte [placement](#) y [placement_group](#).

El valor predeterminado es `false`.

```
enable_efa_gdr = true
```

Note

El soporte para `enable_efa_gdr` se agregó en la AWS ParallelCluster versión 2.10.0.

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

placement_group

(Opcional) Si está presente, define el grupo de ubicaciones de esta cola. Esta configuración reemplaza la configuración [placement_group](#).

Las opciones válidas son las siguientes:

- DYNAMIC
- Un nombre de grupo con ubicación en clúster de Amazon EC2 existente

Cuando se establece en DYNAMIC, se crea y elimina un grupo de ubicación único para esta cola como parte de la pila del clúster.

Para obtener más información sobre los grupos de ubicación, consulte [los grupos de ubicación](#) en la Guía del usuario de Amazon EC2. Si se usa el mismo grupo de ubicación para distintos tipos de instancias, es más probable que la solicitud no se realice correctamente debido a un error de capacidad insuficiente. Para obtener más información, consulte [Capacidad de instancia insuficiente](#) en la Guía del usuario de Amazon EC2.

No hay valor predeterminado.

No todos los tipos de instancias admiten grupos de ubicación en clúster. Por ejemplo, `t2.micro` no admite grupos con ubicación en clúster. Para obtener información sobre la lista de tipos de instancias que admiten grupos de ubicación en clústeres, consulte [Reglas y limitaciones de los grupos de ubicación en clústeres](#) en la Guía del usuario de Amazon EC2. Consulte [Problemas con los grupos de ubicación y el lanzamiento de instancias](#) para obtener sugerencias al trabajar con grupos de ubicación.

```
placement_group = DYNAMIC
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

Sección de **[raid]**

Define las opciones de configuración para una matriz de RAID que se ha creado a partir de una serie de volúmenes de Amazon EBS idénticos. La unidad RAID se monta en el nodo principal y se exporta a nodos de computación con NFS.

El formato es `[raid raid-name]`. *raid-name* debe comenzar por una letra, contener no más de 30 caracteres y contener solo letras, números, guiones (-) y guiones bajos (_).

```
[raid rs]
shared_dir = raid
raid_type = 1
num_of_raid_volumes = 2
encrypted = true
```

Temas

- [shared_dir](#)
- [ebs_kms_key_id](#)
- [encrypted](#)
- [num_of_raid_volumes](#)
- [raid_type](#)
- [volume_iops](#)
- [volume_size](#)
- [volume_throughput](#)
- [volume_type](#)

shared_dir

(Obligatorio) Define el punto de montaje de la matriz de RAID en los nodos principal y de computación.

La unidad RAID solo se crea si se especifica este parámetro.

No utilice NONE o /NONE como directorio compartido.

En el siguiente ejemplo, se monta la matriz en /raid.

```
shared_dir = raid
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

ebs_kms_key_id

(Opcional) Especifica una AWS KMS clave personalizada que se utilizará para el cifrado.

Este parámetro se debe usar junto con la opción `encrypted = true` y debe tener un [ec2_iam_role](#) personalizado.

Para obtener más información, consulte [Cifrado de disco con una clave KMS personalizada](#).

```
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

encrypted

(Opcional) Especifique si el sistema de archivos está cifrado.

El valor predeterminado es `false`.

```
encrypted = false
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

num_of_raid_volumes

(Opcional) Define el número de volúmenes de Amazon EBS desde los que se montará la matriz de RAID.

Número mínimo de volúmenes = 2.

Número máximo de volúmenes = 5.

El valor predeterminado es 2.

```
num_of_raid_volumes = 2
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

raid_type

(Obligatorio) Define el tipo de RAID para la matriz de RAID.

La unidad RAID solo se crea si se especifica este parámetro.

Las opciones válidas son las siguientes:

- 0
- 1

Para obtener más información sobre los tipos de RAID, consulte la [información sobre RAID](#) en la Guía del usuario de Amazon EC2.

El siguiente ejemplo crea una matriz de RAID 0:

```
raid_type = 0
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

volume_iops

(Opcional) Define el número de IOPS para volúmenes de tipo io1, io2 y gp3.

El valor predeterminado, los valores admitidos y la proporción de datos `volume_iops` a `volume_size` varía entre [volume_type](#) y [volume_size](#).

`volume_type = io1`

`volume_iops` predeterminado = 100

Valores admitidos `volume_iops` = 100–64 000 †

Relación máxima `volume_iops` a `volume_size` = 50 IOPS por GiB. 5000 IOPS requieren un `volume_size` de al menos 100 GiB.

`volume_type = io2`

`volume_iops` predeterminado = 100

Valores admitidos `volume_iops` = 100–64 000 (256 000 para los volúmenes de io2 Block Express) †

Relación máxima `volume_iops` a `volume_size` = 500 IOPS por GiB. 5000 IOPS requieren un `volume_size` de al menos 10 GiB.

`volume_type = gp3`

`volume_iops` predeterminado = 3000

Valores admitidos `volume_iops` = 3000–16 000

Relación máxima `volume_iops` a `volume_size` = 500 IOPS por GiB. 5000 IOPS requieren un `volume_size` de al menos 10 GiB.

```
volume_iops = 3000
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

† Las IOPS máximas solo se garantizan en [las instancias creadas en el sistema Nitro](#) aprovisionadas con más de 32 000 IOPS. Otras instancias garantizan hasta 32,000 IOPS. Es posible que los volúmenes io1 más antiguos no alcancen el rendimiento máximo a menos que [modifique el volumen](#). io2 Los volúmenes de Block Express admiten valores de `volume_iops` de hasta 256 000. Para obtener más información, consulte [io2Bloque de volúmenes Express \(en versión preliminar\)](#) en la Guía del usuario de Amazon EC2.

`volume_size`

(Opcional) Define el tamaño del volumen que se va a crear, en GiB.

El valor predeterminado y los valores admitidos varía en [volume_type](#).

`volume_type = standard`

Predeterminado `volume_size = 20 GiB`

Valores admitidos `volume_size = 1–1024 GiB`

`volume_type = gp2, io1, io2 y gp3`

Predeterminado `volume_size = 20 GiB`

Valores admitidos `volume_size = 1–16 384 GiB`

`volume_type = sc1 y st1`

Predeterminado `volume_size = 500 GiB`

Valores admitidos `volume_size = 500–16 384 GiB`

```
volume_size = 20
```

Note

Antes de AWS ParallelCluster la versión 2.10.1, el valor predeterminado para todos los tipos de volumen era de 20 GiB.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

volume_throughput

(Opcional) Define el rendimiento de los tipos de volumen gp3, en MiB/s.

El valor predeterminado es 125.

Valores admitidos `volume_throughput` = 125–1000 MiB/s

La relación entre `volume_throughput` y `volume_iops` no puede ser superior a 0,25. El rendimiento máximo de 1000 MiB/s requiere que la configuración de `volume_iops` sea de al menos 4000.

```
volume_throughput = 1000
```

Note

El soporte para `volume_throughput` se agregó en la AWS ParallelCluster versión 2.10.1.

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

volume_type

(Opcional) Define el tipo de volumen que se va a crear.

Las opciones válidas son las siguientes:

gp2, gp3

General Purpose SSD

io1, io2

Provisioned IOPS SSD

st1

HDD con rendimiento optimizado

sc1

HDD en frío

standard

Magnéticos de generaciones anteriores

Para obtener más información, consulte [Tipos de volúmenes de Amazon EBS](#) en la Guía del usuario de Amazon EC2.

El valor predeterminado es gp2.

```
volume_type = io2
```

Note

Support gp3 y io2 se agregó en la AWS ParallelCluster versión 2.10.1.

Política de actualización: si se cambia esta configuración, no se permite la actualización.

Sección [scaling]

Temas

- [scaledown_idletime](#)

Especifica la configuración que define cómo se escalan los nodos de computación.

El formato es [scaling *scaling-name*]. *scaling-name* debe comenzar por una letra, contener no más de 30 caracteres y solo letras, números, guiones (-) y guiones bajos (_).

```
[scaling custom]
scaledown_idletime = 10
```

scaledown_idletime

(Opcional) Especifica la cantidad de minutos sin realizar un trabajo, transcurridos los cuales el nodo de computación termina.

Este parámetro no se usa si awsbatch es el programador.

El valor predeterminado es 10.

```
scaledown_idletime = 10
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

Sección de [vpc]

Especifica los ajustes VPC de configuración de Amazon. Para obtener más información VPCs, consulta [¿Qué es AmazonVPC?](#) y [prácticas recomendadas de seguridad para ti VPC](#) en la Guía del VPC usuario de Amazon.

El formato es [vpc *vpc-name*]. *vpc-name* debe empezar por una letra, no contener más de 30 caracteres y contener únicamente letras, números, guiones (-) y guiones bajos (_).

```
[vpc public]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-xxxxxx
```

Temas

- [additional_sg](#)
- [compute_subnet_cidr](#)
- [compute_subnet_id](#)
- [master_subnet_id](#)
- [ssh_from](#)
- [use_public_ips](#)
- [vpc_id](#)
- [vpc_security_group_id](#)

additional_sg

(Opcional) Proporciona un ID de grupo VPC de seguridad de Amazon adicional para todas las instancias.

No hay valor predeterminado.

```
additional_sg = sg-xxxxxx
```

compute_subnet_cidr

(Opcional) Especifica un bloque de enrutamiento entre dominios sin clase (CIDR). Utilice este parámetro si desea AWS ParallelCluster crear una subred de procesamiento.

```
compute_subnet_cidr = 10.0.100.0/24
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

compute_subnet_id

(Opcional) Especifique el ID de una subred existente en la que se van a aprovisionar los nodos de computación.

Si no se especifica, [compute_subnet_id](#) utiliza el valor de [master_subnet_id](#).

Si la subred es privada, debe configurarla NAT para el acceso web.

```
compute_subnet_id = subnet-xxxxxx
```

[Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.](#)

master_subnet_id

(Obligatorio) Especifique el ID de una subred existente en la que se va a aprovisionar el nodo maestro.

```
master_subnet_id = subnet-xxxxxx
```

[Política de actualización: si se cambia esta configuración, no se permite la actualización.](#)

ssh_from

(Opcional) Especifica un rango CIDR de IP con formato L desde el que permitir SSH el acceso.

Este parámetro solo se usa cuando se AWS ParallelCluster crea el grupo de seguridad.

El valor predeterminado es 0.0.0.0/0.

```
ssh_from = 0.0.0.0/0
```

Política de actualización: esta configuración se puede cambiar durante una actualización.

use_public_ips

(Opcional) Define si se asignarán o no direcciones IP públicas a instancias de computación.

Si se establece en `true`, se asocia una dirección IP elástica al nodo principal.

Si se establece en `false`, el nodo principal tiene una IP pública (o no) según el valor del parámetro de configuración de subred "Auto-assign Public IP" (Asignar automáticamente IP pública).

Para ver ejemplos, consulte la [configuración de redes](#).

El valor predeterminado es `true`.

```
use_public_ips = true
```

Important

De forma predeterminada, todas Cuentas de AWS están limitadas a cinco (5) direcciones IP elásticas para cada una Región de AWS. Para obtener más información, consulta el [límite de direcciones IP elásticas](#) en la Guía EC2 del usuario de Amazon.

Política de actualización: la flota de computación debe estar detenida para poder cambiar esta configuración y proceder a la actualización.

vpc_id

(Obligatorio) Especifica el ID de Amazon VPC en el que se va a aprovisionar el clúster.

```
vpc_id = vpc-xxxxxx
```

Política de actualización: si se cambia esta configuración, no se permite la actualización.

vpc_security_group_id

(Opcional) Especifique el uso de un grupo de seguridad existente para todas las instancias.

No hay valor predeterminado.

```
vpc_security_group_id = sg-xxxxxx
```

El grupo de seguridad creado por AWS ParallelCluster permite el SSH acceso mediante el puerto 22 desde las direcciones especificadas en la [ssh_from](#) configuración o desde todas IPv4 las direcciones (0.0.0.0/0) si no se especifica la [ssh_from](#) configuración. Si Amazon DCV está activado, el grupo de seguridad permite el acceso a Amazon DCV mediante el puerto 8443 (o lo que especifique la [port](#) configuración) desde las direcciones especificadas en la [access_from](#) configuración, o todas IPv4 las direcciones (0.0.0.0/0) si no se especifica la [access_from](#) configuración.

Warning

Puede cambiar el valor de este parámetro y actualizar el clúster si

[\[cluster\]fsx_settings](#) no está especificado, o ambas opciones, `fsx_settings` y si se ha especificado un sistema de archivos externo existente FSx para [fsx-fs-id](#) Lustre. [\[fsx fs\]](#)

No puede cambiar el valor de este parámetro si se especifica un sistema de archivos AWS ParallelCluster gestionado FSx para Lustre en `fsx_settings` y. `[fsx fs]`

[Política de actualización: si los sistemas de archivos AWS ParallelCluster gestionados de Amazon FSx for Lustre no se especifican en la configuración, esta configuración se puede cambiar durante una actualización.](#)

Ejemplos

Los siguientes ejemplos de configuraciones muestran AWS ParallelCluster las configuraciones que utilizan planificadores Slurm, Torque y AWS Batch.

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de planificadores SGE o Torque.

Contenido

- [Slurm Workload Manager \(slurm\)](#)
- [Son of Grid Engine \(sge\) y Torque Resource Manager \(torque\)](#)
- [AWS Batch \(awsbatch\)](#)

Slurm Workload Manager (**slurm**)

El siguiente ejemplo lanza un clúster con el programador `slurm`. La configuración de ejemplo lanza 1 clúster con 2 colas de trabajos. La primera cola, `spot`, inicialmente tiene 2 instancias de `spot t3.micro` disponibles. Puede ampliarse hasta un máximo de 10 instancias y reducirse hasta un mínimo de 1 instancia cuando no se ha ejecutado ningún trabajo durante 10 minutos (se puede ajustar mediante la configuración [scaledown_idletime](#)). La segunda cola, `ondemand`, comienza sin instancias y puede ampliarse hasta un máximo de 5 instancias `t3.micro` bajo demanda.

```
[global]
update_check = true
sanity_check = true
cluster_template = slurm

[aws]
aws_region_name = <your Región de AWS>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster slurm]
key_name = <your EC2 keypair name>
base_os = alinux2 # optional, defaults to alinux2
scheduler = slurm
master_instance_type = t3.micro # optional, defaults to t3.micro
vpc_settings = public
queue_settings = spot,ondemand

[queue spot]
compute_resource_settings = spot_i1
compute_type = spot # optional, defaults to ondemand

[compute_resource spot_i1]
instance_type = t3.micro
min_count = 1 # optional, defaults to 0
initial_count = 2 # optional, defaults to 0
```

```
[queue ondemand]
compute_resource_settings = ondemand_i1

[compute_resource ondemand_i1]
instance_type = t3.micro
max_count = 5 # optional, defaults to 10
```

Son of Grid Engine (**sge**) y Torque Resource Manager (**torque**)

Note

Este ejemplo solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de planificadores SGE o Torque.

El siguiente ejemplo lanza un clúster con el programador torque o sge. Para usar SGE, cambie `scheduler = torque` a `scheduler = sge`. La configuración de ejemplo le permite tener un máximo de 5 nodos simultáneos y baja a dos cuando no se ha ejecutado ningún trabajo durante un plazo de 10 minutos.

```
[global]
update_check = true
sanity_check = true
cluster_template = torque

[aws]
aws_region_name = <your Región de AWS>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster torque]
key_name = <your EC2 keypair name>but they aren't eligible for future updates
base_os = alinux2 # optional, defaults to alinux2
scheduler = torque # optional, defaults to sge
master_instance_type = t3.micro # optional, defaults to t3.micro
vpc_settings = public
initial_queue_size = 2 # optional, defaults to 0
```

```
maintain_initial_size = true      # optional, defaults to false
max_queue_size = 5               # optional, defaults to 10
```

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de planificadores SGE o Torque. Si usa estas versiones, puede seguir usándolas o bien puede solucionar problemas con el soporte de los equipos de AWS servicio y AWS soporte.

AWS Batch (**awsbatch**)

El siguiente ejemplo lanza un clúster con el programador `awsbatch`. Se establece para elegir el tipo de instancia óptimo, en función de las necesidades de sus recursos de trabajo.

La configuración de ejemplo le permite tener un máximo de 40 vCPU simultáneas y baja a cero cuando no se ha ejecutado ningún trabajo durante un plazo de 10 minutos (ajustable usando la configuración [scaledown_idletime](#)).

```
[global]
update_check = true
sanity_check = true
cluster_template = awsbatch

[aws]
aws_region_name = <your Región de AWS>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster awsbatch]
scheduler = awsbatch
compute_instance_type = optimal # optional, defaults to optimal
min_vcpus = 0                   # optional, defaults to 0
desired_vcpus = 0               # optional, defaults to 4
max_vcpus = 40                  # optional, defaults to 20
base_os = alinux2               # optional, defaults to alinux2, controls the base_os
of                               # the head node and the docker image for the compute
fleet
```

```
key_name = <your EC2 keypair name>  
vpc_settings = public
```

Cómo funciona AWS ParallelCluster

AWS ParallelCluster se creó no solo como una forma de administrar clústeres, sino también como referencia sobre cómo usar los servicios de AWS para crear el entorno de HPC.

Temas

- [Procesos de AWS ParallelCluster](#)
- [AWS servicios utilizados por AWS ParallelCluster](#)
- [Auto Scaling de AWS ParallelCluster](#)

Procesos de AWS ParallelCluster

Esta sección se aplica únicamente a clústeres de HPC que se implementan con uno de los programadores de trabajos tradicionales compatibles (SGE, Slurm o Torque). Al usarse con estos programadores, AWS ParallelCluster administra el aprovisionamiento del nodo de computación y su eliminación interactuando con el grupo de escalado automático y el programador de trabajos subyacente.

Para clústeres de HPC basados en AWS Batch, AWS ParallelCluster se basa en las capacidades que proporciona por la administración de nodos de computación.

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de planificadores SGE o Torque. Puede seguir utilizándolos en las versiones anteriores a la 2.11.4 inclusive, pero no son aptos para recibir actualizaciones futuras ni asistencia para la solución de problemas por parte de los equipos de servicio AWS y soporte AWS.

Temas

- [SGE and Torque integration processes](#)
- [Slurm integration processes](#)

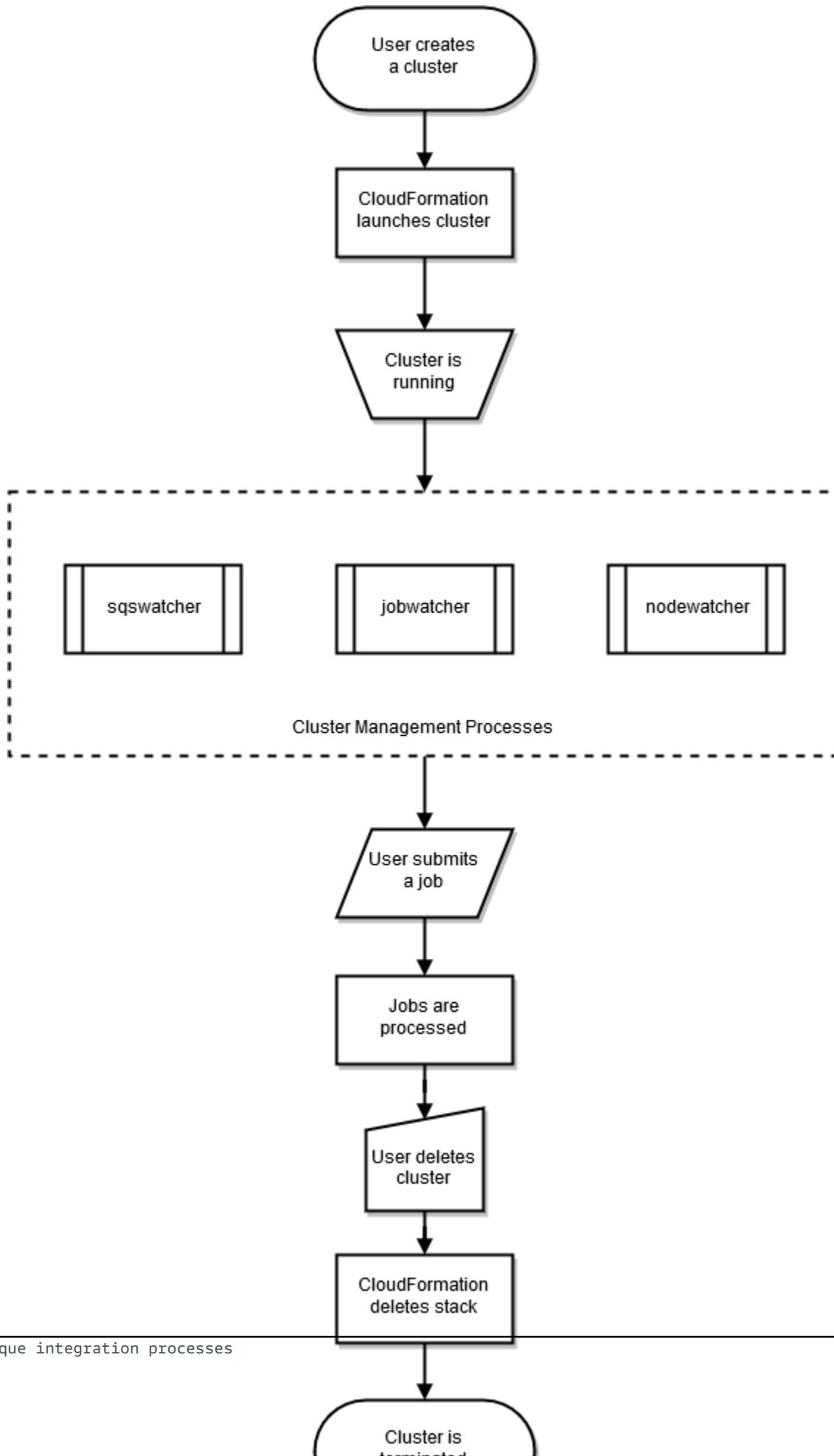
SGE and Torque integration processes

Note

Esta sección solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de Torque planificadores SGE y Amazon SNS y Amazon SQS.

Información general

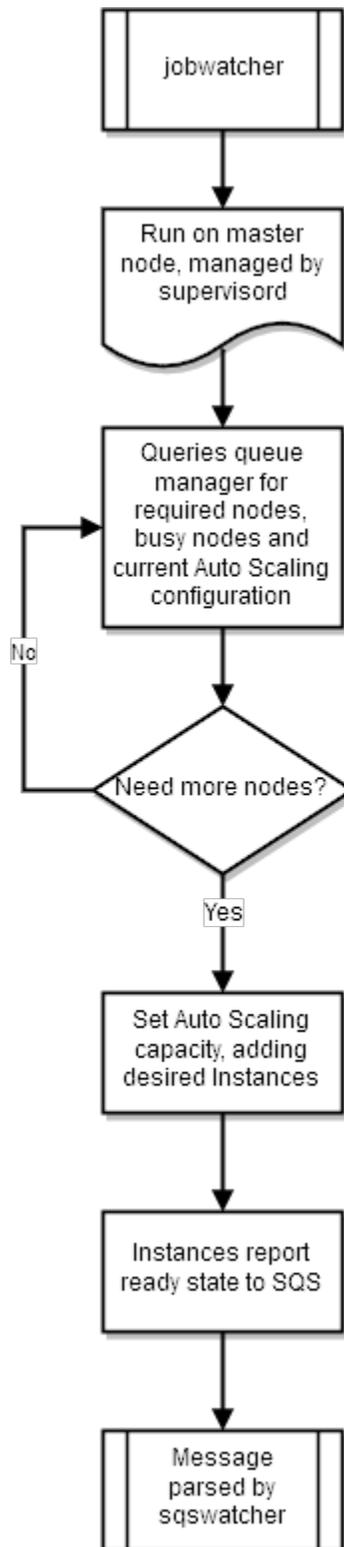
El ciclo de vida de un clúster comienza después de crearlo el usuario. Normalmente, un clúster se crea a partir de la interfaz de línea de comandos (CLI). Después de crearse, un clúster existe hasta que se elimina. Los demonios de AWS ParallelCluster se ejecutan en los nodos de clúster, principalmente para administrar la elasticidad del clúster HPC. En el siguiente diagrama se muestran un flujo de trabajo de usuario y el ciclo de vida del clúster. En las secciones que aparecen a continuación se describen los demonios de AWS ParallelCluster que se utilizan para administrar el clúster.



Con SGE y programadores Torque, AWS ParallelCluster use los procesos `nodewatcher`, `jobwatcher` y `sqswatcher`.

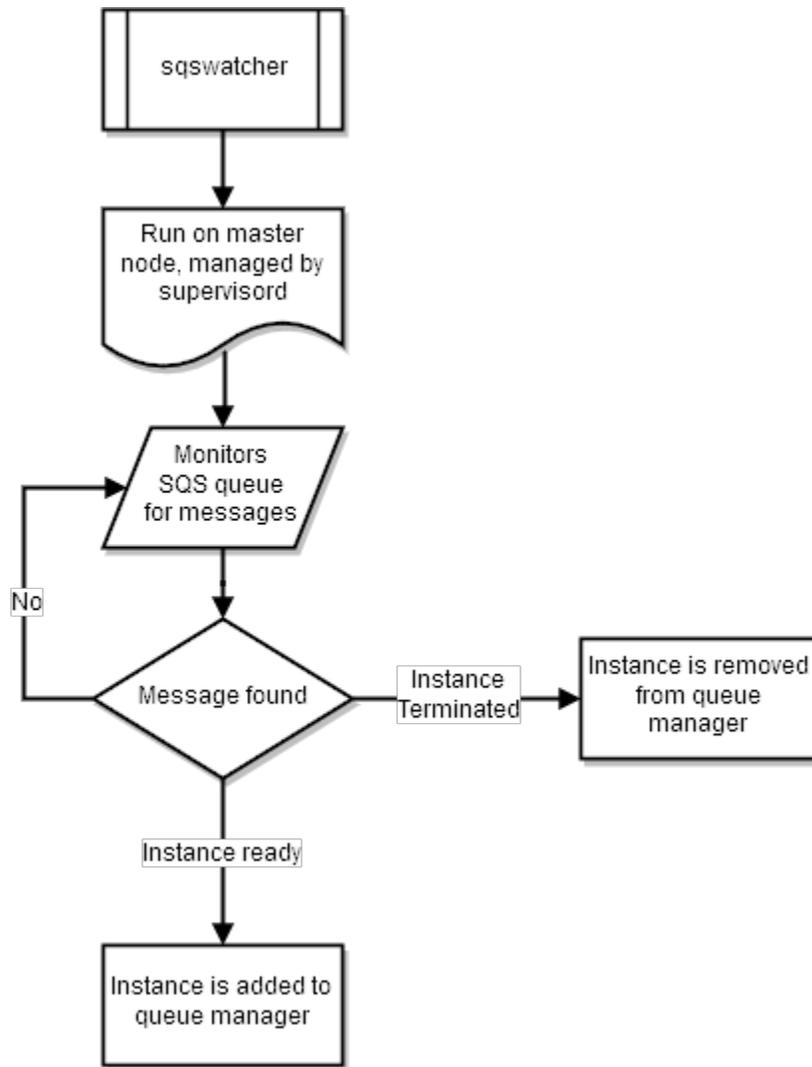
jobwatcher

Cuando se ejecuta un clúster, un proceso propiedad del usuario raíz supervisa el programador configurado (SGE o Torque). Cada minuto evalúa la cola para decidir cuándo escalar verticalmente.



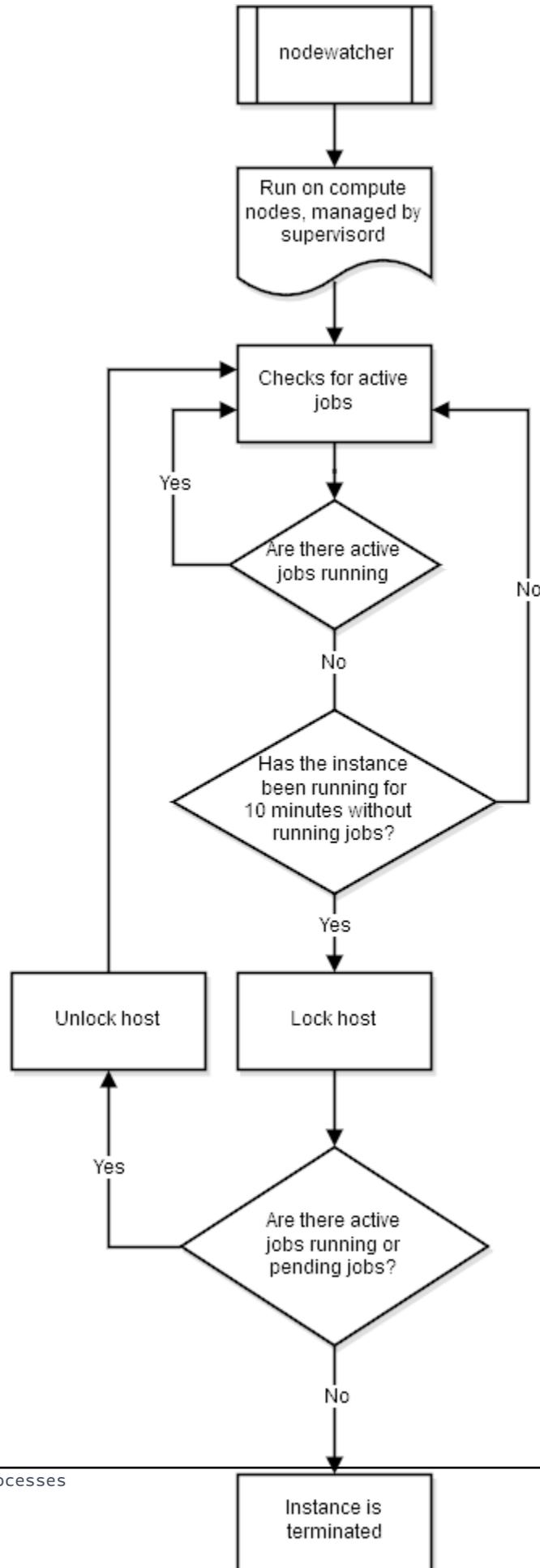
sqswatcher

El proceso monitoriza los mensajes de `sqswatcher` que el escalado automático envía para notificarle los cambios de estado en el clúster. Cuando una instancia está online, envía un mensaje "instancia lista" a Amazon SQS. `sqswatcher` recoge este mensaje y se ejecuta en el nodo principal. Estos mensajes se utilizan para notificar al administrador de la cola que hay instancias nuevas online o que se han terminado instancias, de modo que se puedan añadir o eliminar de la cola.



nodewatcher

El proceso `nodewatcher` se ejecuta en cada nodo de la flota de computación. Transcurrido el periodo `scaledown_idletime`, tal como define el usuario, la instancia se termina.



Slurm integration processes

Con programadores Slurm, AWS ParallelCluster usa los procesos `clustermgtd` y `computemgt`.

`clustermgtd`

Los clústeres que se ejecutan en modo heterogéneo (indicado mediante la especificación de un valor [queue_settings](#)) tienen un proceso daemon de administración de clústeres (`clustermgtd`) que se ejecuta en el nodo principal. Estas tareas las realiza el daemon de administración de clústeres.

- Limpieza de particiones inactivas
- Administración de la capacidad estática: asegúrese de que la capacidad estática esté siempre activa y en buen estado
- Programador de sincronización con Amazon EC2.
- Limpieza de instancias huérfanas
- Restaure el estado del nodo del programador en la terminación de Amazon EC2 que se produce fuera del flujo de trabajo suspendido
- Administración de instancias de Amazon EC2 en mal estado (no se cumplen las comprobaciones de estado de Amazon EC2)
- Administración de eventos de mantenimiento programados
- La administración de los nodos del programador no funciona correctamente (comprobaciones de estado fallidas)

`computemgt`

Los clústeres que se ejecutan en modo heterogéneo (indicado mediante la especificación de un valor [queue_settings](#)) tienen procesos daemon (`computemgt`) de administración de cómputo que se ejecutan en cada uno de los nodos de cómputo. Cada cinco (5) minutos, el daemon de administración de cómputo confirma que se puede acceder al nodo principal y que está en buen estado. Si transcurren cinco (5) minutos durante los cuales no se pueda acceder al nodo principal o éste no esté en buen estado, el nodo de procesamiento se cierra.

AWS servicios utilizados por AWS ParallelCluster

Los siguientes servicios de Amazon Web Services (AWS) son utilizados por AWS ParallelCluster.

Temas

- [AWS Auto Scaling](#)
- [AWS Batch](#)
- [AWS CloudFormation](#)
- [Amazon CloudWatch](#)
- [Amazon CloudWatch Logs](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx para Lustre](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon DCV](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service](#)
- [Amazon Simple Queue Service](#)
- [Amazon Simple Storage Service](#)
- [Amazon VPC](#)

AWS Auto Scaling

Note

Esta sección solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de AWS Auto Scaling

AWS Auto Scaling es un servicio que supervisa sus aplicaciones y ajusta automáticamente la capacidad en función de sus requisitos de servicio específicos y cambiantes. Este servicio administra sus ComputeFleet instancias como un grupo de Auto Scaling. El grupo se puede gestionar de forma

elástica en función de los cambios en la carga de trabajo o se puede fijar de forma estática mediante las configuraciones iniciales de las instancias.

AWS Auto Scaling se usa con ComputeFleet instancias, pero no con AWS Batch clústeres.

Para obtener más información acerca de AWS Auto Scaling, consulte <https://aws.amazon.com/autoscaling/> y <https://docs.aws.amazon.com/autoscaling/>.

AWS Batch

AWS Batch es un servicio AWS gestionado de programación de tareas. Aprovisiona de forma dinámica la cantidad y el tipo óptimos de recursos de cómputo (por ejemplo, CPU instancias optimizadas para la memoria) en los clústeres. Estos recursos se aprovisionan en función de los requisitos específicos de sus trabajos por lotes, incluidos los requisitos de volumen. Con AWS Batch esto, no necesita instalar ni administrar software de computación por lotes adicional ni clústeres de servidores para ejecutar sus trabajos de manera eficaz.

AWS Batch se usa solo con AWS Batch clústeres.

Para obtener más información sobre AWS Batch, consulte <https://aws.amazon.com/batch/> y <https://docs.aws.amazon.com/batch/>.

AWS CloudFormation

AWS CloudFormation es un infrastructure-as-code servicio que proporciona un lenguaje común para modelar AWS y aprovisionar recursos de aplicaciones de terceros en su entorno de nube. Es el servicio principal utilizado por AWS ParallelCluster. Cada clúster AWS ParallelCluster se representa como una pila y todos los recursos que necesita se definen en la AWS ParallelCluster AWS CloudFormation plantilla. En la mayoría de los casos, AWS ParallelCluster CLI los comandos se corresponden directamente con los comandos de AWS CloudFormation pila, como los comandos de creación, actualización y eliminación. Las instancias que se lanzan dentro de un clúster realizan HTTPS llamadas al AWS CloudFormation punto final en el Región de AWS que se lanza el clúster.

Para obtener más información sobre AWS CloudFormation, consulte <https://aws.amazon.com/cloudformation/> y <https://docs.aws.amazon.com/cloudformation/>.

Amazon CloudWatch

Amazon CloudWatch (CloudWatch) es un servicio de monitoreo y observabilidad que le proporciona datos e información procesable. Esta información se puede utilizar para supervisar las aplicaciones, responder a los cambios en el rendimiento y a las excepciones del servicio y optimizar la utilización

de los recursos. En AWS ParallelCluster, CloudWatch se usa como panel de control, para monitorear y registrar los pasos de creación de imágenes de Docker y el resultado de los trabajos. AWS Batch

Antes de AWS ParallelCluster la versión 2.10.0, solo CloudWatch se utilizaba con clústeres. AWS Batch

Para obtener más información al respecto CloudWatch, consulte <https://aws.amazon.com/cloudwatch/> y <https://docs.aws.amazon.com/cloudwatch/>

Amazon CloudWatch Logs

Amazon CloudWatch Logs (CloudWatch Logs) es una de las funciones principales de Amazon CloudWatch. Puede usarla para monitorear, almacenar, ver y buscar en los archivos de registro muchos de los componentes utilizados por AWS ParallelCluster.

Antes de AWS ParallelCluster la versión 2.6.0, CloudWatch Logs solo se utilizaba con AWS Batch clústeres.

Para obtener más información, consulte [Integración con Amazon CloudWatch Logs](#).

AWS CodeBuild

AWS CodeBuild (CodeBuild) es un servicio de integración continua AWS gestionado que cumple con el código fuente, ejecuta pruebas y produce paquetes de software listos para su implementación. En AWS ParallelCluster, CodeBuild se utiliza para crear imágenes de Docker de forma automática y transparente cuando se crean clústeres.

CodeBuild se usa solo con AWS Batch clústeres.

Para obtener más información sobre CodeBuild, consulte <https://aws.amazon.com/codebuild/> y <https://docs.aws.amazon.com/codebuild/>.

Amazon DynamoDB

Amazon DynamoDB (DynamoDB) es un servicio rápido y flexible sin bases de datos. SQL Se utiliza para almacenar la información del estado mínimo del clúster. El nodo principal rastrea las instancias aprovisionadas en una tabla de DynamoDB.

DynamoDB no se usa con clústeres. AWS Batch

Para obtener más información sobre DynamoDB, consulte y <https://aws.amazon.com/dynamodb/> <https://docs.aws.amazon.com/dynamodb/>

Amazon Elastic Block Store

Amazon Elastic Block Store (AmazonEBS) es un servicio de almacenamiento en bloques de alto rendimiento que proporciona almacenamiento persistente para volúmenes compartidos. Todos los EBS ajustes de Amazon se pueden transferir a través de la configuración. EBS Los volúmenes de Amazon se pueden inicializar vacíos o a partir de una EBS instantánea de Amazon existente.

Para obtener más información sobre AmazonEBS, consulta <https://aws.amazon.com/ebs/> y <https://docs.aws.amazon.com/ebs/>.

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (AmazonEC2) proporciona la capacidad informática para AWS ParallelCluster. Los nodos principal y de cómputo son EC2 instancias de Amazon. HVM Se puede seleccionar cualquier tipo de instancia compatible. Los nodos principal y de computación pueden ser de distintos tipos de instancias. Además, si se utilizan varias colas, algunos o todos los nodos de computación también se pueden lanzar como una instancia de spot. Los volúmenes del almacén de instancias que se encuentran en las instancias se montan como LVM volúmenes seccionados.

Para obtener más información sobre AmazonEC2, consulta <https://aws.amazon.com/ec2/> y <https://docs.aws.amazon.com/ec2/>.

Amazon Elastic Container Registry

Amazon Elastic Container Registry (AmazonECR) es un registro de contenedores de Docker totalmente gestionado que facilita el almacenamiento, la administración y la implementación de imágenes de contenedores de Docker. En AWS ParallelCluster, Amazon ECR almacena las imágenes de Docker que se crean al crear los clústeres. Luego, las imágenes de Docker se utilizan AWS Batch para ejecutar los contenedores de los trabajos enviados.

Amazon solo ECR se usa con AWS Batch clústeres.

Para obtener más información, consulte <https://aws.amazon.com/ecr/> y <https://docs.aws.amazon.com/ecr/>.

Amazon EFS

Amazon Elastic File System (AmazonEFS) proporciona un sistema de NFS archivos elástico simple, escalable y totalmente gestionado para su uso con Nube de AWS servicios y recursos locales.

Amazon EFS se utiliza cuando se especifica la [efs_settings](#) configuración y hace referencia a una [\[efs\]sección](#). Support for Amazon EFS se añadió en la AWS ParallelCluster versión 2.1.0.

Para obtener más información sobre AmazonEFS, consulta <https://aws.amazon.com/efs/> y <https://docs.aws.amazon.com/efs/>.

Amazon FSx para Lustre

FSxfor Lustre proporciona un sistema de archivos de alto rendimiento que utiliza el sistema de archivos Lustre de código abierto. FSxfor Lustre se usa cuando se especifica la [fsx_settings configuración y hace referencia a una sección](#). [\[fsx\]](#) El soporte FSx para Lustre se agregó en la AWS ParallelCluster versión 2.2.1.

Para obtener más información acerca de FSx Lustre, consulte <https://aws.amazon.com/fsx/lustre/> y <https://docs.aws.amazon.com/fsx/>

AWS Identity and Access Management

AWS Identity and Access Management (IAM) se usa dentro AWS ParallelCluster para proporcionar un IAM rol de privilegios mínimos a Amazon EC2 para la instancia que es específico de cada clúster individual. AWS ParallelCluster las instancias solo tienen acceso a las API llamadas específicas que se requieren para implementar y administrar el clúster.

Con AWS Batch los clústeres, también se crean IAM roles para los componentes que intervienen en el proceso de creación de imágenes de Docker cuando se crean los clústeres. Estos componentes incluyen las funciones de Lambda que permiten añadir y eliminar imágenes de Docker desde y hacia el repositorio de Amazon. ECR También incluyen las funciones que permiten eliminar el bucket de Amazon S3 creado para el clúster y el CodeBuild proyecto. También hay funciones para AWS Batch los recursos, las instancias y los trabajos.

Para obtener más información sobreIAM, consulte <https://aws.amazon.com/iam/> y <https://docs.aws.amazon.com/iam/>.

AWS Lambda

AWS Lambda (Lambda) ejecuta las funciones que organizan la creación de imágenes de Docker. Lambda también gestiona la limpieza de los recursos del clúster personalizados, como las imágenes de Docker almacenadas en el ECR repositorio de Amazon y en Amazon S3.

Para obtener más información sobre Lambda, consulte <https://aws.amazon.com/lambda/y>. <https://docs.aws.amazon.com/lambda/>

Amazon DCV

Amazon DCV es un protocolo de visualización remota de alto rendimiento que proporciona una forma segura de entregar escritorios remotos y streaming de aplicaciones a cualquier dispositivo en diferentes condiciones de red. Amazon DCV se utiliza cuando se especifica la [dcv_settings](#) configuración y hace referencia a una [\[dcv\]sección](#). Support for Amazon DCV se añadió en la AWS ParallelCluster versión 2.5.0.

Para obtener más información sobre AmazonDCV, consulta <https://aws.amazon.com/hpc/dcv/> y <https://docs.aws.amazon.com/dcv/>

Amazon Route 53

Amazon Route 53 (Route 53) se utiliza para crear zonas alojadas con nombres de host y nombres de dominio totalmente cualificados para cada uno de los nodos de computación.

Para obtener más información sobre Route 53, consulte <https://aws.amazon.com/route53/y>. <https://docs.aws.amazon.com/route53/>

Amazon Simple Notification Service

Note

Esta sección solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso del Amazon Simple Notification Service.

Amazon Simple Notification Service (AmazonSNS) recibe notificaciones de Auto Scaling. Estos eventos se denominan eventos de ciclo de vida y se generan cuando se lanza o termina una instancia en un grupo de escalado automático. Dentro AWS ParallelCluster, el SNS tema de Amazon para el grupo Auto Scaling está suscrito a una SQS cola de Amazon.

Amazon no SNS se usa con AWS Batch clústeres.

Para obtener más información sobre AmazonSNS, consulta <https://aws.amazon.com/sns/y> <https://docs.aws.amazon.com/sns/>.

Amazon Simple Queue Service

Note

Esta sección solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso del Amazon Simple Queue Service.

Amazon Simple Queue Service (AmazonSQS) guarda las notificaciones enviadas desde Auto Scaling, las notificaciones enviadas a través de Amazon SNS y las notificaciones enviadas desde los nodos de cómputo. Amazon SQS desvincula el envío de notificaciones de la recepción de notificaciones. Esto permite que el nodo principal gestione las notificaciones mediante un proceso de sondeo. En este proceso, el nodo principal ejecuta Amazon SQSwatcher y sondea la cola. El escalado automático y los nodos de computación publican mensajes en la cola.

Amazon no SQS se usa con AWS Batch clústeres.

Para obtener más información sobre AmazonSQS, consulta <https://aws.amazon.com/sqs/> y <https://docs.aws.amazon.com/sqs/>.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) AWS ParallelCluster almacena las plantillas ubicadas en cada uno de ellos. Región de AWS AWS ParallelCluster se puede configurar para permitir que CLI/SDKtools utilice Amazon S3.

Cuando utilizas un AWS Batch clúster, se utiliza un bucket de Amazon S3 en tu cuenta para almacenar los datos relacionados. Por ejemplo, el bucket almacena artefactos creados cuando se crea una imagen de Docker y scripts de los trabajos enviados.

Para obtener más información, consulte <https://aws.amazon.com/s3/> y <https://docs.aws.amazon.com/s3/>.

Amazon VPC

Amazon VPC define una red utilizada por los nodos del clúster. La VPC configuración del clúster se define en la [\[vpc\]sección](#).

Para obtener más información sobre AmazonVPC, consulta <https://aws.amazon.com/vpc/> y <https://docs.aws.amazon.com/vpc/>.

Auto Scaling de AWS ParallelCluster

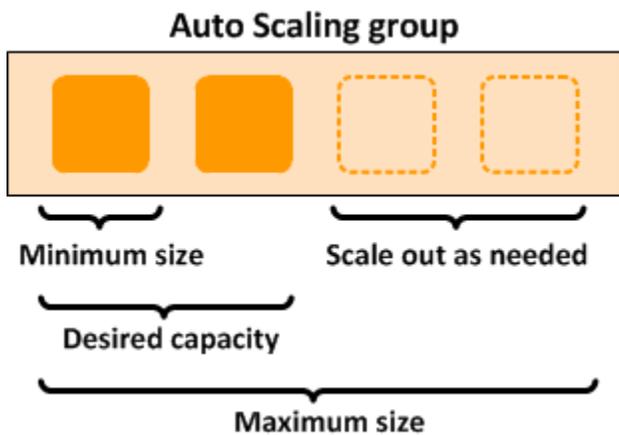
Note

Esta sección solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de planificadores SGE o Torque. Puede seguir utilizándolos en las versiones anteriores a la 2.11.4 inclusive, pero no son aptos para recibir actualizaciones futuras ni asistencia para la solución de problemas por parte de los equipos de servicio AWS y soporte AWS.

A partir de AWS ParallelCluster versión 2.9.0, Auto Scaling no es compatible con Slurm Workload Manager (Slurm). Para obtener información sobre el escalado de colas múltiples Slurm y el escalado de colas múltiples, consulte la [Tutorial sobre el modo de cola múltiple](#).

La estrategia de autoescalado que se describe en este tema se aplica a los clústeres de HPC que se implementan con Son of Grid Engine (SGE) o Torque Resource Manager (Torque). Al implementarse con uno de estos programadores, AWS ParallelCluster aplica las capacidades de escalado administrando el grupo de grupo de escalado automático de los nodos de computación y cambiando a continuación la configuración de programador según sea necesario. Para los clústeres de HPC que se basan en AWS Batch, AWS ParallelCluster se apoya en las capacidades de escalado elástico que proporciona el programador de trabajos administrado por el programador de la tarea AWS. Para obtener más información, consulte [Grupo de Amazon EC2 Auto Scaling](#) en la Guía del usuario de Amazon EC2 Auto Scaling.

Son varias las maneras en que los clústeres que AWS ParallelCluster implementa son elásticos. Al establecer el [initial_queue_size](#) se especifica el valor del tamaño mínimo del grupo de escalado automático de ComputeFleet y también el valor de la capacidad que se quiere tener. Al establecer el [max_queue_size](#) se especifica el valor del tamaño máximo del grupo de escalado automático de ComputeFleet.



Ampliación

Cada minuto, se ejecuta un proceso denominado [jobwatcher](#) en el nodo principal. Evalúa el número actual de instancias que requieren los trabajos pendientes de la cola. Si el número total de nodos ocupados y nodos solicitados es superior al valor deseado actual en el grupo de escalado automático, se agregan más instancias. Si envía más trabajos, la cola se vuelve a evaluar y el grupo de escalado automático se actualiza al valor de [max_queue_size](#) especificado.

Cuando se usa un programador SGE, los trabajos tienen que ejecutar cada uno un número de ranuras (una ranura corresponde a una unidad de procesamiento; por ejemplo, una vCPU). Al evaluar el número de instancias que son necesarias para atender los trabajos pendientes actualmente, `jobwatcher` divide el número total de ranuras solicitadas por la capacidad de un único nodo de computación. La capacidad de un nodo de computación que corresponde al número de vCPU disponibles depende del tipo de instancia que se especifica en la configuración del clúster.

Con los programadores Slurm (anteriormente AWS ParallelCluster versión 2.9.0) y Torque, cada trabajo podría requerir un número de nodos y un número de ranuras por nodo, en función de cada caso. Para cada solicitud, `jobwatcher` determina el número de nodos de computación que son necesarios para cumplir los nuevos requisitos informáticos. Por ejemplo, adoptemos un clúster con `c5.2xlarge` (8 vCPU) como el tipo de instancia de informática, así como tres trabajos pendientes en cola con los siguientes requisitos:

- trabajo1: 2 nodos / 4 ranuras cada uno
- trabajo2: 3 nodos / 2 ranuras
- trabajo3: 1 nodo / 4 ranuras

En este ejemplo, `jobwatcher` requiere que tres nuevas instancias de computación en el grupo de escalado automático se ocupen de los tres trabajos.

Limitación actual: la lógica de escalado automático no considera los nodos ocupados parcialmente cargados. Por ejemplo, un nodo que está ejecutando una tarea se considera ocupado incluso si hay espacios vacíos.

Reducción

El tiempo de inactividad de nodo en cada uno de los nodos de computación se ejecuta y evalúa en un proceso llamado [nodewatcher](#). Una instancia se termina si se cumplen las dos condiciones siguientes:

- Una instancia no tiene ningún trabajo durante un periodo de tiempo superior a [scaledown_idletime](#) (la configuración predeterminada es 10 minutos)
- No hay ningún trabajo pendiente en el clúster

Para terminar una instancia, `nodewatcher` llama a la operación API [TerminateInstanceInAutoScalingGroup](#), la cual elimina una instancia si el tamaño del grupo de escalado automático es al menos el tamaño del grupo de escalado automático mínimo. Este proceso realiza un escalado descendente en el clúster sin que ello afecte a los trabajos en ejecución. También habilita un clúster elástico con un número base fijo de instancias.

Clúster estático

El valor de Auto Scaling es el mismo para HPC que con cualquier otra carga de trabajo. La única diferencia es que AWS ParallelCluster cuenta con un código que hace que interactúe de un modo más inteligente. Por ejemplo, si se requiere un clúster estático, deberá establecer los parámetros [initial_queue_size](#) y [max_queue_size](#) en el tamaño de clúster exacto necesario y, a continuación, establecer el parámetro [maintain_initial_size](#) en `true`. Esto hace que el grupo de escalado automático de ComputeFleet tenga el mismo valor para la capacidad mínima, máxima y deseada.

Tutoriales

En los siguientes tutoriales se le muestra cómo comenzar a utilizar AWS ParallelCluster y se proporciona orientación sobre prácticas recomendadas para algunas tareas comunes.

Temas

- [Ejecución del primer trabajo en AWS ParallelCluster](#)
- [Creación de una AMI de AWS ParallelCluster personalizada](#)
- [Ejecución de un trabajo de MPI con AWS ParallelCluster y el programador awsbatch](#)
- [Cifrado de disco con una clave KMS personalizada](#)
- [Tutorial sobre el modo de cola múltiple](#)

Ejecución del primer trabajo en AWS ParallelCluster

Este tutorial le guía por la ejecución de su primer trabajo Hello World en AWS ParallelCluster.

Requisitos previos

- AWS ParallelCluster [está instalado](#).
- AWS CLI está [instalado y configurado](#).
- Tiene un [EC2 key pair](#).
- Tiene un rol de IAM con los [permisos](#) necesarios para ejecutar el CLI [pcluster](#).

Comprobación de la instalación

En primer lugar, comprobamos que AWS ParallelCluster esté instalado y configurado correctamente.

```
$ pcluster version
```

Esto devuelve la versión en ejecución de AWS ParallelCluster. Si recibe un mensaje del resultado sobre la configuración, tiene que ejecutar lo siguiente para configurar AWS ParallelCluster:

```
$ pcluster configure
```

Creación de su primer clúster

Ahora ha llegado el momento de crear su primer clúster. Debido a que la carga de trabajo de este tutorial no es excesiva, podemos usar el tamaño de instancia predeterminado de `t2.micro`. (Para las cargas de trabajo de producción, puede elegir el tamaño de instancia que mejor se adapte a sus necesidades).

Llamemos a su clúster `hello-world`.

```
$ pcluster create hello-world
```

Al crearse el clúster, verá un resultado parecido al siguiente:

```
Starting: hello-world
Status: parallelcluster-hello-world - CREATE_COMPLETE
MasterPublicIP = 54.148.x.x
ClusterUser: ec2-user
MasterPrivateIP = 192.168.x.x
GangliaPrivateURL = http://192.168.x.x/ganglia/
GangliaPublicURL = http://54.148.x.x/ganglia/
```

El mensaje `CREATE_COMPLETE` muestra que el clúster se ha creado correctamente. El resultado también nos proporciona las direcciones IP públicas y privadas de nuestro nodo principal. Necesitamos esta IP para iniciar sesión.

Iniciar sesión en su nodo principal

Use su archivo OpenSSH para iniciar sesión en el nodo principal.

```
pcluster ssh hello-world -i /path/to/keyfile.pem
```

Después de iniciar sesión, ejecute el comando `qhost` para comprobar que sus nodos de computación se instalan y configuran.

```
$ qhost
HOSTNAME                ARCH      NCPU NSOC  NCOR  NTHR  LOAD  MEMTOT  MEMUSE  SWAPT0
SWAPUS
-----
global                  -         -    -    -    -    -    -    -    -
-
```

| | | | | | | | | | |
|-------------------------|----------|---|---|---|---|------|------|--------|---------|
| ip-192-168-1-125 0.0 | 1x-amd64 | 2 | 1 | 2 | 2 | 0.15 | 3.7G | 130.8M | 1024.0M |
| ip-192-168-1-126 0.0 | 1x-amd64 | 2 | 1 | 2 | 2 | 0.15 | 3.7G | 130.8M | 1024.0M |

El resultado muestra que tenemos dos nodos de computación en nuestro clúster, ambos con subprocesos disponibles.

Ejecución de su primer trabajo con SGE

Note

Este ejemplo solo se aplica a AWS ParallelCluster las versiones anteriores a la 2.11.4 (inclusive). A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de planificadores SGE o Torque.

A continuación, creamos un trabajo que entra en suspensión durante un tiempo y luego genera su propio nombre de host.

Cree un archivo denominado `hellojob.sh` con el siguiente contenido.

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

A continuación, envíe el trabajo mediante `qsub` y compruebe que se ejecuta.

```
$ qsub hellojob.sh
Your job 1 ("hellojob.sh") has been submitted
```

Ahora puede ver la cola y comprobar el estado del trabajo.

```
$ qstat
job-ID prior name user state submit/start at queue
slots ja-task-ID
-----
1 0.55500 hellojob.s ec2-user r 03/24/2015 22:23:48
all.q@ip-192-168-1-125.us-west 1
```

El resultado muestra que el trabajo se encuentra actualmente en estado de ejecución. Espere 30 segundos a que el trabajo se termine y, a continuación, vuelva a ejecutar `qstat`.

```
$ qstat
$
```

Ahora que no hay trabajos en la cola, podemos comprobar el resultado en nuestro directorio actual.

```
$ ls -l
total 8
-rw-rw-r-- 1 ec2-user ec2-user 48 Mar 24 22:34 hellojob.sh
-rw-r--r-- 1 ec2-user ec2-user  0 Mar 24 22:34 hellojob.sh.e1
-rw-r--r-- 1 ec2-user ec2-user 34 Mar 24 22:34 hellojob.sh.o1
```

En el resultado, vemos un archivo "e1" y "o1" en nuestro script de trabajo. Dado que el archivo e1 está vacío, stderr no ha generado respuesta. Si vemos el archivo o1, podemos ver el resultado que nuestro trabajo ha generado.

```
$ cat hellojob.sh.o1
Hello World from ip-192-168-1-125
```

El resultado también muestra que nuestro trabajo se ha ejecutado correctamente en la instancia `ip-192-168-1-125`.

Para obtener más información acerca de la creación y el uso de clústeres, consulte [Prácticas recomendadas](#).

Creación de una AMI de AWS ParallelCluster personalizada

Important

No es recomendable crear una AMI personalizada para personalizar AWS ParallelCluster. Esto se debe a que después de crear su propia AMI, dejará de recibir actualizaciones o correcciones de errores con las versiones futuras de AWS ParallelCluster. Además, si crea una AMI personalizada, debe repetir los pasos que utilizó para crear la AMI personalizada con cada nueva versión de AWS ParallelCluster.

Antes de seguir leyendo, recomendamos que consulte la sección [Acciones de arranque personalizadas](#) para determinar si las modificaciones que desea realizar se pueden incluir en un script y son compatibles con futuras versiones de AWS ParallelCluster.

Aunque crear una AMI personalizada no es lo ideal (por las razones mencionadas anteriormente), todavía hay situaciones en las que es necesario crear una AMI para AWS ParallelCluster personalizada. Este tutorial le guía a través del proceso de creación de una AMI personalizada para estos escenarios.

Note

A partir de AWS ParallelCluster versión 2.6.1, la mayoría de las recetas de instalación se omiten de forma predeterminada al lanzar nodos. Esto sirve para mejorar los tiempos de inicio. Para ejecutar todas las recetas de instalación para una mejor compatibilidad con versiones anteriores a costa de los tiempos de inicio, añada "skip_install_recipes" : "no" a la clave `cluster` en la configuración [extra_json](#). Por ejemplo:

```
extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
```

Requisitos previos

- AWS ParallelCluster [está instalado](#).
- AWS CLI está [instalado y configurado](#).
- Tiene un [EC2 key pair](#).
- Tiene un rol de IAM con los [permisos](#) necesarios para ejecutar el CLI [pcluster](#).

Cómo personalizar la AMI de AWS ParallelCluster

Hay tres formas de utilizar una AMI personalizada para AWS ParallelCluster que se describen en las siguientes secciones. Dos de estos tres métodos requieren que cree una nueva AMI que esté disponible en su Cuenta de AWS. El tercer método (usar una AMI personalizada en tiempo de ejecución) no requiere que se cree nada por adelantado, pero sí añade riesgos a la implementación. Elija el método que mejor se adapte a sus necesidades.

Modificar una AMI de

Este es el método más seguro y recomendado. Como la AMI AWS ParallelCluster básica se actualiza a menudo con nuevas versiones, esta AMI tiene todos los componentes necesarios para que AWS ParallelCluster funcione cuando se instala y configura. Puede comenzar con esto como base.

New EC2 console

1. En la lista de AMI AWS ParallelCluster, busque la AMI que corresponde a la región Región de AWS específica que usa. La lista de AMI que elija debe coincidir con la versión de AWS ParallelCluster que utilice. Ejecute `pccluster version` para verificar la versión. Para AWS ParallelCluster versión 2.11.9, visite <https://github.com/aws/aws-parallelcluster/blob/v2.11.9/amis.txt>. Para seleccionar otra versión, utilice el mismo enlace, pulse el botón Etiqueta: 2.11.9, seleccione la pestaña Etiquetas y a continuación, seleccione la versión adecuada.
2. Inicie sesión en la AWS Management Console y abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
3. En el panel de Amazon EC2, elija Lanzar instancia.
4. En Imágenes de aplicaciones y sistemas operativos, elija Buscar más AMI, vaya a las AMI de la comunidad e introduzca su ID de AWS ParallelCluster AMI Región de AWS en el cuadro de búsqueda.
5. Seleccione la AMI, elija el tipo de instancia y las propiedades, seleccione su Par de claves y Lanza instancia.
6. Inicie sesión en la instancia mediante el usuario del sistema operativo y su clave de SSH. Para obtener más información, vaya a Instancias, seleccione la nueva instancia y Conectar.
7. Personalice su instancia según sea necesario.
8. Ejecute el siguiente comando para preparar su instancia para la creación de la AMI:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Navegue a Instancias, elija la nueva instancia, seleccione Estado de la instancia, y Detener instancia
10. Cree una AMI nueva a partir de la instancia mediante la consola EC2 o AWS CLI [create-image](#).

Desde la consola EC2

- a. Elija Instances (Instancia[s]) en el panel de navegación.

- b. Elija la instancia que creó y modificó.
- c. Elija Acciones, Imágenes y plantillas y a continuación Crear imagen.
- d. Elija Create Image (Crear imagen).

11Escriba el ID de AMI en el campo [custom_ami](#) de la configuración del clúster.

Old EC2 console

1. En la lista de AMI AWS ParallelCluster, busque la AMI que corresponde a la región Región de AWS específica que usa. La lista de AMI que elija debe coincidir con la versión de AWS ParallelCluster que utilice. Ejecute `pccluster version` para verificar la versión. Para AWS ParallelCluster versión 2.11.9, visite <https://github.com/aws/aws-parallelcluster/blob/v2.11.9/amis.txt>. Para seleccionar otra versión, utilice el mismo enlace, pulse el botón Etiqueta: 2.11.9, seleccione la pestaña Etiquetas y a continuación, seleccione la versión adecuada.
2. Inicie sesión en la AWS Management Console y abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
3. En el panel de Amazon EC2, elija Lanzar instancia.
4. Elija las AMI de la comunidad, busque la ID de la AMI AWS ParallelCluster y selecciónela.
5. Elija el tipo de instancia y seleccione Siguiente: configurar los detalles de la instancia o Revisar y lanzar para lanzar la instancia.
6. Elija Lanzar, seleccione su par de claves y lance las instancias.
7. Inicie sesión en la instancia mediante el usuario del sistema operativo y su clave de SSH. Para obtener más información, vaya a Instancias, seleccione la nueva instancia y Conectar.
8. Personalice su instancia según sea necesario.
9. Ejecute el siguiente comando para preparar su instancia para la creación de la AMI:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

10Navegue a Instancias, elija la nueva instancia, seleccione Estado de la instancia, y Detener

11Cree una AMI nueva a partir de la instancia mediante la consola EC2 o AWS CLI [create-image](#).

Desde la consola EC2

- a. Elija Instances (Instancia[s]) en el panel de navegación.
- b. Elija la instancia que creó y modificó.
- c. Elija Acciones, Imagen y a continuación, Crear imagen.

d. Elija Create Image (Crear imagen).

12Escriba el ID de AMI en el campo [custom_ami](#) de la configuración del clúster.

Creación de una AMI de AWS ParallelCluster personalizada

Si ya tiene instalados una AMI personalizada y software, puede aplicar en ellos los cambios que AWS ParallelCluster necesita.

1. Instale lo siguiente en su sistema local, junto con la CLI de AWS ParallelCluster:

- Packer: obtenga la versión más reciente para su sistema operativo en el [sitio web de Packer](#) e instálela. La versión debe ser al menos 1.4.0, pero se recomienda utilizar la versión más reciente. Compruebe que el `packer` comando esté disponible en su RUTA.

Note

Antes de AWS ParallelCluster versión 2.8.0, era necesario usar `pcluster createami` para [Berkshelf](#) (que se instala mediante el uso de `gem install berkshelf`).

2. Configure sus credenciales de modo que Packer pueda realizar llamadas a operaciones de la API en su nombre. El conjunto mínimo de permisos necesarios para que Packer funcione se documenta en la sección [Tarea de IAM o rol de instancia](#) del tema Amazon AMI Builder de la documentación de Packer.
3. Utilice el comando `createami` de la CLI de AWS ParallelCluster para crear una AMI de AWS ParallelCluster a partir de la AMI que ha proporcionado como base:

```
pcluster createami --ami-id <BASE_AMI> --os <BASE_AMI_OS>
```

Important

No debe usar una AMI AWS ParallelCluster de un clúster en ejecución como `<BASE_AMI>` para el comando `createami`. De lo contrario, el comando falla.

Para otros parámetros, consulte [pcluster createami](#).

4. El comando del paso 4 ejecuta Packer, el cual a su vez ejecuta siguiente específicamente:
- a. Lanza una instancia mediante la AMI Base que se le ha proporcionado.

- b. Aplica el libro de recetas de AWS ParallelCluster a la instancia para instalar software y realizar otras tareas de configuración necesarias.
 - c. Detiene la instancia.
 - d. Crea una nueva AMI a partir de la instancia.
 - e. Termina la instancia después de crear la AMI.
 - f. Produce la cadena del nuevo ID de AMI que debe utilizarse para crear el clúster.
5. Para crear el clúster, especifique el ID de AMI en el campo [custom_ami](#) de la configuración del clúster.

Note

El tipo de instancia que se utiliza para crear una AMI de AWS ParallelCluster personalizada es `t2.xlarge`. Este tipo de instancia no cumple los requisitos para la capa AWS gratuita, por lo que se le cobrará por cualquier instancia que se cree al crear esta AMI.

Uso de una AMI personalizada en tiempo de ejecución

Warning

Para evitar el riesgo de usar una AMI que no sea compatible con AWS ParallelCluster, le recomendamos que evite usar este método.

Cuando los nodos de procesamiento se lanzan con AMI potencialmente no probadas en tiempo de ejecución, las incompatibilidades con la instalación en tiempo de ejecución AWS ParallelCluster del software necesario pueden provocar que deje de AWS ParallelCluster funcionar.

Si no quiere crear nada por adelantado, puede usar su AMI y crear un AWS ParallelCluster a partir de esa AMI.

Con este método, se tarda más tiempo en crear AWS ParallelCluster, ya que debe estar instalado todo el software que AWS ParallelCluster se necesita para crear el clúster. Además, la ampliación también lleva más tiempo.

- Escriba el ID de AMI en el campo [custom_ami](#) de la configuración del clúster.

Ejecución de un trabajo de MPI con AWS ParallelCluster y el programador **awsbatch**

Este tutorial lo guía por la ejecución de un trabajo de MPI donde se utilice `awsbatch` como programador.

Requisitos previos

- AWS ParallelCluster [está instalado](#).
- AWS CLI está [instalado y configurado](#).
- Tiene un [EC2 key pair](#).
- Tiene un rol de IAM con los [permisos](#) necesarios para ejecutar el CLI [pcluster](#).

Creación del clúster

En primer lugar, crearemos una configuración para un clúster que use `awsbatch` como programador. Acuérdesse de insertar los datos que faltan en la sección `vpc` y el campo `key_name` con los recursos que ha creado en el momento de realizar la configuración.

```
[global]
sanity_check = true

[aws]
aws_region_name = us-east-1

[cluster awsbatch]
base_os = alinux
# Replace with the name of the key you intend to use.
key_name = key-#####
vpc_settings = my-vpc
scheduler = awsbatch
compute_instance_type = optimal
min_vcpus = 2
desired_vcpus = 2
max_vcpus = 24

[vpc my-vpc]
# Replace with the id of the vpc you intend to use.
vpc_id = vpc-#####
```

```
# Replace with id of the subnet for the Head node.
master_subnet_id = subnet-#####
# Replace with id of the subnet for the Compute nodes.
# A NAT Gateway is required for MNP.
compute_subnet_id = subnet-#####
```

Ahora ya puede comenzar a crear el clúster. Llamemos a nuestro clúster *awsbatch-tutorial*.

```
$ pcluster create -c /path/to/the/created/config/aws_batch.config -t awsbatch awsbatch-tutorial
```

Al crearse el clúster, verá un resultado parecido al siguiente:

```
Beginning cluster creation for cluster: awsbatch-tutorial
Creating stack named: parallelcluster-awsbatch
Status: parallelcluster-awsbatch - CREATE_COMPLETE
MasterPublicIP: 54.160.xxx.xxx
ClusterUser: ec2-user
MasterPrivateIP: 10.0.0.15
```

Iniciar sesión en su nodo principal

Todos los comandos de la [CLI por lotes de AWS ParallelCluster](#) están disponibles en el equipo cliente donde está instalado AWS ParallelCluster. Sin embargo, vamos a SSH en el nodo principal y envíe los trabajos desde allí. Esto nos permite aprovechar el volumen de NFS, que se comparte entre la instancia principal y todas las instancias de Docker que ejecutan trabajos de AWS Batch.

Use su archivo pem SSH para iniciar sesión en el nodo principal.

```
$ pcluster ssh awsbatch-tutorial -i /path/to/keyfile.pem
```

Cuando inicie sesión, ejecute los comandos `awsbqueues` y `awsbhosts` para mostrar la cola de configurada y las instancias Amazon ECS en ejecución.

```
[ec2-user@ip-10-0-0-111 ~]$ awsbqueues
jobQueueName          status
-----
parallelcluster-awsbatch-tutorial  VALID

[ec2-user@ip-10-0-0-111 ~]$ awsbhosts
```

```

ec2InstanceId      instanceType      privateIpAddress  publicIpAddress
  runningJobs
-----
-----
i-0d6a0c8c560cd5bed  m4.large        10.0.0.235       34.239.174.236
0

```

Tal como puede apreciar en el resultado, tenemos un único host ejecutándose. Esto se debe al valor que hemos elegido para [min_vcpus](#) en la configuración. Si desea mostrar detalles adicionales sobre la cola de AWS Batch y los hosts, añada la marca `-d` al comando.

Ejecución de su primer trabajo con AWS Batch

Antes de migrar a MPI, vamos a crear un trabajo ficticio que está en suspensión durante un cierto tiempo y luego genera su propio nombre de host y acoge el nombre que se le ha transferido como parámetro.

Cree un archivo denominado "hellojob.sh" con el siguiente contenido.

```

#!/bin/bash

sleep 30
echo "Hello $1 from $HOSTNAME"
echo "Hello $1 from $HOSTNAME" > "/shared/secret_message_for_${1}_by_
${AWS_BATCH_JOB_ID}"

```

A continuación, envíe el trabajo mediante `awsbsub` y compruebe que se ejecuta.

```

$ awsbsub -jn hello -cf hellojob.sh Luca
Job 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2 (hello) has been submitted.

```

Vea la cola y compruebe el estado del trabajo.

```

$ awsbstat
jobId              jobName      status      startedAt
stoppedAt         exitCode
-----
-----
6efe6c7c-4943-4c1a-baf5-edbfeccab5d2  hello       RUNNING    2018-11-12 09:41:29 -
-

```

El resultado proporciona información detallada del trabajo.

```
$ awsbatch 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
jobId           : 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
jobName        : hello
createdAt      : 2018-11-12 09:41:21
startedAt      : 2018-11-12 09:41:29
stoppedAt      : -
status         : RUNNING
statusReason   : -
jobDefinition  : parallelcluster-myBatch:1
jobQueue       : parallelcluster-myBatch
command        : /bin/bash -c 'aws s3 --region us-east-1 cp s3://
parallelcluster-mybatch-lui1ftboklhps95/batch/job-hellojob_sh-1542015680924.sh /
tmp/batch/job-hellojob_sh-1542015680924.sh; bash /tmp/batch/job-
hellojob_sh-1542015680924.sh Luca'
exitCode       : -
reason         : -
vcpus         : 1
memory[MB]    : 128
nodes         : 1
logStream      : parallelcluster-myBatch/default/c75dac4a-5aca-4238-
a4dd-078037453554
log            : https://console.aws.amazon.com/cloudwatch/home?region=us-
east-1#logEventViewer:group=/aws/batch/job;stream=parallelcluster-myBatch/default/
c75dac4a-5aca-4238-a4dd-078037453554
-----
```

Tenga en cuenta que el trabajo se encuentra actualmente en estado de ejecución (RUNNING). Espere 30 segundos a que el trabajo se termine y, a continuación, vuelva a ejecutar `awsbatch`.

```
$ awsbatch
jobId           jobName      status      startedAt
stoppedAt      exitCode
-----
-----
```

Ahora verá que el trabajo se encuentra en estado realizado correctamente (SUCCEEDED).

```
$ awsbatch -s SUCCEEDED
jobId           jobName      status      startedAt
stoppedAt      exitCode
```

```

-----
-----
6efe6c7c-4943-4c1a-baf5-edbfeccab5d2 hello SUCCEEDED 2018-11-12 09:41:29
2018-11-12 09:42:00 0

```

Dado que ahora no quedan trabajos en la cola, podemos comprobar la salida que se obtiene con el comando `awsbout`.

```

$ awsbout 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
2018-11-12 09:41:29: Starting Job 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
download: s3://parallelcluster-mybatch-lui1ftboklhpns95/batch/job-
hellojob_sh-1542015680924.sh to tmp/batch/job-hellojob_sh-1542015680924.sh
2018-11-12 09:42:00: Hello Luca from ip-172-31-4-234

```

Podemos ver que nuestro trabajo se ha ejecutado correctamente en la instancia "ip-172-31-4-234".

Si entra en el directorio `/shared`, encontrará un mensaje secreto para usted.

Para explorar todas las características disponibles que no forman parte de este tutorial, consulte la [documentación de la CLI por lotes de AWS ParallelCluster](#). Cuando esté listo para continuar con el tutorial avanzaremos y veremos cómo enviar un trabajo de MPI.

Ejecución de un trabajo de MPI en un entorno en paralelo de varios nodos

Aunque siga conectado al nodo principal, cree un archivo en el directorio `/shared` llamado `mpi_hello_world.c`. Añada el siguiente programa de MPI al archivo:

```

// Copyright 2011 www.mpitutorial.com
//
// An intro MPI hello world program that uses MPI_Init, MPI_Comm_size,
// MPI_Comm_rank, MPI_Finalize, and MPI_Get_processor_name.
//
#include <mpi.h>
#include <stdio.h>
#include <stddef.h>

int main(int argc, char** argv) {
    // Initialize the MPI environment. The two arguments to MPI Init are not
    // currently used by MPI implementations, but are there in case future
    // implementations might need the arguments.
    MPI_Init(NULL, NULL);

```

```
// Get the number of processes
int world_size;
MPI_Comm_size(MPI_COMM_WORLD, &world_size);

// Get the rank of the process
int world_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

// Get the name of the processor
char processor_name[MPI_MAX_PROCESSOR_NAME];
int name_len;
MPI_Get_processor_name(processor_name, &name_len);

// Print off a hello world message
printf("Hello world from processor %s, rank %d out of %d processors\n",
       processor_name, world_rank, world_size);

// Finalize the MPI environment. No more MPI calls can be made after this
MPI_Finalize();
}
```

Ahora guarde el siguiente código como `submit_mpi.sh`:

```
#!/bin/bash
echo "ip container: $(/sbin/ip -o -4 addr list eth0 | awk '{print $4}' | cut -d/ -f1)"
echo "ip host: $(curl -s "http://169.254.169.254/latest/meta-data/local-ipv4")"

# get shared dir
IFS=',' _shared_dirs=${PCLUSTER_SHARED_DIRS}
_shared_dir=${_shared_dirs[0]}
_job_dir="${_shared_dir}/${AWS_BATCH_JOB_ID%#*}-${AWS_BATCH_JOB_ATTEMPT}"
_exit_code_file="${_job_dir}/batch-exit-code"

if [[ "${AWS_BATCH_JOB_NODE_INDEX}" -eq "${AWS_BATCH_JOB_MAIN_NODE_INDEX}" ]]; then
    echo "Hello I'm the main node $HOSTNAME! I run the mpi job!"

    mkdir -p "${_job_dir}"

    echo "Compiling..."
    /usr/lib64/openmpi/bin/mpicc -o "${_job_dir}/mpi_hello_world" "${_shared_dir}/
mpi_hello_world.c"

    echo "Running..."
```

```
/usr/lib64/openmpi/bin/mpirun --mca btl_tcp_if_include eth0 --allow-run-as-root --
machinefile "${HOME}/hostfile" "${_job_dir}/mpi_hello_world"

# Write exit status code
echo "0" > "${_exit_code_file}"
# Waiting for compute nodes to terminate
sleep 30
else
    echo "Hello I'm the compute node $HOSTNAME! I let the main node orchestrate the mpi
processing!"
    # Since mpi orchestration happens on the main node, we need to make sure the
containers representing the compute
    # nodes are not terminated. A simple trick is to wait for a file containing the
status code to be created.
    # All compute nodes are terminated by AWS Batch if the main node exits abruptly.
    while [ ! -f "${_exit_code_file}" ]; do
        sleep 2
    done
    exit $(cat "${_exit_code_file}")
fi
```

Ahora ya estamos listos para enviar nuestro primer trabajo de MPI y hacer que se ejecute de forma simultánea en tres nodos:

```
$ awsbsub -n 3 -cf submit_mpi.sh
```

Ahora vamos a monitorizar el estado del trabajo y esperar a que entre en el estado RUNNING:

```
$ watch awsbstat -d
```

Cuando el trabajo entre en el estado RUNNING, podremos examinar su salida. Para mostrar la salida del nodo principal, añade #0 al ID de trabajo. Para mostrar la salida de los nodos de computación, utilice #1 y #2:

```
[ec2-user@ip-10-0-0-111 ~]$ awsbout -s 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#0
2018-11-27 15:50:10: Job id: 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#0
2018-11-27 15:50:10: Initializing the environment...
2018-11-27 15:50:10: Starting ssh agents...
2018-11-27 15:50:11: Agent pid 7
2018-11-27 15:50:11: Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
2018-11-27 15:50:11: Mounting shared file system...
2018-11-27 15:50:11: Generating hostfile...
```

```
2018-11-27 15:50:11: Detected 1/3 compute nodes. Waiting for all compute nodes to
start.
2018-11-27 15:50:26: Detected 1/3 compute nodes. Waiting for all compute nodes to
start.
2018-11-27 15:50:41: Detected 1/3 compute nodes. Waiting for all compute nodes to
start.
2018-11-27 15:50:56: Detected 3/3 compute nodes. Waiting for all compute nodes to
start.
2018-11-27 15:51:11: Starting the job...
download: s3://parallelcluster-awsbatch-tutorial-iwyl4458saiwgwvg/batch/job-
submit_mpi_sh-1543333713772.sh to tmp/batch/job-submit_mpi_sh-1543333713772.sh
2018-11-27 15:51:12: ip container: 10.0.0.180
2018-11-27 15:51:12: ip host: 10.0.0.245
2018-11-27 15:51:12: Compiling...
2018-11-27 15:51:12: Running...
2018-11-27 15:51:12: Hello I'm the main node! I run the mpi job!
2018-11-27 15:51:12: Warning: Permanently added '10.0.0.199' (RSA) to the list of known
hosts.
2018-11-27 15:51:12: Warning: Permanently added '10.0.0.147' (RSA) to the list of known
hosts.
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-180.ec2.internal, rank 1 out
of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-199.ec2.internal, rank 5 out
of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-180.ec2.internal, rank 0 out
of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-199.ec2.internal, rank 4 out
of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-147.ec2.internal, rank 2 out
of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-147.ec2.internal, rank 3 out
of 6 processors

[ec2-user@ip-10-0-0-111 ~]$ awsbatch -s 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#1
2018-11-27 15:50:52: Job id: 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#1
2018-11-27 15:50:52: Initializing the environment...
2018-11-27 15:50:52: Starting ssh agents...
2018-11-27 15:50:52: Agent pid 7
2018-11-27 15:50:52: Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
2018-11-27 15:50:52: Mounting shared file system...
2018-11-27 15:50:52: Generating hostfile...
2018-11-27 15:50:52: Starting the job...
download: s3://parallelcluster-awsbatch-tutorial-iwyl4458saiwgwvg/batch/job-
submit_mpi_sh-1543333713772.sh to tmp/batch/job-submit_mpi_sh-1543333713772.sh
```

```

2018-11-27 15:50:53: ip container: 10.0.0.199
2018-11-27 15:50:53: ip host: 10.0.0.227
2018-11-27 15:50:53: Compiling...
2018-11-27 15:50:53: Running...
2018-11-27 15:50:53: Hello I'm a compute node! I let the main node orchestrate the mpi
  execution!

```

Ahora podemos confirmar que el trabajo se ha completado correctamente:

```

[ec2-user@ip-10-0-0-111 ~]$ awsbststat -s ALL
jobId                jobName              status              startedAt
stoppedAt            exitCode
-----
-----
5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d  submit_mpi_sh      SUCCEEDED          2018-11-27 15:50:10
2018-11-27 15:51:26  -

```

Nota: Si desea terminar un trabajo antes de que este finalice por sí mismo, puede ejecutar el comando `awsbkill`.

Cifrado de disco con una clave KMS personalizada

AWS ParallelCluster admite las opciones de configuración `ebs_kms_key_id` y `fsx_kms_key_id`. Estas opciones le permiten proporcionar una clave AWS KMS personalizada para el cifrado de disco de Amazon EBS Disk o FSx para Lustre. Para utilizarlas, debe especificar un `ec2_iam_role`.

Para que el clúster pueda crear, la clave AWS KMS necesita saber el nombre del rol del clúster. Esto le impide utilizar el rol que se ha creado en la creación del clúster y requerir un `ec2_iam_role` personalizado.

Requisitos previos

- AWS ParallelCluster [está instalado](#).
- AWS CLI está [instalado y configurado](#).
- Tiene un [EC2 key pair](#).
- Tiene un rol de IAM con los [permisos](#) necesarios para ejecutar el CLI [pcluster](#).

Creación de la función

Primero crea una política:

1. Vaya a la consola de IAM: <https://console.aws.amazon.com/iam/home>.
2. En Políticas (Políticas), Create policy (Crear política), haga clic en la pestaña JSON.
3. Como cuerpo de la política, pegue la [Política de instancias](#). Asegúrese de reemplazar todas las coincidencias de `<AWS ACCOUNT ID>` y `<REGION>`.
4. Asigne a la política el nombre `ParallelClusterInstancePolicy` y, a continuación, haga clic en Create Policy (Crear política).

A continuación, cree un rol:

1. En Roles, cree un rol.
2. Haga clic en EC2 como la entidad de confianza.
3. En Permissions (Permisos), busque el rol `ParallelClusterInstancePolicy` que acaba de crear y asícielo.
4. Asigne al rol el nombre `ParallelClusterInstanceRole` y, a continuación, haga clic en Create Role (Crear rol).

Asigne permisos a la clave

En la AWS KMS consola > Claves administradas por el cliente > haga clic en el alias o el ID de clave de su clave.

Haga clic en el botón Añadir del cuadro Usuarios clave, situado debajo de la pestaña Política clave, y busque el `ParallelClusterInstanceRole` que acaba de crear. Asícielo.

Creación del clúster

Ahora cree un clúster. A continuación se muestra un ejemplo de un clúster con unidades Raid 0 cifradas:

```
[cluster default]
...
raid_settings = rs
```

```
ec2_iam_role = ParallelClusterInstanceRole

[raid rs]
shared_dir = raid
raid_type = 0
num_of_raid_volumes = 2
volume_size = 100
encrypted = true
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

A continuación se muestra un ejemplo con el sistema de archivos FSx para Lustre:

```
[cluster default]
...
fsx_settings = fs
ec2_iam_role = ParallelClusterInstanceRole

[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
import_path = s3://bucket
weekly_maintenance_start_time = 1:00:00
fsx_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Se aplican configuraciones similares a los sistemas de archivos basados en Amazon EBS y Amazon FSx.

Tutorial sobre el modo de cola múltiple

Ejecutar sus trabajos en AWS ParallelCluster con el modo de cola múltiple

Este tutorial le guía por la ejecución de su primer trabajo Hello World en AWS ParallelCluster con [Modo de Cola múltiple](#).

Requisitos previos

- AWS ParallelCluster [está instalado](#).
- AWS CLI está [instalado y configurado](#).

- Tiene un [EC2 key pair](#).
- Tiene un rol de IAM con los [permisos](#) necesarios para ejecutar el CLI [pcluster](#).

Note

El modo de cola múltiple solo se admite en AWS ParallelCluster versión 2.9.0 o posterior.

Configuración de su clúster

Para comprobar que AWS ParallelCluster se ha instalado correctamente, ejecute el siguiente comando:

```
$ pcluster version
```

Para obtener más información acerca de `pcluster version`, consulte [pcluster version](#).

Este comando devuelve la versión en ejecución de AWS ParallelCluster.

A continuación, ejecute `pcluster configure` para generar un archivo de configuración básico. Siga todas las instrucciones que aparecen después de este comando.

```
$ pcluster configure
```

Para obtener más información acerca del comando `pcluster configure`, consulte [pcluster configure](#).

Cuando realice este paso, tendrá un archivo de configuración básica en el `~/.parallelcluster/config`. Este archivo debe contener una sección de configuraciones de clúster básicas y una sección de VPC.

La siguiente parte del tutorial describe cómo modificar la configuración recién creada y lanzar un clúster con varias colas.

Note

Algunas instancias utilizadas en este tutorial no son aptas para el nivel gratuito.

Para este tutorial, use la siguiente configuración.

```
[global]
update_check = true
sanity_check = true
cluster_template = multi-queue

[aws]
aws_region_name = <Your Región de AWS>

[scaling demo]
scaledown_idletime = 5                # optional, defaults to 10 minutes

[cluster multi-queue-special]
key_name = < Your key name >
base_os = alinux2                    # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge     # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo              # optional, defaults to no custom scaling settings
queue_settings = efa,gpu

[cluster multi-queue]
key_name = <Your SSH key name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge     # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = spot,ondemand

[queue spot]
compute_resource_settings = spot_i1,spot_i2
compute_type = spot                 # optional, defaults to ondemand

[compute_resource spot_i1]
instance_type = c5.xlarge
min_count = 0                       # optional, defaults to 0
max_count = 10                      # optional, defaults to 10

[compute_resource spot_i2]
instance_type = t2.micro
min_count = 1
initial_count = 2
```

```
[queue ondemand]
compute_resource_settings = ondemand_i1
disable_hyperthreading = true          # optional, defaults to false

[compute_resource ondemand_i1]
instance_type = c5.2xlarge
```

Creación de su clúster

En esta sección se detalla cómo crear el clúster en modo de cola múltiple.

En primer lugar, asigne un nombre al clúster `multi-queue-hello-world` y créelo de acuerdo con la sección de `multi-queue` clúster definida en la sección anterior.

```
$ pcluster create multi-queue-hello-world -t multi-queue
```

Para obtener más información acerca de `pcluster create`, consulte [pcluster create](#).

Cuando se crea el clúster, se muestra el siguiente resultado:

```
Beginning cluster creation for cluster: multi-queue-hello-world
Creating stack named: parallelcluster-multi-queue-hello-world
Status: parallelcluster-multi-queue-hello-world - CREATE_COMPLETE
MasterPublicIP: 3.130.xxx.xx
ClusterUser: ec2-user
MasterPrivateIP: 172.31.xx.xx
```

El mensaje `CREATE_COMPLETE` muestra que el clúster se ha creado correctamente. El resultado también nos proporciona las direcciones IP públicas y privadas del nodo principal.

Iniciar sesión en su nodo principal

Usa tu archivo de clave SSH privado para iniciar sesión en tu nodo principal.

```
$ pcluster ssh multi-queue-hello-world -i ~/path/to/keyfile.pem
```

Para obtener más información acerca de `pcluster ssh`, consulte [pcluster ssh](#).

Después de iniciar sesión, ejecute el comando `sinfo` para comprobar que sus colas de programador se instalan y configuran.

Para obtener más información acerca de `sinfo`, consulte [sinfo](#) en la Slurmdocumentación.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand   up    infinite   10   idle~ ondemand-dy-c52xlarge-[1-10]
spot*      up    infinite   18   idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2micro-[2-9]
spot*      up    infinite    2   idle spot-dy-t2micro-1,spot-st-t2micro-1
```

El resultado muestra que tiene dos nodos de `t2.micro` cómputo en ese `idle` estado que están disponibles en su clúster.

Note

- `spot-st-t2micro-1` es un nodo estático que `st` lleva su nombre. Este nodo está siempre disponible y corresponde a `min_count = 1` la configuración de su clúster.
- `spot-dy-t2micro-1` es un nodo dinámico que `dy` lleva su nombre. Este nodo está disponible actualmente porque corresponde a `initial_count - min_count = 1` la configuración de su clúster. Este nodo se reduce después `scaledown_idletime` de los cinco minutos habituales.

Todos los demás nodos están en estado de ahorro de energía, lo que se indica con el `~` sufijo en el estado de nodo, sin que ninguna instancia de EC2 los respalde. La cola predeterminada se designa con un `*` sufijo después del nombre de la cola, al igual que `spot` la cola de trabajos predeterminada.

Ejecutar el trabajo en modo de cola múltiple

A continuación, intente ejecutar un trabajo para dormir un rato. Más adelante, el trabajo generará su propio nombre de servidor. Asegúrese de que el usuario actual pueda ejecutar este script.

```
$ cat hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

Envíe el trabajo mediante el comando `sbatch`. Solicite dos nodos para este trabajo con la `-N 2` opción y compruebe que el trabajo se envía correctamente. Para obtener más información acerca de `sbatch`, consulte [sbatch](#) en la documentación de Slurm.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 2
```

Ahora puede ver la cola y comprobar el estado del trabajo con el comando `squeue`. Tenga en cuenta que, dado que no especificó una cola específica, se utiliza la cola predeterminada (`spot`). Para obtener más información acerca de `squeue`, consulte [squeue](#) en la Slurmdocumentación.

```
$ squeue
      JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
         2      spot    wrap ec2-user  R      0:10     2 spot-dy-
t2micro-1,spot-st-t2micro-1
```

El resultado muestra que el trabajo se encuentra actualmente en estado de ejecución. Espere 30 segundos a que el trabajo se termine y, a continuación, vuelva a ejecutar `squeue`.

```
$ squeue
      JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
```

Ahora que todos los trabajos de la cola han terminado, busca el archivo de salida `slurm-2.out` en tu directorio actual.

```
$ cat slurm-2.out
Hello World from spot-dy-t2micro-1
Hello World from spot-st-t2micro-1
```

El resultado también muestra que nuestro trabajo se ha ejecutado correctamente en los nodos `spot-st-t2micro-1` y `spot-st-t2micro-2`.

Ahora envíe el mismo trabajo especificando las restricciones para instancias específicas con los siguientes comandos.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 3
```

Ha utilizado estos parámetros `parasbatch`.

- `-N 3`— solicita tres nodos
- `-p spot`— envía el trabajo a la spot cola. También puede enviar un trabajo a la ondemand cola especificándolo. `-p ondemand`
- `-C "[c5.xlarge*1&t2.micro*2]"`— especifica las restricciones de nodo específicas para este trabajo. Esto requiere que se utilicen un (1) `c5.xlarge` nodo y dos (2) `t2.micro` nodos para este trabajo.

Ejecute el `sinfo` comando para ver los nodos y las colas. (Las colas de entrada AWS ParallelCluster se denominan particiones de entrada) Slurm.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand   up    infinite    10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*      up    infinite     1  mix#  spot-dy-c5xlarge-1
spot*      up    infinite    17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*      up    infinite     2  alloc spot-dy-t2micro-1,spot-st-t2micro-1
```

Los nodos se están encendiendo. Esto se indica con el `#` sufijo del estado del nodo. Ejecute el `squeue` comando para ver información sobre los trabajos del clúster.

```
$ squeue
          JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
           3      spot      wrap ec2-user CF      0:04     3 spot-dy-
c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```

Su trabajo está en el estado CF (CONFIGURING), esperando a que las instancias se amplíen y se unan al clúster.

Transcurridos unos tres minutos, los nodos deberían estar disponibles y el trabajo pasará al estado R (RUNNING).

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand   up    infinite    10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*      up    infinite    17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*      up    infinite     1  mix  spot-dy-c5xlarge-1
spot*      up    infinite     2  alloc spot-dy-t2micro-1,spot-st-t2micro-1
$ squeue
          JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
```

```

3      spot      wrap ec2-user R      0:04      3 spot-dy-
c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1

```

El trabajo finaliza y los tres nodos están en ese `idle` estado.

```

$ squeue
      JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand  up    infinite   10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*     up    infinite   17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*     up    infinite    3  idle  spot-dy-c5xlarge-1,spot-dy-t2micro-1,spot-st-
t2micro-1

```

A continuación, cuando no quede ningún trabajo en la cola, puede buscarlo `slurm-3.out` en su directorio local.

```

$ cat slurm-3.out
Hello World from spot-dy-c5xlarge-1
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1

```

El resultado también muestra que el trabajo se ejecutó correctamente en los nodos correspondientes.

Puede observar el proceso de reducción de escala. En la configuración de su clúster, especificó una duración [scaledown_idletime](#) habitual de 5 minutos. Tras cinco minutos en estado inactivo, los nodos dinámicos `spot-dy-t2micro-1` se reducen automáticamente `spot-dy-c5xlarge-1` y entran en modo `POWER_DOWN`. Ten en cuenta que el nodo estático `spot-st-t2micro-1` no se reduce.

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand  up    infinite   10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*     up    infinite    2  idle% spot-dy-c5xlarge-1,spot-dy-t2micro-1
spot*     up    infinite   17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*     up    infinite    1  idle  spot-st-t2micro-1

```

En el código anterior, puede ver que `spot-dy-c5xlarge-1` y `spot-dy-t2micro-1` están en modo `POWER_DOWN`. Se indica mediante el sufijo `%`. Las instancias correspondientes se cierran inmediatamente, pero los nodos permanecen en ese `POWER_DOWN` estado y no están disponibles

para su uso durante 120 segundos (dos minutos). Transcurrido este tiempo, los nodos vuelven a ahorrar energía y están disponibles para su uso de nuevo. Para obtener más información, consulte [Slurm guía para el modo de cola múltiple](#).

Este debería ser el estado final del clúster:

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand   up    infinite    10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*      up    infinite    19  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2micro-[1-9]
spot*      up    infinite     1  idle spot-st-t2micro-1
```

Cuando realice la sesión en el clúster, podrá eliminarla al abandonar el clúster `pcluster delete`. Para obtener más información sobre `pcluster list` y `pcluster delete`, consulte [pcluster list](#) y [pcluster delete](#).

```
$ pcluster list
multi-queue CREATE_COMPLETE 2.11.9
$ pcluster delete multi-queue
Deleting: multi-queue
...
```

Ejecutar trabajos en un clúster con instancias de EFA y GPU

En esta parte del tutorial se detalla cómo modificar la configuración y lanzar un clúster con varias colas que contenga instancias con recursos de GPU y redes EFA. Tenga en cuenta que las instancias utilizadas en este tutorial son instancias de mayor precio.

Comprueba los límites de tu cuenta para asegurarte de que estás autorizado a usar estas instancias antes de continuar con los pasos descritos en este tutorial.

Modifique el archivo con las siguientes configuraciones.

```
[global]
update_check = true
sanity_check = true
cluster_template = multi-queue-special

[aws]
aws_region_name = <Your Región de AWS>
```

```
[scaling demo]
scaledown_idletime = 5

[cluster multi-queue-special]
key_name = <Your SSH key name>
base_os = alinux2                # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge  # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = efa,gpu

[queue gpu]
compute_resource_settings = gpu_i1
disable_hyperthreading = true    # optional, defaults to false

[compute_resource gpu_i1]
instance_type = g3.8xlarge

[queue efa]
compute_resource_settings = efa_i1
enable_efa = true
placement_group = DYNAMIC       # optional, defaults to no placement group settings

[compute_resource efa_i1]
instance_type = c5n.18xlarge
max_count = 5
```

Creación del clúster

```
$ pcluster create multi-queue-special -t multi-queue-special
```

Una vez creado el clúster, usa tu archivo de clave SSH privado para iniciar sesión en tu nodo principal.

```
$ pcluster ssh multi-queue-special -i ~/path/to/keyfile.pem
```

Este debería ser el estado inicial del clúster:

```
$ sinfo
```

| PARTITION | AVAIL | TIMELIMIT | NODES | STATE | NODELIST |
|-----------|-------|-----------|-------|-------|--------------------------|
| efa* | up | infinite | 5 | idle~ | efa-dy-c5n18xlarge-[1-5] |
| gpu | up | infinite | 10 | idle~ | gpu-dy-g38xlarge-[1-10] |

En esta sección se describe cómo enviar algunos trabajos para comprobar que los nodos tienen recursos de EFA o GPU.

En primer lugar, escribe los scripts de trabajo. `efa_job.sh` dormirá por 30 segundos. Después de lo cual, busque EFA en la salida del comando `lspci`. `gpu_job.sh` dormirá por 30 segundos. Después de lo cual, ejecute `nvidia-smi` para mostrar a la GPU información sobre el nodo.

```
$ cat efa_job.sh
#!/bin/bash

sleep 30
lspci | grep "EFA"

$ cat gpu_job.sh
#!/bin/bash

sleep 30
nvidia-smi

$ chmod +x efa_job.sh
$ chmod +x gpu_job.sh
```

Envíe el trabajo con `sbatch`,

```
$ sbatch -p efa --wrap "srun efa_job.sh"
Submitted batch job 2
$ sbatch -p gpu --wrap "srun gpu_job.sh" -G 1
Submitted batch job 3
$ squeue
          JOBID PARTITION      NAME      USER ST       TIME  NODES NODELIST(REASON)
           2      efa      wrap ec2-user CF       0:32     1 efa-dy-
c5n18xlarge-1
           3      gpu      wrap ec2-user CF       0:20     1 gpu-dy-g38xlarge-1
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*      up      infinite   1     mix#  efa-dy-c5n18xlarge-1
efa*      up      infinite   4     idle~ efa-dy-c5n18xlarge-[2-5]
gpu       up      infinite   1     mix#  gpu-dy-g38xlarge-1
```

```
gpu          up    infinite    9  idle~ gpu-dy-g38xlarge-[2-10]
```

Después de unos minutos, debería ver los nodos en línea y los trabajos en ejecución.

```
[ec2-user@ip-172-31-15-251 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*      up    infinite   4  idle~ efa-dy-c5n18xlarge-[2-5]
efa*      up    infinite   1  mix  efa-dy-c5n18xlarge-1
gpu       up    infinite   9  idle~ gpu-dy-g38xlarge-[2-10]
gpu       up    infinite   1  mix  gpu-dy-g38xlarge-1
[ec2-user@ip-172-31-15-251 ~]$ squeue
          JOBID PARTITION    NAME     USER ST       TIME  NODES NODELIST(REASON)
          4      gpu      wrap  ec2-user  R       0:06      1 gpu-dy-g38xlarge-1
          5      efa      wrap  ec2-user  R       0:01      1 efa-dy-
c5n18xlarge-1
```

Cuando se complete el trabajo, compruebe la salida. En el resultado del `slurm-2.out` archivo, puede ver que el EFA está presente en el nodo `efa-dy-c5n18xlarge-1`. En el resultado del `slurm-3.out` archivo, puedes ver que el `nvidia-smi` resultado contiene información sobre la GPU del nodo `gpu-dy-g38xlarge-1`.

```
$ cat slurm-2.out
00:06.0 Ethernet controller: Amazon.com, Inc. Elastic Fabric Adapter (EFA)

$ cat slurm-3.out
Thu Oct  1 22:19:18 2020
+-----+
| NVIDIA-SMI 450.51.05    Driver Version: 450.51.05    CUDA Version: 11.0    |
|-----+-----+-----+
| GPU  Name            Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                               |                  |              MIG M. |
|=====+=====+=====+
|   0   Tesla M60             Off   | 00000000:00:1D.0 Off  |             0 |
| N/A   28C    P0     38W / 150W |  0MiB / 7618MiB |    0%      Default |
|                               |                  |              N/A   |
+-----+-----+-----+
|   1   Tesla M60             Off   | 00000000:00:1E.0 Off  |             0 |
| N/A   36C    P0     37W / 150W |  0MiB / 7618MiB |   98%      Default |
|                               |                  |              N/A   |
+-----+-----+-----+
```

```

+-----+
| Processes:                                     |
| GPU  GI  CI          PID  Type  Process name                      GPU Memory |
|      ID  ID                                     Usage          |
|=====|
| No running processes found                   |
+-----+

```

Puede observar el proceso de reducción de tamaño. En la configuración del clúster, especificó previamente una duración habitual [scaledown_idletime](#) de cinco minutos. Como resultado, tras cinco minutos en estado inactivo, los nodos dinámicos `spot-dy-t2micro-1` se reducen automáticamente `spot-dy-c5xlarge-1` y entran en modo `POWER_DOWN`. Finalmente, los nodos entran en el modo de ahorro de energía y están disponibles para su uso nuevamente.

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*      up    infinite   1  idle% efa-dy-c5n18xlarge-1
efa*      up    infinite   4  idle~ efa-dy-c5n18xlarge-[2-5]
gpu       up    infinite   1  idle% gpu-dy-g38xlarge-1
gpu       up    infinite   9  idle~ gpu-dy-g38xlarge-[2-10]

# After 120 seconds
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*      up    infinite   5  idle~ efa-dy-c5n18xlarge-[1-5]
gpu       up    infinite  10  idle~ gpu-dy-g38xlarge-[1-10]

```

Cuando realice la sesión en el clúster, podrá eliminarla al abandonar el clúster [pcluster delete <cluster name>](#).

```

$ pcluster list
multi-queue-special CREATE_COMPLETE 2.11.9
$ pcluster delete multi-queue-special
Deleting: multi-queue-special
...

```

Para obtener más información, consulte [Slurm guía para el modo de cola múltiple](#).

Desarrollo

Puede utilizar las siguientes secciones para comenzar a desarrollar AWS ParallelCluster.

Important

Las siguientes secciones incluyen instrucciones para utilizar una versión personalizada de las recetas del libro de recetas y un paquete de nodos de AWS ParallelCluster personalizado. Esta información abarca un método avanzado de personalización de AWS ParallelCluster, que presenta posibles problemas que pueden ser difíciles de depurar. El equipo AWS ParallelCluster recomienda encarecidamente utilizar los scripts en [Acciones de arranque personalizadas](#) para la personalización, ya que los enlaces de instalación posteriores suelen ser más fáciles de depurar y más fáciles de transportar a través de las versiones de AWS ParallelCluster.

Temas

- [Configuración de un libro de recetas de AWS ParallelCluster personalizado](#)
- [Configuración de un paquete de nodos de AWS ParallelCluster personalizado](#)

Configuración de un libro de recetas de AWS ParallelCluster personalizado

Important

A continuación, se muestran instrucciones para utilizar una versión personalizada de las recetas del libro de recetas de AWS ParallelCluster. Se trata de un método avanzado de personalizar AWS ParallelCluster, que presenta posibles problemas que pueden ser difíciles de depurar. El equipo AWS ParallelCluster recomienda encarecidamente utilizar los scripts en [Acciones de arranque personalizadas](#) para la personalización, ya que los enlaces de instalación posteriores suelen ser más fáciles de depurar y más fáciles de transportar a través de las versiones de AWS ParallelCluster.

Pasos

1. Identifique el directorio de trabajo del libro de recetas de AWS ParallelCluster donde ha clonado el código del [libro de recetas de AWS ParallelCluster](#).

```
_cookbookDir=<path to cookbook>
```

2. Detecte la versión actual del libro de recetas de AWS ParallelCluster.

```
_version=$(grep version ${_cookbookDir}/metadata.rb|awk '{print $2}'|tr -d \')
```

3. Cree un archivo del libro de recetas de AWS ParallelCluster y calcule su md5.

```
cd "${_cookbookDir}"
_stashName=$(git stash create)
git archive --format tar --prefix="aws-parallelcluster-cookbook-${_version}/"
"${_stashName}:-HEAD" | gzip > "aws-parallelcluster-cookbook-${_version}.tgz"
md5sum "aws-parallelcluster-cookbook-${_version}.tgz" > "aws-parallelcluster-
cookbook-${_version}.md5"
```

4. Cree un bucket de Amazon S3 y cargue el archivo, su md5 y su fecha de última modificación en el bucket. Conceda permiso legible público a través de una ACL public-read.

```
_bucket=<the bucket name>
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.tgz s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.tgz
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.md5 s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.md5
aws s3api head-object --bucket ${_bucket} --key cookbooks/aws-parallelcluster-
cookbook-${_version}.tgz --output text --query LastModified > aws-parallelcluster-
cookbook-${_version}.tgz.date
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.tgz.date s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.tgz.date
```

5. Añada las siguientes variables al archivo de configuración de AWS ParallelCluster, en la [sección \[cluster\]](#).

```
custom_chef_cookbook = https://${_bucket}.s3.<the bucket region>.amazonaws.com/
cookbooks/aws-parallelcluster-cookbook-${_version}.tgz
extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
```

Note

A partir de AWS ParallelCluster versión 2.6.1, la mayoría de las recetas de instalación se omiten de forma predeterminada al lanzar nodos para mejorar los tiempos de inicio. Para omitir la mayoría de las recetas de instalación para mejorar los tiempos de inicio a costa de la compatibilidad con versiones anteriores, elimine "skip_install_recipes" : "no" de la clave `cluster` en la configuración [extra_json](#).

Configuración de un paquete de nodos de AWS ParallelCluster personalizado

Warning

A continuación, se muestran instrucciones para utilizar una versión personalizada del paquete de nodos de AWS ParallelCluster. Se trata de un método avanzado de personalizar AWS ParallelCluster, que presenta posibles problemas que pueden ser difíciles de depurar. El equipo AWS ParallelCluster recomienda encarecidamente utilizar los scripts en [Acciones de arranque personalizadas](#) para la personalización, ya que los enlaces de instalación posteriores suelen ser más fáciles de depurar y más fáciles de transportar a través de las versiones de AWS ParallelCluster.

Pasos

1. Identifique el directorio de trabajo del nodo de AWS ParallelCluster donde ha clonado el código de nodo de AWS ParallelCluster.

```
_nodeDir=<path to node package>
```

2. Detecte la versión actual del nodo de AWS ParallelCluster.

```
_version=$(grep "version = \" ${_nodeDir}/setup.py |awk '{print $3}' | tr -d \"\")
```

3. Cree un archivo del nodo de AWS ParallelCluster.

```
cd "${_nodeDir}"
```

```
_stashName=$(git stash create)
git archive --format tar --prefix="aws-parallelcluster-node-${_version}/"
"${_stashName:-HEAD}" | gzip > "aws-parallelcluster-node-${_version}.tgz"
```

4. Cree un bucket de Amazon S3 y cargue el archivo en este. Conceda permiso legible público a través de una ACL public-read.

```
_bucket=<the bucket name>
aws s3 cp --acl public-read aws-parallelcluster-node-${_version}.tgz s3://${_bucket}/
node/aws-parallelcluster-node-${_version}.tgz
```

5. Añada la siguiente variable al archivo de configuración de AWS ParallelCluster, en la [sección \[cluster\]](#).

```
extra_json = { "cluster" : { "custom_node_package" : "https://${_bucket}.s3.<the
bucket region>.amazonaws.com/node/aws-parallelcluster-node-${_version}.tgz",
"skip_install_recipes" : "no" } }
```

Note

A partir de AWS ParallelCluster versión 2.6.1, la mayoría de las recetas de instalación se omiten de forma predeterminada al lanzar nodos para mejorar los tiempos de inicio. Para omitir la mayoría de las recetas de instalación para mejorar los tiempos de inicio a costa de la compatibilidad con versiones anteriores, elimine "skip_install_recipes" : "no" de la clave `cluster` en la configuración [extra_json](#).

AWS ParallelCluster solución de problemas

La AWS ParallelCluster comunidad mantiene una página wiki que proporciona muchos consejos para la solución de problemas en la [AWS ParallelCluster GitHub wiki](#). Para obtener una lista de problemas conocidos, consulte [Problemas conocidos](#).

Temas

- [Recuperación y conservación de registros](#)
- [Solución de problemas de implementación de pilas](#)
- [Solución de problemas en varios clústeres en modo de cola](#)
- [Solución de problemas en clústeres en modo de cola única](#)
- [Problemas con los grupos de ubicación y el lanzamiento de instancias](#)
- [Directorios que no se pueden reemplazar](#)
- [Solución de problemas en Amazon DCV](#)
- [Solución de problemas en clústeres con integración de AWS Batch](#)
- [Solución de problemas cuando un recurso no se crea](#)
- [Solución de problemas IAM de tamaño de la política](#)
- [Compatibilidad adicional](#)

Recuperación y conservación de registros

Los registros son un recurso útil para solucionar problemas. Antes de poder usar los registros para solucionar problemas con sus recursos de AWS ParallelCluster, primero debe crear un archivo de registros del clúster. Siga los pasos descritos en el tema [Creación de un archivo de registros de un clúster](#) de la [AWS ParallelCluster GitHub wiki](#) para iniciar este proceso.

Si uno de sus clústeres en ejecución tiene problemas, debe colocar el clúster en ese estado STOPPED ejecutando el comando `pcluster stop <cluster_name>` antes de empezar a solucionar el problema. Esto evita incurrir en costes inesperados.

Si `pcluster` deja de funcionar o si desea eliminar un clúster sin dejar de conservar sus registros, ejecute el comando `pcluster delete --keep-logs <cluster_name>`. Al ejecutar este comando, se elimina el clúster, pero se conserva el grupo de registros que está almacenado en

Amazon CloudWatch. Para obtener información sobre este comando, consulte la documentación de [pcluster delete](#).

Solución de problemas de implementación de pilas

Si el clúster no se puede crear y revierte la creación de la pila, puede revisar los siguientes archivos de registro para diagnosticar el problema. Desea buscar el resultado de `ROLLBACK_IN_PROGRESS` en estos registros. El mensaje de error debe ser similar al siguiente:

```
$ pcluster create mycluster
Creating stack named: parallelcluster-mycluster
Status: parallelcluster-mycluster - ROLLBACK_IN_PROGRESS
Cluster creation failed. Failed events:
  - AWS::EC2::Instance MasterServer Received FAILURE signal with UniqueId
    i-07af1cb218dd6a081
```

Para diagnosticar el problema, vuelva a crear el clúster utilizando [pcluster create](#), incluida la marca `--norollback`. A continuación, SSH en el clúster:

```
$ pcluster create mycluster --norollback
...
$ pcluster ssh mycluster
```

Una vez que haya iniciado sesión en el nodo principal, encontrará tres archivos de registro principales que podrá usar para identificar el error.

- `/var/log/cfn-init.log` es el registro del script `cfn-init`. Compruebe primero este registro. Es probable que veas un error como `Command chef failed` de este registro. Consulte las líneas inmediatamente anteriores a esta línea para obtener información más específica relacionada con el mensaje de error. Para obtener más información, consulte [cfn-init](#).
- `/var/log/cloud-init.log` es el registro de [cloud-init](#). Si no ve nada en `cfn-init.log`, intente revisar este registro a continuación.
- `/var/log/cloud-init-output.log` es el resultado de los comandos ejecutados por [cloud-init](#). Esto incluye el resultado de `cfn-init`. En la mayoría de los casos, no es necesario consultar este registro para solucionar problemas de este tipo.

Solución de problemas en varios clústeres en modo de cola

Esta sección es relevante para los clústeres que se instalaron con AWS ParallelCluster la versión 2.9.0 y versiones posteriores con Slurm programador de tareas. Para obtener más información sobre el modo de colas múltiples, consulte [Modo de Cola múltiple](#).

Temas

- [Registros clave](#)
- [Solución de problemas de inicialización de nodos](#)
- [Solución de problemas de sustituciones y terminaciones inesperadas de nodos](#)
- [Reemplazar, terminar o apagar las instancias y nodos problemáticos](#)
- [Solución de otros problemas de nodos y trabajos conocidos](#)

Registros clave

En la siguiente tabla se proporciona una descripción general de los registros clave del nodo principal:

`/var/log/cfn-init.log`

Este es el registro de AWS CloudFormation inicio. Contiene todos los comandos que se ejecutaron al configurar una instancia. Es útil para solucionar problemas de inicialización.

`/var/log/chef-client.log`

Este es el registro del cliente de Chef. Contiene todos los comandos que se ejecutaron a través de CINC Chef/. Es útil para solucionar problemas de inicialización.

`/var/log/parallelcluster/slurm_resume.log`

Se trata de un registro ResumeProgram. Lanza instancias para nodos dinámicos y es útil para solucionar problemas de lanzamiento de nodos dinámicos.

`/var/log/parallelcluster/slurm_suspend.log`

Este es el registro de SuspendProgram. Se llama cuando las instancias se terminan en el caso de los nodos dinámicos y es útil para solucionar problemas de terminación de los nodos dinámicos. Cuando revise este registro, también debe comprobar el registro de `clustermgtd`.

`/var/log/parallelcluster/clustermgtd`

Este es el registro de `clustermgtd`. Se ejecuta como el daemon centralizado que gestiona la mayoría de las acciones operativas del clúster. Es útil para solucionar cualquier problema de inicio, finalización o funcionamiento del clúster.

`/var/log/slurmctld.log`

Este es el Slurm registro de daemon de control. AWS ParallelCluster no toma decisiones de escalado. Más bien, solo intenta lanzar recursos para satisfacer las Slurm requisitos. Es útil para problemas de escalado y asignación, problemas relacionados con el trabajo y cualquier problema de lanzamiento y finalización relacionado con el programador.

Estas son las notas clave para los nodos de cómputo:

`/var/log/cloud-init-output.log`

Este es el registro de [cloud-init](#). Contiene todos los comandos que se ejecutaron al configurar una instancia. Es útil para solucionar problemas de inicialización.

`/var/log/parallelcluster/computemgtd`

Este es el registro de `computemgtd`. Se ejecuta en cada nodo de cómputo para monitorizarlo en el raro caso de que el daemon `clustermgtd` del nodo principal esté desconectado. Es útil para solucionar problemas de terminación inesperados.

`/var/log/slurmd.log`

Este es el Slurm compute el registro de daemon. Es útil para solucionar problemas relacionados con la inicialización y los errores de computación.

Solución de problemas de inicialización de nodos

En esta sección, se explica cómo solucionar los problemas de inicialización de los nodos. Esto incluye los problemas en los que el nodo no puede iniciar, encender o unirse a un clúster.

Nodo principal:

Registros aplicables:

- `/var/log/cfn-init.log`

- `/var/log/chef-client.log`
- `/var/log/parallelcluster/clustermgtd`
- `/var/log/parallelcluster/slurm_resume.log`
- `/var/log/slurmctld.log`

Compruebe los registros de `/var/log/cfn-init.log` y `/var/log/chef-client.log`. Estos registros deben contener todas las acciones que se ejecutaron cuando se configuró el nodo principal. La mayoría de los errores que se producen durante la configuración deberían incluir un mensaje de error en el registro de `/var/log/chef-client.log`. Si se especifican scripts previos o posteriores a la instalación en la configuración del clúster, compruebe que el script se ejecute correctamente a través de los mensajes de registro.

Cuando se crea un clúster, el nodo principal debe esperar a que los nodos de procesamiento se unan al clúster antes de poder unirse al clúster. Por lo tanto, si los nodos de computación no se unen al clúster, el nodo principal también falla. Puede seguir uno de estos procedimientos, dependiendo del tipo de notas de computación que utilice para solucionar problemas de este tipo:

Nodos de computación dinámicos:

- Busque en el registro de `ResumeProgram (/var/log/parallelcluster/slurm_resume.log)` el nombre de su nodo de computación para ver si alguna vez se llamó a `ResumeProgram` con el nodo. (Si `ResumeProgram` nunca se llamó, puede comprobar el `slurmctld` registro (`/var/log/slurmctld.log`) para determinar si Slurm alguna vez intentó llamar `ResumeProgram` con el nodo.)
- Tenga en cuenta que los permisos incorrectos activados en `ResumeProgram` pueden provocar un error silencioso en `ResumeProgram`. Si utilizas una configuración personalizada AMI con modificaciones en la `ResumeProgram` configuración, comprueba que `ResumeProgram` sea propiedad del `slurm` usuario y que tenga el permiso `744 (rwxr--r--)`.
- Si se llama a `ResumeProgram`, compruebe si se ha lanzado una instancia para el nodo. Si no se lanzó ninguna instancia, debería poder ver un mensaje de error que describe el error de lanzamiento.
- Si se lanza la instancia, es posible que haya habido un problema durante el proceso de configuración. Debería ver la dirección IP privada y el ID de instancia correspondientes en el registro de `ResumeProgram`. Además, puede consultar los registros de configuración correspondientes a la instancia específica. Para obtener más información acerca de la solución de problemas relacionados con un nodo de computación, consulte la siguiente sección.

Nodos de computación estáticos:

- Compruebe el registro `clustermgtd (/var/log/parallelcluster/clustermgtd)` para ver si se han lanzado instancias para el nodo. Si no se lanzaron, debería haber un mensaje de error claro que detalle el error de lanzamiento.
- Si se lanza la instancia, hay algún problema durante el proceso de configuración. Debería ver la dirección IP privada y el ID de instancia correspondientes en el registro de `ResumeProgram`. Además, puede consultar los registros de configuración correspondientes a la instancia específica.
- Nodos de computación:
 - Registros aplicables:
 - `/var/log/cloud-init-output.log`
 - `/var/log/slurmd.log`
 - Si se ha lanzado un nodo de computación, compruebe primero que `/var/log/cloud-init-output.log` contenga los registros de configuración similares al registro de `/var/log/chef-client.log` del nodo principal. La mayoría de los errores que se producen durante la configuración deberían tener mensajes de error ubicados en el registro de `/var/log/cloud-init-output.log`. Si se especifican scripts previos o posteriores a la instalación en la configuración del clúster, compruebe que se hayan ejecutado correctamente.
 - Si utilizas una personalización AMI con una modificación de Slurm configuración, entonces puede haber una Slurm error relacionado que impide que el nodo de cómputo se una al clúster. Para ver los errores relacionados con el programador, consulte el registro de `/var/log/slurmd.log`.

Solución de problemas de sustituciones y terminaciones inesperadas de nodos

En esta sección se continúa analizando cómo puede solucionar problemas relacionados con los nodos, específicamente cuando un nodo se reemplaza o se cierra inesperadamente.

- Registros aplicables:
 - `/var/log/parallelcluster/clustermgtd` (nodo principal)
 - `/var/log/slurmctld.log` (nodo principal)
 - `/var/log/parallelcluster/computemgtd` (nodo de computación)
- Nodos sustituidos o finalizados de forma inesperada:

- Consulte el registro de `clustermgtd (/var/log/parallelcluster/clustermgtd)` para ver si `clustermgtd` se tomó la medida necesaria para reemplazar o terminar un nodo. Tenga en cuenta que `clustermgtd` gestiona todas las acciones normales de mantenimiento del nodo.
- Si `clustermgtd` se reemplaza o se cierra el nodo, debería haber un mensaje que detalle por qué se realizó esta acción en el nodo. Si el motivo está relacionado con el programador (por ejemplo, el nodo está en DOWN), consulte el registro `slurmctld` para obtener más detalles. Si el motivo está EC2 relacionado con Amazon, debería haber un mensaje informativo que detalle el problema EC2 relacionado con Amazon que requirió el reemplazo.
- Si `clustermgtd` no canceló el nodo, compruebe primero si se trataba de una terminación prevista por parte de AmazonEC2, más específicamente, de una terminación puntual. `computemgtd`, que se ejecuta en un nodo de cómputo, también puede realizar una acción para cerrar un nodo si `clustermgtd` se determina que está en mal estado. Compruebe el registro de `computemgtd (/var/log/parallelcluster/computemgtd)` para ver si `computemgtd` ha terminado el nodo.
- Los nodos fallaron
 - Compruebe el registro de `slurmctld (/var/log/slurmctld.log)` para ver por qué ha fallado un trabajo o un nodo. Tenga en cuenta que los trabajos se vuelven a poner en cola automáticamente si se produce un error en un nodo.
 - Si `slurm_resume` informa de que el nodo se ha lanzado y, después de varios minutos, `clustermgtd` informa de que no hay ninguna instancia correspondiente en Amazon EC2 para ese nodo, es posible que el nodo produzca un error durante la configuración. Para recuperar el registro de un compute (`/var/log/cloud-init-output.log`), siga estos pasos:
 - Envía un trabajo para alquilarlo Slurm hacer girar un nuevo nodo.
 - Una vez que se inicie el nodo, habilite la protección de terminación mediante este comando.

```
aws ec2 modify-instance-attribute --instance-id i-xyz --disable-api-termination
```

- Recupere la salida de la consola del nodo con este comando.

```
aws ec2 get-console-output --instance-id i-xyz --output text
```

Reemplazar, terminar o apagar las instancias y nodos problemáticos

- Registros aplicables:
 - `/var/log/parallelcluster/clustermgtd` (nodo principal)

- `/var/log/parallelcluster/slurm_suspend.log` (nodo principal)
- En la mayoría, `clustermgtd` administra todas las acciones de finalización de instancias esperadas. Consulte el registro `clustermgtd` para ver por qué no se ha podido sustituir o finalizar un nodo.
- En el caso de los nodos dinámicos que no superan el [scaledown_idletime](#), consulte el registro de `SuspendProgram` para ver si lo llamó `slurmctld` con el nodo específico como argumento. Tenga en cuenta que `SuspendProgram` no realiza ninguna acción específica. Más bien, solo se encarga de registrar cuando se le llama. La terminación de todas las instancias y el restablecimiento de `NodeAddr` los realiza `clustermgtd`. Slurm devuelve los nodos a un `POWER_SAVING` estado posterior `SuspendTimeout` automáticamente.

Solución de otros problemas de nodos y trabajos conocidos

Otro tipo de problema conocido es que es AWS ParallelCluster posible que no se puedan asignar las tareas ni tomar decisiones de escalado. Con este tipo de problema, AWS ParallelCluster solo se lanzan, finalizan o mantienen los recursos de acuerdo con Slurm instrucciones. En el caso de estos problemas, consulte el registro de `slurmctld` para solucionarlos.

Solución de problemas en clústeres en modo de cola única

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

Esta sección se aplica a los clústeres que no tienen el modo de cola múltiple con una de las dos configuraciones siguientes:

- Lanzado con una AWS ParallelCluster versión anterior a la 2.9.0 y SGE, Torque, o Slurm programadores de trabajos.
- Lanzado con la AWS ParallelCluster versión 2.9.0 o posterior y SGE o Torque programadores de tareas.

Temas

- [Registros clave](#)

- [Solución de problemas en las operaciones de inicio y unión fallidas](#)
- [Solución de problemas de escalar](#)
- [Solución de otros problemas relacionados con el clúster](#)

Registros clave

Los siguientes archivos de registro son los registros clave del nodo principal.

Para la AWS ParallelCluster versión 2.9.0 o posterior:

```
/var/log/chef-client.log
```

Este es el registro del cliente CINC (chef). Contiene todos los comandos que se ejecutaron CINC. Es útil para solucionar problemas de inicialización.

Para todas AWS ParallelCluster las versiones:

```
/var/log/cfn-init.log
```

Este es el registro de `cfn-init`. Contiene todos los comandos que se ejecutaron al configurar una instancia y, por lo tanto, resulta útil para solucionar problemas de inicialización. Para obtener más información, consulte [cfn-init](#).

```
/var/log/clustermgtd.log
```

Este es el `clustermgtd` registro de Slurm programadores. `clustermgtd` se ejecuta como el daemon centralizado que gestiona la mayoría de las acciones operativas del clúster. Es útil para solucionar cualquier problema de inicio, finalización o funcionamiento del clúster.

```
/var/log/jobwatcher
```

Este es el `jobwatcher` registro de SGE y Torque programadores. `jobwatcher` supervisa la cola del planificador y actualiza el Grupo de Auto Scaling. Es útil para solucionar problemas relacionados con el escalado vertical de los nodos.

```
/var/log/sqswatcher
```

Este es el registro de `sqswatcher` SGE y Torque programadores. `sqswatcher` procesa el evento listo para la instancia enviado por una instancia de cómputo tras una inicialización correcta. También agrega nodos de cómputo a la configuración del programador. Este registro es útil para solucionar los motivos por los que uno o varios nodos no pudieron unirse a un clúster.

A continuación, se muestran los registros clave de los nodos de procesamiento.

AWS ParallelCluster versión 2.9.0 o posterior

`/var/log/cloud-init-output.log`

Este es el registro de inicio de la nube. Contiene todos los comandos que se ejecutaron al configurar una instancia. Es útil para solucionar problemas de inicialización.

AWS ParallelCluster versiones anteriores a la 2.9.0

`/var/log/cfn-init.log`

Este es el registro de CloudFormation inicio. Contiene todos los comandos que se ejecutaron al configurar una instancia. Es útil para solucionar problemas de inicialización

Todas las versiones

`/var/log/nodewatcher`

Este es el `nodewatcher` registro. `nodewatcherdemonios` que se ejecutan en cada nodo de cómputo cuando se utilizan SGE y Torque planificadores. Reducen verticalmente un nodo si está inactivo. Este registro es útil para cualquier problema relacionado con la reducción vertical de los recursos.

Solución de problemas en las operaciones de inicio y unión fallidas

- Registros aplicables:
 - `/var/log/cfn-init-cmd.log` (nodo principal y nodo de computación)
 - `/var/log/sqswatcher` (nodo principal)
- Si los nodos no se pudieron iniciar, consulte el registro de `/var/log/cfn-init-cmd.log` para ver el mensaje de error específico. En la mayoría de los casos, los errores en el lanzamiento de los nodos se deben a un error de configuración.
- Si los nodos de cómputo no pudieron unirse a la configuración del programador a pesar de haberse configurado correctamente, compruebe en el registro de `/var/log/sqswatcher` si `sqswatcher` procesó el evento. En la mayoría de los casos, estos problemas se deben a `sqswatcher` que no procesó el evento.

Solución de problemas de escalar

- Registros aplicables:
 - `/var/log/jobwatcher` (nodo principal)
 - `/var/log/nodewatcher` (nodo de computación)
- Problemas de escalado vertical: en el caso del nodo principal, compruebe el registro de `/var/log/jobwatcher` para ver si el daemon de `jobwatcher` calculó el número correcto de nodos necesarios y actualizó el grupo de escalado automático. Tenga en cuenta que `jobwatcher` monitorea la cola del programador y actualiza el grupo de escalado automático.
- Problemas de reducción de escala vertical: en el caso de los nodos de cómputo, compruebe el registro de `/var/log/nodewatcher` del nodo problemático para ver por qué se ha reducido verticalmente el nodo. Tenga en cuenta que los daemons de `nodewatcher` reducen la escala de un nodo de computación si está inactivo.

Solución de otros problemas relacionados con el clúster

Un problema conocido es que una nota de cálculo aleatoria falla en los clústeres de gran escala, específicamente en aquellos con 500 o más nodos de procesamiento. Este problema está relacionado con una limitación de la arquitectura de escalado de los clústeres de cola única. Si desea utilizar un clúster a gran escala, está utilizando la AWS ParallelCluster versión v2.9.0 o posterior, utilice Slurm, y para evitar este problema, debe actualizar y cambiar a un clúster compatible con el modo de cola múltiple. Puede hacerlo ejecutando [pcluster-config convert](#).

En el caso de los clústeres a gran escala, es posible que sea necesario realizar ajustes adicionales en el sistema. Para obtener más información, póngase en contacto con AWS Support.

Problemas con los grupos de ubicación y el lanzamiento de instancias

Para obtener la latencia entre nodos más baja, use un grupo de ubicación. Un grupo de ubicación garantiza que las instancias estén en la misma red troncal. Si no hay suficientes instancias disponibles cuando se realiza la solicitud, se devuelve un error `InsufficientInstanceCapacity`. Para reducir la posibilidad de recibir este error al utilizar grupos con ubicación en clúster, establezca el parámetro [placement_group](#) en `DYNAMIC` y establezca el parámetro [placement](#) en `compute`.

Si necesita un sistema de archivos compartidos de alto rendimiento, considere usarlo [FSxpara Lustre](#).

Si el nodo principal debe estar en el grupo de ubicación, utilice el mismo tipo de instancia y subred para los nodos principal y de computación. Al hacer esto, el parámetro [compute_instance_type](#) tiene el mismo valor que el parámetro [master_instance_type](#); el parámetro [placement](#) está establecido en `cluster`; y el parámetro [compute_subnet_id](#) no se ha especificado. Con esta configuración, el parámetro [master_subnet_id](#) se utiliza para los nodos de computación.

Para obtener más información, consulta la sección [Solución de problemas de lanzamiento de instancias](#) y las [funciones y limitaciones de los grupos de ubicación](#) en la Guía del EC2 usuario de Amazon

Directorios que no se pueden reemplazar

Los siguientes directorios se comparten entre los nodos y no se pueden reemplazar.

`/home`

Esto incluye la carpeta de inicio del usuario predeterminada (`/home/ec2_user` en Amazon Linux, `/home/centos` en CentOS, y así `/home/ubuntu` sucesivamente Ubuntu).

`/opt/intel`

Esto incluye IntelMPI, Intel Parallel Studio y archivos relacionados.

`/opt/sge`

Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

Esto incluye Son of Grid Engine y archivos relacionados. (Condicional, solo si [scheduler](#) = `sge`).

`/opt/slurm`

Esto incluye Slurm Workload Manager y archivos relacionados. (Condicional, solo si [scheduler](#) = `slurm`).

/opt/torque

 Note

A partir de la versión 2.11.5, AWS ParallelCluster no admite el uso de SGE o Torque planificadores.

Esto incluye Torque Resource Manager y archivos relacionados. (Condicional, solo si [scheduler](#) = torque).

Solución de problemas en Amazon DCV

Temas

- [Logs para Amazon DCV](#)
- [Memoria de tipo DCV instancia de Amazon](#)
- [DCVProblemas con Ubuntu Amazon](#)

Logs para Amazon DCV

Los registros de Amazon DCV se escriben en los archivos del `/var/log/dcv/` directorio. Revisar estos registros puede ayudar a solucionar problemas.

Memoria de tipo DCV instancia de Amazon

El tipo de instancia debe tener al menos 1,7 gibibyte (GiB) para RAM ejecutar Amazon. DCV Nano y micro los tipos de instancia no tienen memoria suficiente para ejecutar AmazonDCV.

DCVProblemas con Ubuntu Amazon

Al ejecutar la Terminal de Gnome durante una DCV sesión en Ubuntu, es posible que no tengas acceso automáticamente al entorno de usuario que AWS ParallelCluster está disponible a través del shell de inicio de sesión. El entorno de usuario proporciona módulos de entorno, como `openmpi` o `intelmpi`, y otros ajustes de usuario.

La configuración predeterminada de la Terminal de Gnome impide que el intérprete de comandos se inicie como un intérprete de comandos de inicio de sesión. Esto significa que los perfiles de shell no se obtienen automáticamente y el entorno AWS ParallelCluster de usuario no está cargado.

Para obtener correctamente el perfil de shell y acceder al entorno AWS ParallelCluster de usuario, realice una de las siguientes acciones:

- Cambie la configuración predeterminada del terminal:
 1. Seleccione el menú Editar en la terminal de Gnome.
 2. Seleccione Preferencias y, a continuación, Perfiles.
 3. Elija Comando y seleccione Ejecutar comando como intérprete de comandos de inicio de sesión.
 4. Abrir una nueva terminal.
- Utilice la línea de comandos para obtener los perfiles disponibles:

```
$ source /etc/profile && source $HOME/.bashrc
```

Solución de problemas en clústeres con integración de AWS Batch

Esta sección es relevante para los clústeres con integración de AWS Batch planificadores.

Problemas con el nodo principal

Los problemas de configuración relacionados con el nodo principal se pueden solucionar de la misma manera que con un clúster de cola única. Para obtener más información sobre estos problemas, consulte [Solución de problemas en clústeres en modo de cola única](#).

AWS Batch problemas de envío de trabajos paralelos de varios nodos

Si tiene problemas para enviar trabajos paralelos de varios nodos al utilizarlos AWS Batch como programador de trabajos, debe actualizar a la AWS ParallelCluster versión 2.5.0. Si eso no es posible, puede utilizar la solución alternativa que se detalla en el tema: [Self patch a cluster used for submitting multi-node parallel jobs through AWS Batch](#).

Problemas informáticos

AWS Batch gestiona los aspectos de escalado y computación de sus servicios. Si tiene problemas relacionados con la informática, consulte la documentación AWS Batch [de solución de problemas](#) para obtener ayuda.

Errores en los trabajos

Si se produce un error en un trabajo, puede ejecutar el comando [awsbout](#) para recuperar el resultado del trabajo. También puedes ejecutar el [awsbstat](#) -d comando para obtener un enlace a los registros de trabajos almacenados por Amazon CloudWatch.

Solución de problemas cuando un recurso no se crea

Esta sección es relevante para los recursos del clúster cuando no se pueden crear.

Cuando no se puede crear un recurso, ParallelCluster devuelve un mensaje de error como el siguiente.

```
pcluster create -c config my-cluster
Beginning cluster creation for cluster: my-cluster
WARNING: The instance type 'p4d.24xlarge' cannot take public IPs. Please make sure that
the subnet with
id 'subnet-1234567890abcdef0' has the proper routing configuration to allow private IPs
reaching the
Internet (e.g. a NAT Gateway and a valid route table).
WARNING: The instance type 'p4d.24xlarge' cannot take public IPs. Please make sure that
the subnet with
id 'subnet-1234567890abcdef0' has the proper routing configuration to allow private IPs
reaching the Internet
(e.g. a NAT Gateway and a valid route table).
Info: There is a newer version 3.0.3 of AWS ParallelCluster available.
Creating stack named: parallelcluster-my-cluster
Status: parallelcluster-my-cluster - ROLLBACK_IN_PROGRESS
Cluster creation failed. Failed events:
- AWS::CloudFormation::Stack MasterServerSubstack Embedded stack
arn:aws:cloudformation:region-id:123456789012:stack/parallelcluster-my-cluster-
MasterServerSubstack-ABCDEFGHIJKL/a1234567-b321-c765-d432-dcba98766789
was not successfully created:
The following resource(s) failed to create: [MasterServer].
```

```
- AWS::CloudFormation::Stack parallelcluster-my-cluster-MasterServerSubstack-
ABCDEFGHIJKL The following resource(s) failed to create: [MasterServer].
- AWS::EC2::Instance MasterServer You have requested more vCPU capacity than your
  current vCPU limit of 0 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.
(Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request ID:
 a9876543-b321-c765-d432-dcba98766789; Proxy: null)
}
```

Por ejemplo, si ves el mensaje de estado que se muestra en la respuesta del comando anterior, debes usar tipos de instancias que no superen tu CPU límite v actual o solicitar más CPU capacidad de v.

También puedes usar la CloudFormation consola para ver información sobre el "Cluster creation failed" estado.

Vea los mensajes de CloudFormation error de la consola.

1. Inicie sesión en /cloudformation AWS Management Console y dirijase a <https://console.aws.amazon.com/cloudformation>.
2. Seleccione la pila denominada parallelcluster-*cluster_name*.
3. Seleccione la pestaña Eventos.
4. Compruebe el estado del recurso que no se pudo crear desplazándose por la lista de eventos del recurso por identificador lógico. Si no se pudo crear una subtarea, retroceda para encontrar el evento de recurso fallido.
5. Un ejemplo de AWS CloudFormation mensaje de error:

```
2022-02-07 11:59:14 UTC-0800 MasterServerSubstack CREATE_FAILED Embedded stack
arn:aws:cloudformation:region-id:123456789012:stack/parallelcluster-my-cluster-
MasterServerSubstack-ABCDEFGHIJKL/a1234567-b321-c765-d432-dcba98766789
was not successfully created: The following resource(s) failed to create:
 [MasterServer].
```

Solución de problemas IAM de tamaño de la política

Consulte las [IAM AWS STS cuotas, los requisitos de nombres y los límites de caracteres](#) para comprobar las cuotas de las políticas gestionadas asociadas a las funciones. Si el tamaño de una

política administrada supera la cuota, divida la política en dos o más políticas. Si superas la cuota de políticas asociadas a un IAM rol, crea roles adicionales y distribuye las políticas entre ellos para cumplir con la cuota.

Compatibilidad adicional

Para ver una lista de problemas conocidos, consulta la página principal de la [GitHubwiki](#) o la página de [problemas](#). Para problemas más urgentes, contacta AWS Support o abre un [nuevo GitHub número](#).

Política de compatibilidad de AWS ParallelCluster

AWS ParallelCluster admite varios lanzamientos al mismo tiempo. Cada lanzamiento AWS ParallelCluster tiene una fecha de fin de vida útil (EOSL) programada. A partir de la fecha de EOSL no se proporcionará más soporte ni mantenimiento para ese lanzamiento.

AWS ParallelCluster utiliza un esquema de versión `major.minor.patch`. Las nuevas versiones secundarias de la última versión principal incluyen nuevas funciones, mejoras de rendimiento, actualizaciones de seguridad y correcciones de errores. Las versiones secundarias son compatibles con versiones anteriores de una versión principal. Para los problemas críticos, AWS proporciona correcciones mediante versiones de parches, pero solo para las versiones secundarias más recientes de las versiones que no han llegado a la EOSL. Si desea utilizar las actualizaciones de una versión nueva, debe actualizar a la nueva versión secundaria o a la nueva versión de parche.

| Versiones de AWS ParallelCluster | Fecha de fin de compatibilidad (EOSL) |
|----------------------------------|---------------------------------------|
| 210.4 y versiones posteriores | 12/31/2021 |
| 2.11.x | 12/31/2022 |

Seguridad en AWS ParallelCluster

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 usted. El [modelo de responsabilidad compartida](#) describe esto como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS Programas de conformidad de](#) . Para obtener información sobre los programas de conformidad que se aplican a AWS ParallelCluster, consulte [Servicios de AWS en el ámbito del programa de conformidad](#).
- Seguridad en la nube: su responsabilidad viene determinada por el servicio o servicios específicos de AWS que utilice. También es responsable de otros factores varios, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación describe cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS ParallelCluster. En los siguientes temas, se le mostrará cómo configurar AWS ParallelCluster para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar AWS ParallelCluster de manera que le ayudará a supervisar y a proteger los recursos de AWS.

Temas

- [Información de seguridad de los servicios utilizados por AWS ParallelCluster](#)
- [Protección de datos en AWS ParallelCluster](#)
- [Identity and Access Management para AWS ParallelCluster](#)
- [Validación de la conformidad en AWS ParallelCluster](#)
- [Aplicación de una versión mínima de TLS 1.2](#)

Información de seguridad de los servicios utilizados por AWS ParallelCluster

- [Seguridad en Amazon EC2](#)
- Seguridad en Amazon API Gateway
- [Seguridad en AWS Batch](#)
- [Seguridad en AWS CloudFormation](#)
- Seguridad en Amazon CloudWatch
- [Seguridad en AWS CodeBuild](#)
- [Seguridad en Amazon DynamoDB](#)
- [Seguridad en Amazon ECR](#)
- [Seguridad en Amazon ECS](#)
- [Seguridad en Amazon EFS](#)
- [Seguridad en FSx para Lustre](#)
- [Seguridad en AWS Identity and Access Management \(IAM\)](#)
- [Seguridad en EC2 Image Builder](#)
- [Seguridad en AWS Lambda](#)
- Seguridad en Amazon Route 53
- [Seguridad en Amazon SNS](#)
- [Seguridad en Amazon SQS \(para AWS ParallelCluster versión 2.x.\)](#)
- [Seguridad en Amazon S3](#)
- [Seguridad en Amazon VPC](#)

Protección de datos en AWS ParallelCluster

La AWS [modelo de responsabilidad compartida](#) de se aplica a la protección de datos en AWS ParallelCluster. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido que está alojado en esta infraestructura. También es responsable de las tareas de configuración y administración de la seguridad del Servicios de AWS que utilices. Para obtener más información sobre la privacidad de los datos, consulte la sección [Privacidad de datos FAQ](#). Para obtener información sobre la protección de datos en Europa,

consulte la [AWS Modelo de responsabilidad compartida y entrada de GDPR](#) blog sobre AWS Blog de seguridad.

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- Utilice SSL/TLS para comunicarse con AWS recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad del usuario con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Trabajar con CloudTrail senderos](#) en la AWS CloudTrail Guía del usuario.
- Use AWS soluciones de cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder AWS a través de una interfaz de línea de comandos o API, utilice un FIPS punto final. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la [Norma Federal de Procesamiento de Información \(FIPS\) 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS ParallelCluster u otro Servicios de AWS utilizando la consola API, AWS CLI, o AWS SDKs. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar su solicitud a ese servidor.

Cifrado de datos

Una característica clave de cualquier servicio seguro es que la información se cifre cuando no se está utilizando activamente.

Cifrado en reposo

AWS ParallelCluster no almacena en sí mismo ningún dato del cliente aparte de las credenciales que necesita para interactuar con el AWS servicios en nombre del usuario.

En el caso de los datos de los nodos del clúster, los datos se pueden cifrar en reposo.

Para EBS los volúmenes de Amazon, el cifrado se configura mediante los [ebs_kms_key_id](#) ajustes de la [\[ebs\]sección](#) de AWS ParallelCluster versión 2.x.) Para obtener más información, consulta el [EBScifrado de Amazon](#) en la Guía del EC2 usuario de Amazon.

Para EFS los volúmenes de Amazon, el cifrado se configura mediante los [efs_kms_key_id](#) ajustes [encrypted](#) y de la [\[efs\]sección](#) de AWS ParallelCluster versión 2.x). Para obtener más información, consulte [Cómo funciona el cifrado en reposo](#) en la Guía del usuario de Amazon Elastic File System.

En el FSx caso de los sistemas de archivos Lustre, el cifrado de los datos en reposo se habilita automáticamente al crear un sistema de FSx archivos de Amazon. Para obtener más información, consulta [Cómo cifrar datos en reposo en](#) la Guía del usuario de Amazon FSx for Lustre.

En el caso de los tipos con NVMe volúmenes, los datos de los volúmenes del almacén de NVMe instancias se cifran mediante un cifrado de XTS AES -256 implementado en un módulo de hardware de la instancia. Las claves de cifrado se generan mediante el módulo de hardware y son únicas para cada dispositivo de almacenamiento de la NVMe instancia. Todas las claves de cifrado se destruyen cuando se detiene o termina la instancia y no se pueden recuperar. No puede deshabilitar este cifrado ni tampoco proporcionar su propia clave de cifrado. Para obtener más información, consulta [Encriptación en reposo](#) en la Guía del EC2 usuario de Amazon.

Si usas AWS ParallelCluster para invocar un AWS servicio que transmite los datos de los clientes a su ordenador local para su almacenamiento; a continuación, consulte el capítulo sobre seguridad y conformidad de la Guía del usuario de ese servicio para obtener información sobre cómo se almacenan, protegen y cifran esos datos.

Cifrado en tránsito

De forma predeterminada, todos los datos transmitidos desde el equipo cliente en ejecución AWS ParallelCluster y AWS Los puntos finales del servicio se cifran enviando todo a través de una TLS conexiónHTTPS/. El tráfico entre los nodos del clúster se puede cifrar automáticamente, según los tipos de instancias seleccionados. Para obtener más información, consulta [Cifrado en tránsito](#) en la Guía del EC2 usuario de Amazon.

Véase también

- [Protección de datos en Amazon EC2](#)
- [Protección de datos en EC2 Image Builder](#)
- [Protección de datos en AWS CloudFormation](#)
- [Protección de datos en Amazon EFS](#)
- [Protección de los datos en Amazon S3](#)
- [La protección de datos es FSx para Lustre](#)

Identity and Access Management para AWS ParallelCluster

AWS ParallelCluster utiliza los mismos usuarios y roles para acceder a los recursos de AWS y sus servicios. Las políticas de instancia y usuario que se AWS ParallelCluster utilizan para conceder permisos se documentan en [AWS Identity and Access Management funciones en AWS ParallelCluster](#).

La única diferencia importante es cómo se realiza la autenticación cuando se utiliza un usuario de estándar y credenciales a largo plazo. Aunque los usuarios de AWS estándar necesitan una contraseña para acceder a la consola de un servicio de , ese mismo usuario de necesita un par de claves de acceso para realizar las mismas operaciones a través de AWS ParallelCluster. Todas las demás credenciales a corto plazo se utilizan de la misma manera que con la consola.

Las credenciales que se utilizan en AWS ParallelCluster se almacenan en archivos de texto sin formato y no se cifran.

- El archivo `$HOME/.aws/credentials` almacena las credenciales a largo plazo necesarias para acceder a los recursos de AWS. Cómo recuperar el ID de clave de acceso y la clave de acceso secreta
- Las credenciales a corto plazo, como las de los roles que se adoptan o que se utilizan para servicios de AWS IAM Identity Center, también se almacenan en las carpetas `$HOME/.aws/cli/cache` y `$HOME/.aws/sso/cache`, respectivamente.

Mitigación de riesgos

- Le recomendamos encarecidamente que configure los permisos del sistema de archivos en la carpeta `$HOME/.aws`, sus subcarpetas y archivos para restringir el acceso exclusivamente a los usuarios autorizados.
- Utilice roles con credenciales temporales siempre que sea posible para reducir la posibilidad de que se produzcan daños si las credenciales se ven comprometidas. Utilice credenciales a largo plazo solo para solicitar y actualizar las credenciales a corto plazo de los roles.

Validación de la conformidad en AWS ParallelCluster

Los auditores externos evalúan la seguridad y la conformidad de los servicios de AWS en distintos programas de conformidad de AWS. Usar AWS ParallelCluster para acceder a un servicio no altera la conformidad de dicho servicio.

Para obtener una lista de los servicios de AWS en el ámbito de programas de conformidad específicos, consulte [Servicios de AWS en el ámbito del programa de conformidad](#). 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 AWS ParallelCluster 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.
- [Arquitectura de las medidas de seguridad y cumplimiento de HIPAA en Amazon Web Services](#) [AWS Documento técnico](#) : en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones que cumplan los requisitos de la HIPAA.
- [Recursos de conformidad de AWS](#): este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- [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 vista integral de su estado de seguridad en AWS que lo ayuda a verificar la conformidad con los estándares y las prácticas recomendadas del sector de seguridad.

Aplicación de una versión mínima de TLS 1.2

Para aumentar la seguridad al comunicarse con los servicios de AWS, debe configurar la () de modo que use TLS 1.2 o una versión posterior. Cuando se utiliza AWS ParallelCluster, Python se usa para establecer la versión de TLS.

Para asegurarse de que AWS ParallelCluster no use ninguna versión de TLS anterior a TLS 1.2, es posible que tenga que volver a compilar OpenSSL para que aplique este mínimo y, a continuación, volver a compilar Python para que use el OpenSSL recién creado.

Determinar los protocolos admitidos actualmente

Primero, cree un certificado autofirmado con el fin de usarlo para el servidor de prueba y Python SDK mediante OpenSSL.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

A continuación, cree un servidor de prueba mediante OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

En una nueva ventana de terminal, cree un entorno virtual e instale Python SDK.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Cree un script de Python denominado `check.py` que use la biblioteca HTTP subyacente del SDK.

```
$ import urllib3
URL = 'https://localhost:4433/'
```

```
http = urllib3.PoolManager(
    ca_certs='cert.pem',
    cert_reqs='CERT_REQUIRED',
)
r = http.request('GET', URL)
print(r.data.decode('utf-8'))
```

Ejecute el nuevo script.

```
$ python check.py
```

Muestra detalles sobre la conexión realizada. Busque «Protocol:» en la salida. Si el resultado es «TLSv1.2» o posterior, el SDK predeterminado es TLS v1.2 o posterior. Si se trata de una versión anterior, debe volver a compilar OpenSSL y Python.

Sin embargo, aunque la instalación de Python se establezca de forma predeterminada en TLS v1.2 o posterior, es posible que Python renegocie a una versión anterior a TLS v1.2 si el servidor no admite TLS v1.2 o posterior. Para comprobar que Python no renegocia automáticamente a versiones anteriores, reinicie el servidor de prueba con lo siguiente.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Si está utilizando una versión anterior de OpenSSL, es posible que la opción `-no_tls1_3` no esté disponible. Si este es el caso, elimine la opción, porque la versión de OpenSSL que está utilizando no admite TLS v1.3. A continuación, vuelva a ejecutar el script de Python.

```
$ python check.py
```

Si su instalación de Python no renegocia correctamente las versiones anteriores a TLS 1.2, debería recibir un error de SSL.

```
$ urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='localhost',
port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL:
UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

Si puede establecer una conexión, debe volver a compilar OpenSSL y Python para deshabilitar la negociación de protocolos anteriores a TLS v1.2.

Compilar OpenSSL y Python

Para asegurarse de que AWS ParallelCluster no negocie a nada anterior a TLS 1.2, debe volver a compilar OpenSSL y Python. Para ello, copie el siguiente contenido con el fin de crear un script y ejecútelos.

```
#!/usr/bin/env bash
set -e

OPENSSL_VERSION="1.1.1d"
OPENSSL_PREFIX="/opt/openssl-with-min-tls1_2"
PYTHON_VERSION="3.8.1"
PYTHON_PREFIX="/opt/python-with-min-tls1_2"

curl -O "https://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz"
tar -xzf "openssl-$OPENSSL_VERSION.tar.gz"
cd openssl-$OPENSSL_VERSION
./config --prefix=$OPENSSL_PREFIX no-ssl3 no-tls1 no-tls1_1 no-shared
make > /dev/null
sudo make install_sw > /dev/null

cd /tmp
curl -O "https://www.python.org/ftp/python/$PYTHON_VERSION/Python-$PYTHON_VERSION.tgz"
tar -xzf "Python-$PYTHON_VERSION.tgz"
cd Python-$PYTHON_VERSION
./configure --prefix=$PYTHON_PREFIX --with-openssl=$OPENSSL_PREFIX --disable-shared > /dev/null
make > /dev/null
sudo make install > /dev/null
```

Se compila una versión de Python que tiene un OpenSSL enlazado estáticamente que no negocia automáticamente a nada anterior a TLS 1.2. Esto también instala OpenSSL en el directorio `/opt/openssl-with-min-tls1_2` directorio e instala Python en el directorio `/opt/python-with-min-tls1_2`. Después de ejecutar este script, confirme la instalación de la nueva versión de Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Se debería imprimir lo siguiente.

Python 3.8.1

Para confirmar que esta nueva versión de Python no negocia a una versión anterior de TLS 1.2, vuelva a ejecutar los pasos de [Determinar los protocolos admitidos actualmente](#) usando la versión de Python recién instalada (es decir, `/opt/python-with-min-tls1_2/bin/python3`).

Notas de la versión e historial de revisión

En la siguiente tabla se describen las principales actualizaciones y nuevas características de la Guía del usuario de AWS ParallelCluster . Actualizamos la documentación con frecuencia para dar respuesta a los comentarios que se nos envía.

| Cambio | Descripción | Fecha |
|---|---|------------------------|
| Publicación de la documentación únicamente | <p>AWS ParallelCluster Se ha publicado la guía de usuario específica de la versión 2.</p> <p>Publicación de la documentación únicamente:</p> <ul style="list-style-type: none">• AWS ParallelCluster la versión 2 tiene su propia guía de usuario independiente. | 17 de julio de 2023 |
| AWS ParallelCluster publicada la versión 2.11.9 | <p>AWS ParallelCluster publicada la versión 2.11.9.</p> <p>Correcciones de errores:</p> <ul style="list-style-type: none">• Impida la sustitución de los sistemas de archivos gestionados FSx por Lustre y la pérdida de datos en las actualizaciones de clústeres que incluyan cambios en los mismos. <code>vpc_security_group_id</code> | 2 de diciembre de 2022 |

Para obtener más información sobre los cambios, consulte el CHANGELOG archivo del paquete [aws-parallelcluster](#) en. GitHub

[AWS ParallelCluster Publicada la versión 2.11.8](#)

AWS ParallelCluster publicada la versión 2.11.8.

14 de noviembre de 2022

Cambios:

- Actualice la MPI biblioteca a Intel a la versión 2021 Update 6 (actualizada desde la versión 2021 Update 4). Para obtener más información, consulte la [actualización 6 de 2021 de la MPI Biblioteca Intel®](#).
- Actualice el EFA instalador a la versión 1.19.0
 - Efa-driver: efa-1.16.0-1
 - Efa-config: efa-config-1.11-1 (desde efa-config-1.9-1)
 - Efa-profile: efa-profile-1.5-1 (sin cambios)
 - Libfabric-aws: libfabric-aws-1.16.0-1 (desde libfabric-1.13.2)
 - Rdma-core: rdma-core-41.0-2 (desde rdma-core-37.0)
 - AbrirMPI: openmpi40-aws-4.1.4-3 (desde openmpi40-aws-4.1.1-2

- Actualice el tiempo de ejecución de Python, que utilizan las funciones de Lambda en la AWS Batch integración, a python3.9.

Correcciones de errores:

- Evita que las etiquetas de clúster se cambien durante una actualización porque no es compatible.

[Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del paquete aws-parallelcluster en. GitHub](#)

[AWS ParallelCluster publicada la versión 2.11.7](#)

AWS ParallelCluster publicada la versión 2.11.7.

13 de mayo de 2022

Cambios:

- Actualiza Slurm a la versión 20.11.9

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del paquete [aws-parallelcluster](#) en. GitHub

[AWS ParallelCluster publicada la versión 2.11.6](#)

AWS ParallelCluster publicada la versión 2.11.6. 19 de abril de 2022

Mejoras:

- Mejore la gestión de excepciones en caso de que falte una red.

Cambios

- Actualizaciones del paquete del sistema operativo y correcciones de seguridad.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del paquete [aws-parallelcluster](#) en. GitHub

[AWS ParallelCluster publicada la versión 2.11.5](#)

AWS ParallelCluster publicada la versión 2.11.5. 1 de marzo de 2022

Mejoras:

- Agregue soporte para NEW_CHANGED_DELETE como valor de la opción FSx AutoImportPolicy Lustre.
- Elimine el soporte para los SGE planificadores de par.
- Deshabilita el servicio log4j-cve-2021-44228-hotpatch en Amazon Linux para evitar una posible degradación del rendimiento.

Cambios

- Actualice NVIDIA el controlador a la versión 470.103.01 (desde 470.82.01).
- Actualice NVIDIA Fabric Manager a la versión 470.103.01 (desde 470.82.01).
- Actualice CUDA la biblioteca a la versión 11.4.4 (desde 11.4.3).
- [Intel MPI](#) se actualizó a la versión 2021 Update

4 (actualizada desde la versión 2019 Update 8).

Para obtener más información, consulte la [actualización 4 de 2021 de la MPI Biblioteca Intel®](#).

- Amplíe el tiempo de espera para la creación del nodo principal a una hora.

Correcciones de errores:

- Corrija DCV la conexión a través de los navegadores.
- Se YAML corrigió el uso de comillas para evitar que las etiquetas personalizadas se analizaran como números.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del paquete [aws-parallelcluster](#) en. GitHub

[AWS ParallelCluster publicada la versión 2.11.4](#)

AWS ParallelCluster publicada la versión 2.11.4. 20 de diciembre de 2021

Estos son algunos de los cambios en :

- CentOS Se ha eliminado el soporte 8. CentOS 8 llega al final de su vida útil (EOL) el 31 de diciembre de 2021.
- Upgrade Slurm Workload Manager a la versión 20.11.8.
- Actualice Cinc Client a 17.2.29.
- [Amazon DCV](#) se actualizó a Amazon DCV 2021.2-11190. Para obtener más información, consulta [DCV2021.2-11190, del 11 de octubre de 2021, en la Guía del administrador de Amazon. DCV](#)
- Actualice el NVIDIA controlador a la versión (desde)470.82.01 . 460.73.01
- Actualice CUDA la bibliotec a a la versión 11.4.3 (desde11.3.0).
- Actualice NVIDIA Fabric Manager a470.82.01 .
- Deshabilita la actualización del paquete en el momento del lanzamiento

de la instancia en Amazon Linux 2.

- Desactive la actualización de paquetes desatendida el Ubuntu y Amazon Linux 2.
- Instale la versión Python 3 de los [scripts AWS CloudFormation auxiliares](#) en CentOS 7 y Ubuntu 18.04. (Estos ya se utilizaban en Amazon Linux 2 y Ubuntu 20.04.)

Las correcciones incluyen:

- Deshabilita la actualización del [ec2_iam_role](#) parámetro.
- Corrija la CpuOptions configuración de la plantilla de lanzamiento para T2 instancias.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y los paquetes correspondientes.

[aws-parallelcluster-cookbook](#)

[aws-parallelcluster-node](#)

GitHub

[AWS ParallelCluster publicada la versión 2.11.3](#)

AWS ParallelCluster publicada la versión 2.11.3. 3 de noviembre de 2021

- Soluciona un error [pcluster createami](#) debido a Son of Grid Engine las fuentes no están disponibles en `arc.liv.ac.uk`.

Actualice el instalador [Elastic Fabric Adapter](#) a la versión 1.14.1 (desde la versión 1.13.0)

- EFAconfiguración: `efa-config-1.9-1` (desde `efa-config-1.9`)
- EFAperfil: `efa-profile-1.5-1` (sin cambios)
- EFAMódulo Kernel: `efa-1.14.2` (desde `efa-1.13.0`)
- RDMAnúcleo: `rdma-core-37.0` (desde `rdma-core-35.0`)
- Libfabric: `libfabric-1.13.2` (desde `libfabric-1.13.0`)
- AbiertoMPI: `openmpi40-aws-4.1.1-2` (sin cambios)

GPUDirectRDMA siempre está activado si el tipo de instancia lo admite.

- Los ajustes de configuración [enable_efa_gdr](#) y [enable_efa_gdr](#) no tienen efecto.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y los paquetes correspondientes. [aws-parallelcluster-cookboo](#)
[kaws-parallelcluster-node](#)
GitHub

[AWS ParallelCluster publicada la versión 2.11.2](#)

AWS ParallelCluster publicada la versión 2.11.2. 27 de agosto de 2021

Estos son algunos de los cambios en :

- No lo instale EFA con GPUDirect RDMA (GDR) activado en el momento del arranque si EFA está instalado en la base. AMI
- Bloquee la versión del `nvidia-fabricmanager` paquete para que permanezca sincronizada con la versión del NVIDIA controlador instalada por AWS ParallelCluster.
- Slurm: Se solucionó el problema que se producía cuando el clúster se detenía y reiniciaba mientras se encendía un nodo.
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.13.0:
 - EFAconfiguración: `efa-config-1.9` (sin cambios)
 - EFAperfil: `efa-profile-1.5-1` (sin cambios)
 - EFAMódulo Kernel: `efa-1.13.0` (sin cambios)

- RDMA núcleo: `rdma-core-35.0amzn`
(`derdma-core-32.1amzn`)
- Libfabric: `libfabric-1.13.0amzn1.0`
(desde `libfabric-1.11.2amzn1.1`)
- AbiertoMPI: `openmpi40-aws-4.1.1-2` (sin cambios)
- Cuando se utiliza un paquete personalizado AMI con un EFA paquete preinstalado, no se realizan cambios en el EFA momento del arranque del nodo. Se conserva la implementación EFA del paquete original.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y sus paquetes. [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster publicada la versión 2.11.1](#)

AWS ParallelCluster publicada la versión 2.11.1. 23 de julio de 2021

Estos son algunos de los cambios en :

- Monte los sistemas de archivos mediante la opción de montaje `noatime` para dejar de registrar la hora del último acceso cuando se lee un archivo. Esto mejora el rendimiento del sistema de archivos remoto.
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.12.3:
 - EFAconfiguración: `efa-config-1.9` (desde `efa-config-1.8-1`)
 - EFAperfil: `efa-profile-1.5-1` (sin cambios)
 - EFAMódulo Kernel: `efa-1.13.0` (`deefa-1.12.3`)
 - RDMAnúcleo: `rdma-core-32.1amzn` (sin cambios)
 - Libfabric: `libfabric-1.11.2amzn1.1` (sin cambios)
 - AbiertoMPI: `openmpi40-aws-4.1.1-2` (sin cambios)

- Vuelva a intentar instalar el `aws-parallelcluster` paquete en el nodo principal cuando AWS Batch lo utilice como planificador.
- Evite errores al compilar SGE en un tipo de instancia con más de 31vCPUs.
- Se ha fijado a la versión 1.247347.6 del Amazon CloudWatch Agent para evitar los problemas de la versión 1.247348.0.

[Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del aws-parallelcluster y sus paquetes. aws-parallelcluster-cookbook](#) [GitHub](#)

[AWS ParallelCluster publicada la versión 2.11.0](#)

AWS ParallelCluster publicada la versión 2.11.0. 1 de julio de 2021

Estos son algunos de los cambios en :

- Se agregó soporte para Ubuntu 20.04 (ubuntu2004) y se eliminó el soporte para Ubuntu 16.04 (ubuntu1604) y Amazon Linux (alinux). Amazon Linux 2 (alinux2) sigue siendo totalmente compatible. Para obtener más información, consulte [base_os](#).
- Se ha eliminado la compatibilidad con las versiones de Python anteriores a la 3.6.
- El tamaño de volumen raíz predeterminado se aumentó hasta 35 gibibytes (GiB). Para obtener más información, consulte [compute_root_volume_size](#) y [master_root_volume_size](#).
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.12.2:
 - EFAconfiguración:
efa-config-1.8-1
(desde efa-config-1.7)

- EFAperfil: efa-profile-1.5-1 (deefa-profile-1.4)
- EFAMódulo Kernel: efa-1.12.3 (deefa-1.10.2)
- RDMAnúcleo: rdma-core-32.1amzn (derdma-core-31.2amzn)
- Libfabric: libfabric-1.11.2amzn1.1 (desde libfabric-1.11.1amzn1.0)
- AbrirMPI: openmpi40-aws-4.1.1-2 (desdeopenmpi40-aws-4.1.0)
- Actualizado Slurm a la versión 20.11.7 (desde20.02.7).
- Instale el SSM agente en centos7 ycentos8. (El SSM agente viene preinstalado en alinux2ubuntu1804 , yubuntu2004 .)
- SGE: Utilice siempre el nombre abreviado como filtro de nombre de host. qstat
- Utilice la versión 2 (IMDSv2) del servicio de metadatos de la instancia en lugar de

la versión 1 (IMDSv1) del servicio de metadatos de la instancia para recuperar los metadatos de la instancia. Para obtener más información, consulta [los metadatos de las instancias y los datos de usuario](#) en la Guía del EC2 usuario de Amazon.

- Actualice NVIDIA el controlador a la versión 460.73.01 (desde 450.80.02).
- Actualice CUDA la biblioteca a la versión 11.3.0 (desde 11.0).
- Actualice NVIDIA Fabric Manager a `nvidia-fabricmanager-460` .
- Actualice Python utilizado en AWS ParallelCluster virtualenvs a 3.7.10 (desde 3.6.13).
- Actualice Cinc Client a 16.13.16.
- Actualice las dependencias de terceros de: [aws-parallelcluster-cookbook](#)
 - `apt-7.4.0` (desde `apt-7.3.0`).
 - `iptables-8.0.0` (desde `iptables-7.1.0`).
 - `line-4.0.1` (desde `line-2.9.0`).

- openssh-2.9.1 (desde openssh-2.8.1).
- pyenv-3.4.2 (desde pyenv-3.1.1).
- selinux-3.1.1 (desde selinux-2.1.1).
- ulimit-1.1.1 (desde ulimit-1.0.0).
- yum-6.1.1 (desde yum-5.1.0).
- yum-epel-4.1.2 (desde yum-epel-3.3.0).

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y los paquetes correspondientes. [aws-parallelcluster-cookboo](#)
[kaws-parallelcluster-node](#)
GitHub

[AWS ParallelCluster publicada la versión 2.10.4](#)

AWS ParallelCluster publicada la versión 2.10.4. 15 de mayo de 2021

Estos son algunos de los cambios en :

- Actualizado Slurm a la versión 20.02.7 (desde 20.02.4).

Para obtener más información sobre los cambios, consulte el CHANGELOG archivo del paquete [aws-parallelcluster](#) en. GitHub

[AWS ParallelCluster publicada la versión 2.10.3](#)

AWS ParallelCluster publicada la versión 2.10.3. 18 de marzo de 2021

Estos son algunos de los cambios en :

- Se agregó soporte para Ubuntu 18.04 y Amazon Linux 2 en instancias de AWS Graviton basadas en ARM en China y. AWS GovCloud (US) Regiones de AWS
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.11.2:
 - EFAconfiguración: efa-config-1.7 (sin cambios)
 - EFAperfil: efa-profile-1.4 (de efa-profile-1.3)
 - EFAMódulo Kernel: efa-1.10.2 (sin cambios)
 - RDMA núcleo: rdma-core-31.2amzn (sin cambios)
 - Libfabric: libfabric-1.11.1amzn1.0 (sin cambios)
 - AbiertoMPI: openmpi40-aws-4.1.0 (sin cambios)

Para obtener más información sobre los cambios, consulte el CHANGELOG archivo del paquete [aws-parallelcluster](#) en. GitHub

[AWS ParallelCluster publicada la versión 2.10.2](#)

AWS ParallelCluster publicada la versión 2.10.2. 2 de marzo de 2021

Estos son algunos de los cambios en :

- Mejore la validación de la configuración del clúster para usar el destino del clúster AMI al invocar la EC2 [RunInstances](#) API operación de Amazon en `--dry-run` modo.
- Actualice la versión de Python utilizada en los entornos AWS ParallelCluster virtuales a la 3.6.13.
- Corrija [sanity_check](#) para los tipos de instancias de Arm.
- Se corrige `enable_efa` cuando se usa con centos8 Slurm tipos de instancia de programador o Arm.
- Se ejecuta `apt update` en modo no interactivo (`-y`).
- Fix [encrypted_ephemeral](#) = true con `alinux2` y centos8.

Para obtener más información sobre los cambios, consulte el CHANGELOG archivo del

paquete [aws-parallelcluster](#)
en. GitHub

[AWS ParallelCluster publicada la versión 2.10.1](#)

AWS ParallelCluster publicada la versión 2.10.1. 22 de diciembre de 2020

Estos son algunos de los cambios en :

- Se agregó soporte para África (Ciudad del Cabo) (af-south-1), Europa (Milán) (me-south-1) y Medio Oriente (Bahréin) (me-south-1) Regiones de AWS. En el momento del lanzamiento, el soporte está limitado de las siguientes maneras:
 - FSxpara Lustre y las instancias de Graviton basadas en ARM no son compatibles con ninguna de ellas. Regiones de AWS
 - AWS Batch no es compatible en África (Ciudad del Cabo).
 - Amazon EBS io2 y los tipos de gp3 volumen no son compatibles en África (Ciudad del Cabo) ni en Europa (Milán) Regiones de AWS.
- Se agregó soporte para Amazon EBS io2 y los tipos de gp3 volumen. Para obtener más información, consulte la [sección](#)

de [\[ebs\]](#) y la [sección de \[raid\]](#).

- Se ha agregado compatibilidad para [Elastic Fabric Adapter](#) en las instancias de Graviton2 basadas en ARM que ejecutan `alinux2`, `ubuntu1804` o `ubuntu2004`. Para obtener más información, consulte [Elastic Fabric Adapter](#).
- Instale las bibliotecas de rendimiento de Arm 20.2.1 en Arm AMIs (`alinux2centos8`, `yubuntu1804`). Para obtener más información, consulte [Bibliotecas de rendimiento de Arm](#).
- [Intel MPI](#) se actualizó a la versión 2019 Update 8 (actualizada desde la versión 2019 Update 7). Para obtener más información, consulte la [actualización 8 de 2019 de la MPI Biblioteca Intel®](#).
- Se eliminó la llamada de AWS CloudFormation `DescribeStacks` API operación del punto de entrada de AWS Batch Docker para poner fin a los errores de trabajo causados

por la aceleración. AWS
CloudFormation

- Se han mejorado las llamadas a la llamada de EC2 DescribeInstanceTypes API operación de Amazon al validar la configuración de un clúster.
- Las imágenes de Docker de Amazon Linux 2 se extraen de Amazon ECR Public al crear la imagen de Docker para el awsbatch programador.
- El tipo de instancia predeterminado cambió del tipo de t2.micro instancia codificado al tipo de instancia de nivel gratuito para Región de AWS (t2.microroot3.micro, según). Región de AWS Regiones de AWS las que no tienen una capa gratuita son las opciones predeterminadas para el tipo de t3.micro instancia.
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.11.1:
 - EFAconfiguración:
efa-config-1.7
(desde efa-config-1.5)

- EFAperfil: efa-profile-1.3 (de efa-profile-1.1)
- EFAMódulo Kernel: efa-1.10.2 (sin cambios)
- RDMA núcleo: rdma-core-31.2amzn (de rdma-core-31.amzn0)
- Libfabric: libfabric-1.11.1amzn1.0 (desde libfabric-1.10.1amzn1.1)
- AbrirMPI: openmpi40-aws-4.1.0 (desde openmpi40-aws-4.0.5)
- Los parámetros [vpc_settings](#) , [vpc_id](#) y [master_subnet_id](#) ahora son obligatorios.
- El daemon de nfsd del nodo principal ahora está configurado para usar al menos 8 subprocesos. Si hay más de 8 núcleos, utilizará tantos hilos como núcleos haya. Cuando ubuntu1604 se usa, la configuración solo cambia después de reiniciar el nodo.
- [Amazon DCV](#) se actualizó a Amazon DCV 2020.2-9662.

Para obtener más información, consulta [DCV2020.2-9662, del 4 de diciembre de 2020, en la Guía del administrador de Amazon DCV](#)

- El Intel MPI y HPC los paquetes AWS ParallelCluster correspondientes provienen de Amazon S3. Ya no se extraen de los repositorios de Intel yum.
- Cambió el valor predeterminado `systemd` el nivel de ejecución pasa a `multi-user.target` estar activado OSs durante la creación del oficial AWS ParallelCluster AMIs. El nivel de ejecución se establece `graphical.target` en el nodo principal solo cuando DCV está activado. Esto impide que los servicios gráficos (por ejemplo `x/gdm`) se ejecuten cuando no son necesarios.
- Compatibilidad con instancias de `p4d.24xlarge` en el nodo principal.
- Aumente el número máximo de reintentos al registrarse Slurm nodos en Amazon Route 53.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y los paquetes correspondientes [aws-parallelcluster-cookbook](#). [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster publicada la versión 2.10.0](#)

AWS ParallelCluster publicada la versión 2.10.0. 18 de noviembre de 2020

Estos son algunos de los cambios en :

- Se agregó soporte para CentOS 8 en total Regiones de AWS (fuera de las regiones de AWS China y AWS GovCloud (EE. UU.)). Se ha eliminado el soporte para CentOS 6.
- Se ha agregado compatibilidad para instancias de p4d.24xlarge para los nodos de computación.
- Se agregó compatibilidad con NVIDIA GPUDirect RDMA on EFA mediante la nueva [enable_efa_gdr](#) configuración.
- Se ha añadido compatibilidad con las funciones FSx de Amazon for Lustre.
 - Configura tu sistema de archivos Amazon FSx for Lustre para importar las preferencias mediante el [auto_import_policy](#) ajuste.
- Se ha añadido compatibilidad con los sistemas de archivos HDD basados en Amazon FSx for Lustre que utilizan la

[drive_cache_type](#)
configuración [storage_t](#)
[ype](#) y.

- Se agregó un CloudWatch panel de Amazon, que incluye métricas del nodo principal y fácil acceso a los registros del clúster. Para obtener más información, consulte [CloudWatch Panel de control de Amazon](#).
- Se ha agregado compatibilidad para usar un bucket de Amazon S3 existente para almacenar la información de configuración del clúster mediante la configuración de [cluster_resource_bucket](#).
- Se ha mejorado el comando [pcluster createami](#).
 - Se agregó un `--post-install` parámetro para usar un script posterior a la instalación al crear unAMI.
 - Se agregó un paso de validación que fallaba al usar una base AMI creada por una versión diferente de AWS ParallelCluster.
 - Se ha añadido un paso de validación que puede fallar si el sistema operativo seleccionado

es diferente del sistema operativo baseAMI.

- Se ha añadido soporte para el uso de una AWS ParallelCluster baseAMI.
- Se ha mejorado el comando [pcluster update](#).
 - La configuración [tags](#) ahora se puede cambiar durante una actualización.
 - Ahora se puede cambiar el tamaño de las colas durante una actualización sin detener la flota de computación
- Se ha agregado un parámetro de configuración de `all_or_nothing_batch` para el script `slurm_resume`. Cuando `True`, solo `slurm_resume` tendrá éxito si todas las instancias requeridas por todos los trabajos pendientes están en Slurm estará disponible. Para obtener más información, consulta la sección [Introducción a los all_or_nothing_batch lanzamientos](#) en la AWS ParallelCluster Wiki sobre GitHub.

- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.10.1:
 - EFAconfiguración: efa-config-1.5 (deefa-config-1.4)
 - EFAperfil: efa-profile-1.1 (deefa-profile-1.0.0)
 - EFAMódulo Kernel: efa-1.10.2 (deefa-1.6.0)
 - RDMAnúcleo: rdma-core-31.amzn0 (derdma-core-28.amzn0)
 - Libfabric: libfabric-1.11.1amzn1.0 (desde libfabric-1.10.1amzn1.1)
 - AbrirMPI: openmpi40-aws-4.0.5 (desdeopenmpi40-aws-4.0.3)
- En las AWS GovCloud (US) regiones, habilite el soporte para Amazon DCV y AWS Batch.
- En las regiones de AWS China, habilite la compatibilidad con Amazon FSx for Lustre.
- Actualice NVIDIA el controlador a la versión

450.80.02 (desde la 450.51.05).

- Instale NVIDIA Fabric Manager para activarlo en las plataformas compatibles. NVIDIA NVSwitch
- Se ha eliminado el valor predeterminado Región de AWS deus-east-1 . El valor predeterminado usa este orden de búsqueda.
 - Región de AWS especificado en `-r nuestro --region` argumento.
 - `AWS_DEFAULT_REGION` variable de entorno
 - `aws_region_name` configuración en la [\[aws\]sección](#) del archivo de AWS ParallelCluster configuración (el valor predeterminado es `~/parallelcluster/config`).
 - `region` configuración en la `[default]` sección del archivo de AWS CLI configuración (el valor predeterminado es). `~/aws/config`

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y sus

paquetes. [aws-parallelcluster-cookbook](#)
[aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster publicada la versión 2.9.0](#)

AWS ParallelCluster publicada la versión 2.9.0 11 de septiembre de 2020

Estos son algunos de los cambios en :

- Se agregó soporte para múltiples colas y múltiples tipos de instancias en la flota de cómputo cuando se usa con Slurm Workload Manager. Cuando se utilizan colas, los grupos de escalado automático ya no se utilizan en Slurm. Ahora se crea una zona alojada en Amazon Route 53 con el clúster y se utiliza para la DNS resolución de los nodos de procesamiento cuando el Slurm se utiliza el programador. Para obtener más información, consulte [Modo de Cola múltiple](#).
- Se agregó soporte para [Amazon DCV en instancias](#) basadas en AWS Graviton basadas en ARM.
- Se ha añadido compatibilidad para deshabilitar el hiperproceso en los tipos de instancias que no admiten CPU opciones en las plantillas de lanzamiento (por ejemplo, los tipos de instancias). * .metal

- Se agregó soporte para NFS 4 sistemas de archivos compartidos desde el nodo principal.
- Se ha eliminado la dependencia de [cfn-init](#) al iniciar los nodos de cómputo para evitar la ralentización AWS CloudFormation cuando un gran número de nodos se unen al clúster.
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.9.5:
 - EFAconfiguraciónefa-config-1.3 : (desde) efa-config-1.4
 - EFAperfil: efa-profile-1.0.0 (nuevo)
 - Módulo de kernel: efa-1.6.0 (sin cambios)
 - RDMAnúcleo: rdma-core-28.amzn0 (sin cambios)
 - Libfabric: libfabric-1.10.1amzn1.1 (sin cambios)
 - AbiertoMPI: openmpi40-aws-4.0.3 (sin cambios)
- Actualizado Slurm a la versión 20.02.4 (desde 19.05.5).

- [Amazon DCV](#) se actualizó a Amazon DCV 2020.1-9012. Para obtener más información, consulta las notas de la [versión DCV 2020.1-9012, del 24 de agosto de 2020, en la Guía del administrador de Amazon. DCV](#)
- Al montar NFS unidades compartidas, utilice la dirección IP privada del nodo principal en lugar del nombre de host.
- Se agregaron nuevos flujos de registro a CloudWatch Logs: `chef-client clustermgtd ,computemgtd ,slurm_resume ,yslurm_suspend .`
- Se ha agregado compatibilidad para los nombres de las colas en los scripts previos y posteriores a la instalación.
- En AWS GovCloud (US) Regiones de AWS, utilice la opción de facturación bajo demanda de Amazon DynamoDB. Para obtener más información, consulte [Modo en demanda](#) en la Guía para desarrolladores de Amazon DynamoDB.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y sus paquetes. [aws-parallelcluster-cookbook](#). [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster publicada la versión 2.8.1](#)

AWS ParallelCluster publicada la versión 2.8.1. 4 de agosto de 2020

Estos son algunos de los cambios en :

- Desactiva el bloqueo de pantalla de DCV las sesiones de Amazon para evitar que los usuarios queden bloqueados.
- Se corrige [pcluster configure](#) al incluir un tipo de instancia basada en Graviton de AWS basada en ARM.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos del [aws-parallelcluster](#) y sus paquetes [aws-parallelcluster-cookbook](#). [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster publicada la versión 2.8.0](#)

AWS ParallelCluster publicada la versión 2.8.0 23 de julio de 2020

Estos son algunos de los cambios en :

- Se agregó soporte para instancias basadas en AWS Graviton basadas en ARM (como and). A1 C6g
- Se ha añadido compatibilidad con las funciones de copia de seguridad automática diaria de Amazon FSx for Lustre. Para obtener más información, consulte [automatic_backup_retention_days](#) , [copy_tags_to_backups](#) , [daily_automatic_backup_start_time](#) y [fsx_backup_id](#) .
- Se eliminó la dependencia de Berkshelf desde [pcluster createami](#) .
- Se mejoraron la solidez y la experiencia de usuario de [pcluster update](#). Para obtener más información, consulte [Uso de pcluster update](#).
- Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.9.4:

- Módulo Kernel:
efa-1.6.0 (actualizado desde efa-1.5.1)
- RDMA núcleo: rdma-core-28.amzn0 (actualizado desde rdma-core-25.0)
- Libfabric: libfabric-1.10.1amzn1.1 (actualizado desde libfabric-aws-1.9.0amzn1.1)
- AbiertoMPI: openmpi40-aws-4.0.3 (sin cambios)
- Actualice NVIDIA el controlador a la versión 440.95.01 de Tesla en adelante CentOS 6 y la versión 450.51.05 en todas las demás distribuciones.
- Actualice la CUDA biblioteca a la versión 11.0 en todas las distribuciones que no sean CentOS 6.

Para obtener más información sobre los cambios, consulte los CHANGELOG archivos de [aws-parallelcluster](#) y sus paquetes. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster publicada la versión 2.7.0](#)

AWS ParallelCluster publicada la versión 2.7.0 19 de mayo de 2020

Estos son algunos de los cambios en :

- [base_os](#) ahora es un parámetro obligatorio.
- [scheduler](#) ahora es un parámetro obligatorio.
- [Amazon DCV](#) se actualizó a Amazon DCV 2020.0. Para obtener más información, consulte [Amazon DCV lanza la versión 2020.0 con sonido envolvente 7.1 y compatibilidad con lápiz óptico.](#)

[Intel MPI](#) ha actualizado a la versión 2019 Update 7 (actualizada a partir de la versión 2019 Update 6). Para obtener más información, consulte la [actualización 7 de 2019 de la MPI Biblioteca Intel®](#).

Se ha actualizado el instalador de [Elastic Fabric Adapter](#) a 1.8.4:

- Módulo de kernel: efa-1.5.1 (sin cambios)
- RDMA núcleo: rdma-core-25.0 (sin cambios)

- Libfabric: libfabric-aws-1.9.0amzn1.1 (sin cambios)
- AbrirMPI: openmpi40-aws-4.0.3 (actualizado desde openmpi40-aws-4.0.2)
- Upgrade CentOS 7 AMI a la versión 7.8-2003 (actualizada del 7 de julio a 1908). Para obtener más información, consulte [CentOSNotas de la versión -7 \(2003\)](#).

[AWS ParallelCluster Publicada la versión 2.6.1](#)

AWS ParallelCluster publicada la versión 2.6.1.

17 de abril de 2020

Estos son algunos de los cambios en :

- Eliminado `cfn-init-cmd` y `cfn-wire` de los registros almacenados en Amazon CloudWatch Logs. Para obtener más información, consulte [Integración con Amazon CloudWatch Logs](#).

[AWS ParallelCluster Publicada la versión 2.6.0](#)

AWS ParallelCluster publicada la versión 2.6.0. 27 de febrero de 2020

Estos son algunos de los cambios en :

- Se ha agregado compatibilidad para Amazon Linux 2.
- Ahora Amazon CloudWatch Logs se utiliza para recopilar registros de clústeres y planificadores. Para obtener más información, consulte [Integración con Amazon CloudWatch Logs](#).
- Se agregó compatibilidad con los nuevos tipos de implementación de Amazon FSx for Lustre SCRATCH_2 y PERSISTENT_1 . Support FSx for Lustre en Ubuntu 18.04 y Ubuntu 16.04. Para obtener más información, consulte [fsx](#).
- Se agregó soporte para Amazon DCV en Ubuntu 18.04. Para obtener más información, consulte [Conéctese al nodo principal a través de Amazon DCV](#).

[AWS ParallelCluster publicada la versión 2.5.1](#)

AWS ParallelCluster publicada la versión 2.5.1. 13 de diciembre de 2019

[AWS ParallelCluster publicada la versión 2.5.0](#)

AWS ParallelCluster publicada la versión 2.5.0. 18 de noviembre de 2019

| | | |
|--|--|---------------------|
| AWS ParallelCluster presenta el soporte para Intel MPI | AWS ParallelCluster la versión 2.4.1 introduce la compatibilidad con IntelMPI. | 29 de julio de 2019 |
| AWS ParallelCluster presenta el soporte para EFA | AWS ParallelCluster la versión 2.4.0 introduce la compatibilidad con Elastic Fabric Adapter (EFA). | 11 de junio de 2019 |
| AWS ParallelCluster documentación publicada en el sitio de AWS documentación | La AWS ParallelCluster documentación ahora está disponible en 10 idiomas y en ambos HTML PDF formatos. | 24 de mayo de 2018 |

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.