사용자 가이드

AWS PCS

AWS PCS: 사용자 가이드

Table of Contents

AWS 병렬 컴퓨팅 서비스란?

AWS 병렬 컴퓨팅 서비스 (AWS PCS) 는 Slurm을 AWS 사용하여 고성능 컴퓨팅 (HPC) 워크로드를 더 쉽게 실행 및 확장하고 과학 및 엔지니어링 모델을 구축할 수 있는 관리형 서비스입니다. 동급 최고의 컴퓨팅, 스토리지, 네트워킹 및 시각화를 통합하는 AWS 컴퓨팅 클러스터를 구축하는 데 사용합니다 AWS PCS. 시뮬레이션을 실행하거나 과학 및 엔지니어링 모델을 구축하세요. 내장된 관리 및 관찰 기능을 사용하여 클러스터 운영을 간소화하고 간소화하십시오. 친숙한 환경에서 애플리케이션과 작업을 실행할 수 있도록 지원하여 사용자가 연구와 혁신에 집중할 수 있도록 하세요.

주요 개념

의 AWS PCS 클러스터에는 하나 이상의 컴퓨팅 노드 그룹과 관련된 대기열이 1개 이상 있습니다. 작업은 대기열에 제출되고 컴퓨팅 노드 그룹에 의해 정의된 EC2 인스턴스에서 실행됩니다. 이러한 기반을 사용하여 정교한 아키텍처를 구현할 수 있습니다. HPC

클러스터

클러스터는 리소스를 관리하고 워크로드를 실행하기 위한 리소스입니다. 클러스터는 컴퓨팅, 네트워킹, 스토리지, ID 및 작업 스케줄러 구성의 어셈블리를 정의하는 AWS PCS 리소스입니다. 사용할 작업 스케줄러 (현재 Slurm), 원하는 스케줄러 구성, 클러스터를 관리할 서비스 컨트롤러, 클러스터 리소스를 시작할 위치를 지정하여 VPC 클러스터를 생성합니다. 스케줄러는 작업을 수락하고 스케줄링하며 해당 작업을 처리하는 컴퓨팅 노드 (EC2인스턴스) 도 시작합니다.

컴퓨팅 노드 그룹

컴퓨팅 노드 그룹은 작업을 실행하거나 클러스터에 대한 대화형 액세스를 제공하는 데 AWS PCS 사용하는 컴퓨팅 노드 모음입니다. 컴퓨팅 노드 그룹을 정의할 때 Amazon EC2 인스턴스 유형, 최소 및 최대 인스턴스 수, 대상 VPC 서브넷, Amazon Machine Image (AMI), 구매 옵션, 사용자 지정 시작 구성과 같은 일반적인 특성을 지정합니다. AWS PCS이러한 설정을 사용하여 컴퓨팅 노드 그룹에서 컴퓨팅 노드를 효율적으로 시작, 관리 및 종료합니다.

대기열

특정 클러스터에서 작업을 실행하려면 특정 대기열 (파티션이라고도 함) 에 작업을 제출합니다. 작업은 컴퓨팅 노드 그룹에서 실행되도록 AWS PCS 예약할 때까지 대기열에 남아 있습니다. 하나 이상의 컴퓨팅 노드 그룹을 각 대기열에 연결합니다. 작업 스케줄러가 제공하는 다양한 스케줄링 정책을 사용하여 기본 컴퓨팅 노드 그룹 리소스에서 작업을 예약하고 실행하려면 대기열이 필요합니다. 사용자는 컴퓨팅 노드 또는 컴퓨팅 노드 그룹에 직접 작업을 제출하지 않습니다.

시스템 관리자

시스템 관리자는 클러스터를 배포, 유지 관리 및 운영합니다. 이들은 AWS Management Console AWS PCSAPI, 및 AWS PCS AWS SDK 를 통해 액세스할 수 있습니다. 사용자는 SSH 또는 AWS Systems Manager를 통해 특정 클러스터에 액세스하여 관리 작업을 실행하고, 작업을 실행하고, 데이터를 관리하고, 기타 셸 기반 활동을 수행할 수 있습니다. 자세한 내용은 AWS Systems Manager 설명서를 참조하세요.

최종 사용자

최종 사용자는 클러스터를 배포하거나 운영할 day-to-day 책임이 없습니다. 이들은 터미널 인터페이스(예:SSH) 를 사용하여 클러스터 리소스에 액세스하고, 작업을 실행하고, 데이터를 관리하고, 기타 셸 기반 활동을 수행합니다.

AWS 병렬 컴퓨팅 서비스 설정

AWS 병렬 컴퓨팅 서비스 (AWS PCS) 를 설정하려면 다음 작업을 완료하십시오.

주제

등록하여 AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/등록 열기.
2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 가입하면 AWS 계정 루트 사용자팬이 생성됩니다. AWS 계정루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제라도 https://aws.amazon.com/로 이동하여 내 계정을 선택하여 현재 계정 활동을 확인하고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

가입한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오. AWS 계정 루트 사용자

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 AWS Management Console소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 User Guide의 루트 사용자로 로그인을 참조하십시오.

2. 루트 사용자에 대한 다단계 인증 (MFA) 을 켜십시오.

   지침은 사용 설명서의 AWS 계정 IAM루트 사용자 (콘솔) 용 가상 MFA 기기 활성화를 참조하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAMID 센터를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.

2. IAMID 센터에서 사용자에게 관리 액세스 권한을 부여하십시오.

   를 ID 소스로 사용하는 방법에 대한 자습서는 사용 설명서의 기본값으로 IAM Identity Center 디렉터리AWS IAM Identity Center 사용자 액세스 구성을 참조하십시오. IAM Identity Center 디렉터리

관리 액세스 권한이 있는 사용자로 로그인

- IAMIdentity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 URL 로그인을 사용하십시오.

  IAMIdentity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인을 참조하십시오.AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAMIdentity Center에서 최소 권한 권한 적용의 모범 사례를 따르는 권한 집합을 생성하십시오.

   지침은AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

설치 AWS CLI

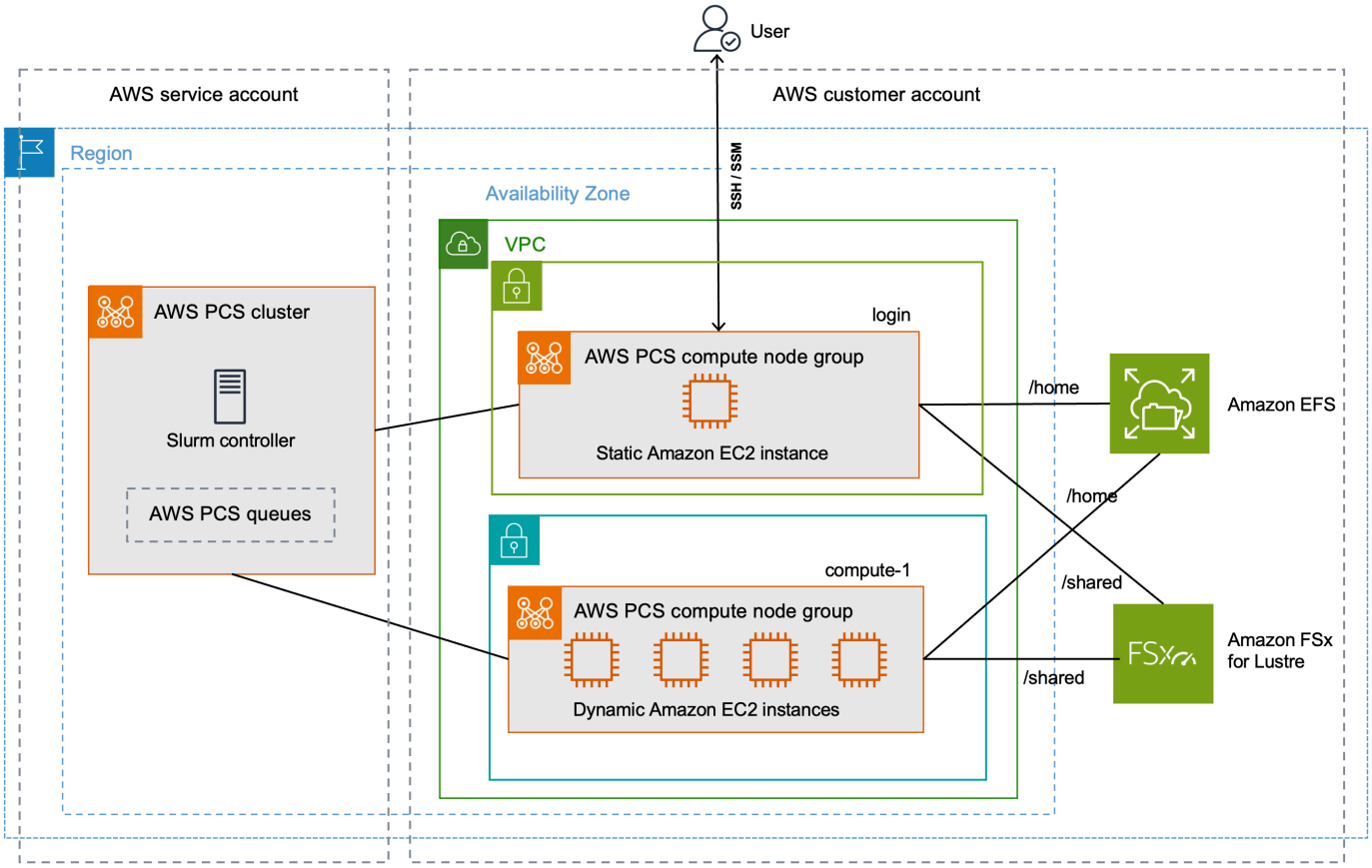
의 최신 버전을 사용해야 합니다 AWS CLI. 자세한 내용은 버전 2용AWS Command Line Interface 사용 설명서의 최신 버전 설치 또는 업데이트를 참조하십시오. AWS CLI

명령 프롬프트에 다음 명령을 입력하여 확인하십시오 AWS CLI. 도움말 정보가 표시될 것입니다.

```
aws pcs help
```

시작하기 AWS PCS

이 자습서는 사용해 볼 수 있는 간단한 클러스터를 만드는 방법에 대한 AWS PCS 자습서입니다. 다음 그림은 클러스터의 설계를 보여줍니다.



자습서 클러스터 설계에는 다음과 같은 주요 구성 요소가 있습니다.

- AWS PCS네트워킹 요구 사항을 충족하는 A VPC 및 서브넷.
- 공유 홈 디렉토리로 사용될 Amazon EFS 파일 시스템.
- Amazon FSx for Lustre 파일 시스템으로, 고성능 공유 디렉터리를 제공합니다.
- Slurm 컨트롤러를 제공하는 AWS PCS 클러스터.
- 컴퓨팅 노드 그룹 2개.
  - 시스템에 대한 셸 기반 대화형 액세스를 제공하는 `login` 노드 그룹.
  - `compute-1`노드 그룹은 작업을 실행할 수 있는 탄력적으로 확장되는 인스턴스를 제공합니다.
- 노드 그룹 내 EC2 인스턴스로 작업을 전송하는 대기열 1개. `compute-1`

클러스터에는 다이어그램에 표시되지 않은 보안 그룹, IAM 역할, EC2 시작 템플릿 등의 추가 AWS 리소스가 필요합니다.

주제

시작하기 위한 사전 요구 사항 AWS PCS

이 자습서를 시작하기 전에 클러스터를 생성하고 관리하는 데 필요한 다음 도구와 리소스를 설치하고 구성하십시오. AWS PCS

- AWS CLI— 다음을 포함한 AWS 서비스 작업을 위한 명령줄 도구입니다 AWS PCS. 자세한 내용은 버전 2용AWS Command Line Interface 사용 설명서의 최신 버전 설치 또는 업데이트를 참조하십시오. AWS CLI를 설치한 AWS CLI후에는 함께 구성하는 것이 좋습니다. 자세한 내용은 버전 2용AWS Command Line Interface 사용 설명서의 AWS CLI 구성을 참조하십시오.
- 필수 IAM 권한 - 사용 중인 IAM 보안 주체에는 AWS PCS IAM 역할, 서비스 연결 역할 AWS CloudFormation VPC, a 및 관련 리소스를 사용할 수 있는 권한이 있어야 합니다. 자세한 내용은 및 AWS Identity and Access Management 사용 설명서의 서비스 연결 역할 만들기를 참조하십시오AWS 병렬 컴퓨팅을 위한 Identity 및 Access Management 서비스. 이 가이드의 모든 단계를 동일한 사용자로 완료해야 합니다. 현재 사용자를 확인하려면 다음 명령을 실행합니다.

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

- 이 항목의 명령줄 단계는 Bash 셸에서 완료하는 것이 좋습니다. Bash 셸을 사용하지 않는 경우 줄 연속 문자 및 변수 설정 및 사용 방식과 같은 일부 스크립트 명령을 통해 셸이 조정되어야 합니다. 또한 쉘의 인용 및 이스케이프 규칙이 다를 수 있습니다. 자세한 내용은 버전 2 AWS Command Line Interface 사용 설명서의 AWS CLI 문자열 포함 따옴표 및 리터럴을 참조하십시오.

에 대한 a VPC 및 서브넷 생성 AWS PCS

템플릿으로 VPC A와 서브넷을 만들 수 있습니다 CloudFormation . 다음을 URL 사용하여 CloudFormation 템플릿을 다운로드한 다음 AWS CloudFormation 콘솔에 템플릿을 업로드하여 새 CloudFormation 스택을 생성합니다. 자세한 내용은 사용 AWS CloudFormation 설명서의 AWS CloudFormation 콘솔 사용을 참조하십시오.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/net/hpc_large_scale/assets/
main.yaml
```

AWS CloudFormation 콘솔에서 템플릿을 연 상태에서 다음 옵션을 입력합니다. 템플릿에 제공된 기본값을 사용할 수 있습니다.

- 스택 이름 제공에서:
  - 스택 이름에 다음을 입력합니다.

    ```
    hpc-networking
    ```
- 매개변수에서:
  - 미만 VPC:
    - 아래에 CidrBlock다음을 입력합니다.

      ```
      10.3.0.0/16
      ```
  - 서브넷 A에서:
    - CidrPublicSubnetA에 다음을 입력합니다.

      ```
      10.3.0.0/20
      ```
    - CidrPrivateSubnetA에 다음을 입력합니다.

      ```
      10.3.128.0/20
      ```

- 서브넷 B에서:
  - CidrPublicSubnetB에 다음을 입력합니다.

```
10.3.16.0/20
```

  - CidrPrivateSubnetB에 다음을 입력합니다.

```
10.3.144.0/20
```

- 서브넷 C에서:
  - ProvisionSubnetsC의 경우 True를 선택합니다.
  - CidrPublicSubnetC에 다음을 입력합니다.

```
10.3.32.0/20
```

  - CidrPrivateSubnetC에 다음을 입력합니다.

```
10.3.160.0/20
```

- 기능에서:
  - IAM리소스가 AWS CloudFormation 생성될 수 있음을 인정한다는 확인란에 체크하세요.

CloudFormation 스택 상태를 모니터링하세요. `CREATE_COMPLETE`도달하면 새 보안 그룹에서 기본 보안 그룹의 ID를 찾으십시오VPC. 이 ID는 자습서 뒷부분에서 사용합니다.

클러스터의 기본 보안 그룹을 찾으십시오. VPC

새 보안 그룹의 ID를 VPC 찾으려면 다음 절차를 따르십시오.

- Amazon VPC 콘솔로 이동합니다.
- VPC대시보드에서 필터링 기준을 선택합니다VPC.
  - 이름이 시작되는 VPC 위치를 선택합니다`hpc-networking`.
  - 보안에서 보안 그룹을 선택합니다.
- 이름이 지정된 그룹의 보안 그룹 ID를 찾으십시오`default`. 설명이 있습니다`default VPC security group`. 나중에 이 ID를 사용하여 EC2 시작 템플릿을 구성할 수 있습니다.

에 대한 보안 그룹 생성 AWS PCS

AWS PCS보안 그룹을 사용하여 클러스터 및 컴퓨팅 노드 그룹으로 들어오고 나가는 네트워크 트래픽을 관리합니다. 이 주제에 대한 자세한 내용은 을 참조하십시오보안 그룹 요구 사항 및 고려 사항.

이 단계에서는 두 보안 그룹에 대한 CloudFormation 템플릿을 사용합니다.

- AWS PCS컨트롤러, 컴퓨팅 노드 및 로그인 노드 간의 통신을 지원하는 클러스터 보안 그룹입니다.
- 액세스를 지원하기 SSH 위해 로그인 노드에 선택적으로 추가할 수 있는 인바운드 SSH 보안 그룹

에 대한 보안 그룹을 생성합니다. AWS PCS

이 CloudFormation 템플릿으로 A VPC 및 서브넷을 생성할 수 있습니다. 다음을 URL 사용하여 CloudFormation 템플릿을 다운로드한 다음 AWS CloudFormation 콘솔에 템플릿을 업로드하여 새 CloudFormation 스택을 생성합니다. 자세한 내용은 사용 AWS CloudFormation 설명서의 AWS CloudFormation 콘솔 사용을 참조하십시오.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-cluster-sg.yaml
```

AWS CloudFormation 콘솔에서 템플릿을 연 상태에서 다음 옵션을 입력합니다. 일부 옵션은 템플릿에 미리 채워지므로 기본값으로 두기만 하면 됩니다.

- 스택 이름 제공에서
  - 스택 이름에 다음을 입력합니다.

    ```
    getstarted-sg
    ```
- 파라미터에서
  - 아래에서 VpcId이름이 시작되는 VPC 위치를 선택합니다`hpc-networking`.
  - (선택 사항) 에서 ClientIpCidr인바운드 SSH 보안 그룹에 대해 더 제한적인 IP 범위를 입력합니다. 자체 IP/서브넷 (자체 IP의 경우 x.x.x/32, 범위의 경우 x.x.x.x/24) 으로 이를 제한하는 것이 좋습니다. x.x.xx를 자신의 IP로 바꾸십시오. PUBLIC https://ifconfig.co/ 같은 도구를 사용하여 퍼블릭 IP를 얻을 수 있습니다.)

CloudFormation 스택 상태를 모니터링하세요. 보안 그룹에 `CREATE_COMPLETE` 도달하면 리소스가 준비된 것입니다.

다음과 같은 이름을 가진 두 개의 보안 그룹이 생성됩니다.

- `cluster-getstarted-sg`— 클러스터 보안 그룹입니다.
- `inbound-ssh-getstarted-sg`— 인바운드 SSH 액세스를 허용하는 보안 그룹입니다.

에서 클러스터 생성 AWS PCS

에서 AWS PCS 클러스터는 리소스를 관리하고 워크로드를 실행하기 위한 영구 리소스입니다. 새 서브넷 또는 기존 서브넷에 특정 스케줄러 (AWS PCS현재 Slurm 지원) 용 클러스터를 생성합니다. VPC 클러스터는 작업을 수락하고 일정을 잡으며 해당 작업을 처리하는 컴퓨팅 노드 (EC2인스턴스) 도 시작합니다.

클러스터 생성

1. AWS PCS콘솔을 열고 클러스터 생성을 선택합니다.
2. 클러스터 설정 섹션에 다음 필드를 입력합니다.
   - 클러스터 이름 — Enter `get-started`
   - 컨트롤러 크기 — 소형 선택
3. 네트워킹 섹션에서 다음 필드의 값을 선택합니다.
   - VPC— VPC 이름을 선택합니다. `hpc-networking:Large-Scale-HPC`
   - 서브넷 - 이름이 시작되는 서브넷을 선택합니다. `hpc-networking:PrivateSubnetA`
   - 보안 그룹 - 이름이 지정된 클러스터 보안 그룹을 선택합니다. `cluster-getstarted-sg`
4. 클러스터 생성을 선택합니다.

> **Note**
>
> 상태 필드에는 클러스터가 프로비저닝되는 동안 만들기가 표시됩니다. 클러스터를 생성하는 데 몇 분이 걸릴 수 있습니다.

Amazon Elastic File AWS PCS System에서 사용할 공유 스토리지 생성

Amazon Elastic File System (AmazonEFS) 은 스토리지 용량 및 성능을 프로비저닝하거나 관리하지 않고도 파일 데이터를 공유할 수 있도록 완전히 탄력적인 서버리스 파일 스토리지를 제공하는 AWS 서비스입니다. 자세한 내용은 Amazon Elastic File System 사용 설명서에서 Amazon Elastic File System 이란?을 참조하세요.

AWS PCS데모 클러스터는 EFS 파일 시스템을 사용하여 클러스터 노드 간에 공유 홈 디렉터리를 제공합니다. 클러스터와 VPC 동일한 위치에 EFS 파일 시스템을 생성합니다.

Amazon EFS 파일 시스템을 만들려면

1. Amazon EFS 콘솔로 이동합니다.
2. 시도하려는 AWS 리전 위치와 동일하게 설정되어 있는지 확인하십시오 AWS PCS.
3. 파일 시스템 생성을 선택합니다.
4. 파일 시스템 생성 페이지에서 다음 매개변수를 설정합니다.
   - 이름에 `getstarted-efs`을 입력합니다.
   - 가상 사설 클라우드 (VPC) 에서 VPC 이름을 선택합니다. `hpc-networking:Large-Scale-HPC`
   - 생성(Create)을 선택합니다. 그러면 파일 시스템 페이지로 돌아갑니다.
5. 파일 시스템의 파일 시스템 ID를 기록해 `getstarted-efs` 둡니다. 나중에 이 정보가 필요합니다.

Amazon for AWS PCS FSx Lustre용 공유 스토리지 생성

Amazon FSx for Lustre를 사용하면 널리 사용되는 고성능 Lustre 파일 시스템을 쉽고 비용 효율적으로 시작하고 실행할 수 있습니다. 기계 학습, 고성능 컴퓨팅 (HPC), 비디오 처리, 금융 모델링과 같이 속도가 중요한 워크로드에는 Lustre를 사용합니다. 자세한 내용은 Amazon FSx for Lustre란 무엇입니까? 를 참조하십시오. Amazon FSx for Lustre 사용 설명서에서 확인할 수 있습니다.

AWS PCS데모 클러스터는 FSx for Lustre 파일 시스템을 사용하여 클러스터 노드 간에 고성능 공유 디렉터리를 제공할 수 있습니다. 클러스터와 동일한 VPC 위치에 FSx Lustre용 파일 시스템을 생성합니다.

FSxLustre용 파일 시스템을 만들려면

1. Amazon FSx 콘솔로 이동합니다.
2. 콘솔이 클러스터와 AWS 리전 동일한 것을 사용하도록 설정되어 있는지 확인하십시오.
3. 파일 시스템 생성을 선택합니다.
   - 파일 시스템 유형 선택에서 Amazon FSx for Lustre를 선택하고 다음을 선택합니다.
4. 파일 시스템 세부 정보 지정 페이지에서 다음 매개 변수를 설정합니다.
   - 파일 시스템 세부 정보에서
     - 이름에 `getstarted-fsx`을 입력합니다.
     - 배포 및 스토리지 유형에서 퍼시스턴트를 선택하고 SSD
     - 스토리지 단위당 처리량에서 125MB/s/tiB를 선택합니다.
     - 스토리지 용량에는 1.2TiB를 입력합니다.
     - 메타데이터 구성에서 자동을 선택합니다.
     - 데이터 압축 유형에서 다음을 선택합니다. LZ4
   - 네트워크 및 보안에서
     - 가상 사설 클라우드 (VPC) 의 경우 VPC 이름을 선택합니다. `hpc-networking:Large-Scale-HPC`
     - VPC보안 그룹의 경우 보안 그룹을 이름을 그대로 두십시오. `default`
     - 서브넷의 경우, 이름이 시작하는 서브넷을 선택합니다. `hpc-networking:PrivateSubnetA`
   - 다른 옵션은 기본값으로 설정된 상태로 둡니다.
   - Next(다음)를 선택합니다.
5. 검토 및 생성 페이지에서 파일 시스템 생성을 선택합니다. 그러면 파일 시스템 페이지로 돌아갑니다.
6. 생성한 Lustre용 파일 시스템의 세부 정보 페이지로 이동합니다. FSx
7. 파일 시스템 ID와 마운트 이름을 기록해 둡니다. 나중에 이 정보가 필요합니다.

> **Note**
>
> 상태 필드에는 파일 시스템이 프로비저닝되는 동안 생성 중이라는 내용이 표시됩니다. 파일 시스템 생성에는 몇 분이 걸릴 수 있습니다. 튜토리얼의 나머지 부분을 진행하기 전에 완료될 때까지 기다리십시오.

에서 컴퓨팅 노드 그룹 생성 AWS PCS

컴퓨팅 노드 그룹은 AWS PCS 시작 및 관리하는 컴퓨팅 노드 (EC2인스턴스) 의 가상 컬렉션입니다. 컴퓨팅 노드 그룹을 정의할 때는 EC2 인스턴스 유형, 최소 및 최대 인스턴스 수, 대상 VPC 서브넷, 선호 구매 옵션, 사용자 지정 시작 구성과 같은 일반적인 특성을 지정합니다. AWS PCS이러한 설정에 따라 컴퓨팅 노드 그룹의 컴퓨팅 노드를 효율적으로 시작, 관리 및 종료합니다. 데모 클러스터는 컴퓨팅 노드 그룹을 사용하여 사용자 액세스를 위한 로그인 노드를 제공하고 별도의 컴퓨팅 노드 그룹을 사용하여 작업을 처리합니다. 다음 항목에서는 클러스터에서 이러한 컴퓨팅 노드 그룹을 설정하는 절차를 설명합니다.

주제

- 에 대한 인스턴스 프로필 생성 AWS PCS
- 에 대한 시작 템플릿 생성 AWS PCS
- 로그인 노드에 대한 컴퓨팅 노드 그룹 생성 AWS PCS
- 에서 컴퓨팅 작업을 실행하기 위한 컴퓨팅 노드 그룹 생성 AWS PCS

에 대한 인스턴스 프로필 생성 AWS PCS

Compute 노드 그룹을 생성할 때는 인스턴스 프로필이 필요합니다. 를 사용하여 EC2 Amazon의 AWS Management Console 역할을 생성하는 경우 콘솔은 자동으로 인스턴스 프로필을 생성하고 역할과 동일한 이름을 지정합니다. 자세한 내용은 사용 AWS Identity and Access Management 설명서의 인스턴스 프로필 사용을 참조하십시오.

다음 절차에서는 를 사용하여 Amazon의 AWS Management Console 역할을 생성하고EC2, 이 역할을 통해 컴퓨팅 노드 그룹에 대한 인스턴스 프로필도 생성합니다.

역할 및 인스턴스 프로필을 만들려면

- IAM 콘솔로 이동합니다.
- 액세스 관리(Access management)에서 정책(Policies)을 선택합니다.

- [Create policy]를 선택합니다.
- 권한 지정에서 정책 편집기의 경우 을 선택합니다 JSON.
- 텍스트 편집기의 내용을 다음과 같이 바꿉니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "pcs:RegisterComputeNodeGroupInstance"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

- Next(다음)를 선택합니다.
- 검토 및 생성에서 정책 이름에 를 입력합니다`AWSPCS-getstarted-policy`.
- 정책 생성을 선택합니다.

- 액세스 관리에서 역할을 선택합니다.
- 역할 생성을 선택합니다.
- 신뢰할 수 있는 엔티티 선택에서:
  - 신뢰할 수 있는 엔티티 유형에서AWS 서비스를 선택합니다.
  - 사용 사례에서 을 선택합니다 EC2.
    - 그런 다음 지정된 서비스의 사용 사례 선택에서 을 선택합니다 EC2.
  - Next(다음)를 선택합니다.
- 권한 추가에서:
  - 권한 정책에서 AWSPCS-getstarted-policy를 검색합니다.
  - AWSPCS-getstarted-policy 옆의 확인란을 선택하여 역할에 추가합니다.
  - 권한 정책에서 A를 검색합니다. mazonSSMManaged InstanceCore
  - A mazonSSMManaged InstanceCore 옆의 체크박스를 선택하여 역할에 추가합니다.
  - Next(다음)를 선택합니다.
- 이름 아래에서 다음을 검토하고 생성하십시오.
  - 역할 세부 정보에서:

  - [역할 이름(Role name)]에 `AWSPCS-getstarted-role`을 입력합니다.
- 역할 생성을 선택합니다.

에 대한 시작 템플릿 생성 AWS PCS

컴퓨팅 노드 그룹을 생성할 때는 해당 그룹이 EC2 시작하는 EC2 인스턴스를 구성하는 데 AWS PCS 사용하는 시작 템플릿을 제공합니다. 여기에는 인스턴스 시작 시 실행되는 보안 그룹 및 스크립트와 같은 설정이 포함됩니다.

이 단계에서는 하나의 CloudFormation 템플릿을 사용하여 두 개의 EC2 시작 템플릿을 생성합니다. 한 템플릿은 로그인 노드를 생성하는 데 사용되고 다른 템플릿은 컴퓨팅 노드를 생성하는 데 사용됩니다. 두 노드 간의 주요 차이점은 인바운드 SSH 액세스를 허용하도록 로그인 노드를 구성할 수 있다는 것입니다.

템플릿에 CloudFormation 액세스

다음을 URL 사용하여 CloudFormation 템플릿을 다운로드한 다음 AWS CloudFormation 콘솔에 템플릿을 업로드하여 새 CloudFormation 스택을 생성합니다. 자세한 내용은 사용 AWS CloudFormation 설명서의 AWS CloudFormation 콘솔 사용을 참조하십시오.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-lt-efs-fsxl.yaml
```

템플릿을 사용하여 EC2 시작 CloudFormation 템플릿을 생성합니다.

다음 절차를 사용하여 AWS CloudFormation 콘솔에서 CloudFormation 템플릿을 완성하십시오.

- 스택 이름 제공에서:
  - 스택 이름에 을 입력합니다`getstarted-lt`.
- 매개변수에서:
  - 보안 상태
    - `default`에서 클러스터에 이름이 지정된 보안 그룹을 선택합니다VPC. VpcSecurityGroupId
    - 에서 ClusterSecurityGroupId이름이 지정된 그룹을 선택합니다. `cluster-getstarted-sg`
    - 이름이 지정된 그룹을 선택합니다. SshSecurityGroupIdinbound`-ssh-getstarted-sg`
    - 에서 SshKeyName원하는 SSH 키 페어를 선택합니다.
  - 파일 시스템에서

  - 에 EfsFilesystemId자습서 앞부분에서 만든 EFS 파일 시스템의 파일 시스템 ID를 입력합니다.
  - 의 경우 FSxLustreFilesystemId자습서 앞부분에서 만든 FSx Lustre용 파일 시스템의 파일 시스템 ID를 입력합니다.
  - 의 FSxLustreFilesystemMountName경우 Lustre 파일 시스템에도 동일한 FSx 마운트 이름을 입력합니다.
- [다음] 을 선택한 후 다시 [다음] 을 선택합니다.
- 제출을 선택합니다.

CloudFormation 스택 상태를 모니터링합니다. 시작 `CREATE_COMPLETE` 템플릿에 도달하면 사용할 준비가 된 것입니다.

> **Note**
>
> CloudFormation 템플릿에서 생성한 모든 리소스를 보려면 AWS CloudFormation 콘솔을 여십시오. `getstarted-lt` 스택을 선택한 다음 리소스(Resources) 탭을 선택합니다.

로그인 노드에 대한 컴퓨팅 노드 그룹 생성 AWS PCS

컴퓨팅 노드 그룹은 AWS PCS 시작 및 관리하는 컴퓨팅 노드 (EC2인스턴스) 의 가상 컬렉션입니다. 컴퓨팅 노드 그룹을 정의할 때는 EC2 인스턴스 유형, 최소 및 최대 인스턴스 수, 대상 VPC 서브넷, 선호 구매 옵션, 사용자 지정 시작 구성과 같은 일반적인 특성을 지정합니다. AWS PCS이러한 설정에 따라 컴퓨팅 노드 그룹의 컴퓨팅 노드를 효율적으로 시작, 관리 및 종료합니다.

이 단계에서는 클러스터에 대한 대화형 액세스를 제공하는 정적 컴퓨팅 노드 그룹을 시작합니다. SSH 또는 Amazon EC2 Systems Manager (SSM) 를 사용하여 로그인한 다음 셸 명령을 실행하고 Slurm 작업을 관리할 수 있습니다.

컴퓨팅 노드 그룹을 생성하려면

- AWS PCS콘솔을 열고 클러스터로 이동합니다.
- 이름이 지정된 클러스터를 선택합니다. `get-started`
- Compute 노드 그룹으로 이동하여 [Create] 를 선택합니다.
- Compute 노드 그룹 설정 섹션에서 다음을 제공합니다.
  - 컴퓨팅 노드 그룹 이름 — `Enterlogin`.
- 컴퓨팅 구성에서 다음 값을 입력하거나 선택합니다.

  - EC2시작 템플릿 - 이름이 있는 시작 템플릿을 선택합니다. `login-getstarted-lt`
  - IAM인스턴스 프로필 - 이름이 지정된 인스턴스 프로필을 선택합니다. `AWSPCS-getstarted-role`
  - 서브넷 - 이름이 시작되는 서브넷을 선택합니다. `hpc-networking:PublicSubnetA`
  - 인스턴스 — 선택. `c6i.xlarge`
  - 구성 조정 — 최소 인스턴스 수에는 을 입력합니다**1**. 최대 인스턴스 수에는 을 입력합니다. `1`
- 추가 설정에서 다음을 지정합니다.
  - AMIID - 이름이 시작되는 AMI 위치를 선택합니다. `aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`
- 컴퓨팅 노드 그룹 생성을 선택합니다.

상태 필드에는 컴퓨팅 노드 그룹이 프로비저닝되는 동안 생성 중이라는 내용이 표시됩니다. 자습서가 진행 중인 동안 자습서의 다음 단계로 진행할 수 있습니다.

에서 컴퓨팅 작업을 실행하기 위한 컴퓨팅 노드 그룹 생성 AWS PCS

이 단계에서는 클러스터에 제출된 작업을 실행하도록 탄력적으로 확장되는 컴퓨팅 노드 그룹을 시작합니다.

컴퓨팅 노드 그룹을 만들려면

- AWS PCS콘솔을 열고 클러스터로 이동합니다.
- 이름이 지정된 클러스터를 선택합니다. `get-started`
- Compute 노드 그룹으로 이동하여 [Create] 를 선택합니다.
- Compute 노드 그룹 설정 섹션에서 다음을 제공합니다.
  - 컴퓨팅 노드 그룹 이름 — Entercompute-1.
- 컴퓨팅 구성에서 다음 값을 입력하거나 선택합니다.
  - EC2시작 템플릿 - 이름이 있는 시작 템플릿을 선택합니다. `compute-getstarted-lt`
  - IAM인스턴스 프로필 - 이름이 지정된 인스턴스 프로필을 선택합니다. `AWSPCS-getstarted-role`
  - 서브넷 - 이름이 시작되는 서브넷을 선택합니다. `hpc-networking:PrivateSubnetA`
  - 인스턴스 — 선택. `c6i.xlarge`
  - 구성 조정 — 최소 인스턴스 수에는 을 입력합니다**0**. 최대 인스턴스 수에는 을 입력합니다. `4`
- 추가 설정에서 다음을 지정합니다.

  - AMIID - 이름이 시작되는 AMI 위치를 선택합니다`aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`.
- 컴퓨팅 노드 그룹 생성을 선택합니다.

상태 필드에는 컴퓨팅 노드 그룹이 프로비저닝되는 동안 생성 중이라는 내용이 표시됩니다.

> **Important**
> 이 자습서의 다음 단계로 진행하기 전에 상태 필드에 Active가 표시될 때까지 기다리십시오.

에서 작업을 관리할 대기열을 생성합니다. AWS PCS

작업을 실행하려면 대기열에 작업을 제출합니다. 작업은 컴퓨팅 노드 그룹에서 실행되도록 AWS PCS 예약할 때까지 대기열에 남아 있습니다. 각 대기열은 처리를 수행하는 데 필요한 EC2 인스턴스를 제공하는 하나 이상의 컴퓨팅 노드 그룹과 연결되어 있습니다.

이 단계에서는 컴퓨팅 노드 그룹을 사용하여 작업을 처리하는 대기열을 생성합니다.

대기열을 생성하려면

- AWS PCS콘솔을 엽니다.
- 이름이 `get-started`인 클러스터를 선택합니다.
- Compute 노드 그룹으로 이동하여 `compute-1` 그룹 상태가 활성인지 확인합니다.

> **Important**
> 다음 단계로 진행하기 전에 `compute-1` 그룹 상태가 활성이어야 합니다.

- 대기열로 이동한 다음 대기열 생성을 선택합니다.
  - 대기열 구성 섹션에서 다음 값을 입력합니다.
    - 대기열 이름 - 다음을 입력합니다. `demo`
    - 컴퓨팅 노드 그룹 - 이름이 지정된 컴퓨팅 노드 그룹을 선택합니다`compute-1`.
- 대기열 생성을 선택합니다.

상태 필드에는 대기열을 생성하는 동안 생성 중이라는 내용이 표시됩니다.

> **Important**
>
> 이 자습서의 다음 단계로 진행하기 전에 상태 필드에 Active가 표시될 때까지 기다리십시오.

AWS PCS클러스터에 연결

`login`컴퓨팅 노드 그룹의 상태가 Active가 되면 생성한 EC2 인스턴스에 연결할 수 있습니다.

로그인 노드에 연결하려면

- AWS PCS콘솔을 열고 클러스터로 이동합니다.
- 이름이 `get-started`인 클러스터를 선택합니다.
- Compute 노드 그룹을 선택합니다.
- 이름이 지정된 컴퓨팅 노드 그룹으로 이동합니다`login`.
- 컴퓨트 노드 그룹 ID를 찾으십시오.
- 다른 브라우저 창이나 탭에서 Amazon EC2 콘솔을 엽니다.
  - 인스턴스를 선택합니다.
  - 다음 태그가 있는 EC2 인스턴스를 검색하십시오. Replace *`node-group-id`* 이전 단계의 Compute 노드 그룹 ID 값을 사용합니다. 인스턴스는 1개여야 합니다.

    ```
    aws:pcs:compute-node-group-id=node-group-id
    ```

  - EC2인스턴스에 연결합니다. 세션 관리자 또는 을 사용할 수 SSH 있습니다.

    Session Manager

    - 인스턴스를 선택합니다.
    - 연결을 선택합니다.
    - [인스턴스에 연결] 에서 [세션 관리자] 를 선택합니다.
    - 연결을 선택합니다.
    - 연결을 선택합니다. 브라우저에서 대화형 터미널이 실행됩니다.

    SSH

    - 인스턴스를 선택합니다.
    - 연결을 선택합니다.
    - [인스턴스에 연결] 에서 [SSH클라이언트] 를 선택합니다.

- 콘솔에서 제공하는 지침을 따르십시오.

> **Note**
>
> 인스턴스의 사용자 이름은 **ec2-user**아닙니다root.

에서 클러스터 환경 살펴보기 AWS PCS

클러스터에 로그인한 후 셸 명령을 실행할 수 있습니다. 예를 들어 사용자를 변경하고, 공유 파일 시스템의 데이터로 작업하고, Slurm과 상호 작용할 수 있습니다.

사용자 변경

세션 관리자를 사용하여 클러스터에 로그인한 경우 로 연결되었을 수 있습니다ssm-user. 이 사용자는 세션 관리자용으로 생성된 특수 사용자입니다. 다음 명령을 사용하여 Amazon Linux 2에서 기본 사용자로 전환합니다. 를 사용하여 연결한 경우에는 이 작업을 수행할 필요가 없습니다SSH.

```
sudo su - ec2-user
```

공유 파일 시스템으로 작업하세요

명령을 사용하여 EFS 파일 시스템 및 Lustre 파일 시스템을 FSx 사용할 수 있는지 확인할 수 있습니다. df -h 클러스터의 출력은 다음과 비슷해야 합니다.

```
[ec2-user@ip-10-3-6-103 ~]$ df -h
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   3.8G     0  3.8G   0% /dev
tmpfs                      3.9G     0  3.9G   0% /dev/shm
tmpfs                      3.9G  556K  3.9G   1% /run
tmpfs                      3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/nvme0n1p1              24G   18G  6.6G  73% /
127.0.0.1:/                8.0E     0  8.0E   0% /home
10.3.132.79@tcp:/zlshxbev  1.2T  7.5M  1.2T   1% /shared
tmpfs                      780M     0  780M   0% /run/user/0
tmpfs                      780M     0  780M   0% /run/user/1000
```

/home파일 시스템은 127.0.0.1을 마운트하며 용량이 매우 큽니다. 이 파일 시스템은 자습서 EFS 앞부분에서 만든 파일 시스템입니다. 여기에 작성된 모든 파일은 클러스터의 모든 /home 노드에서 사용할 수 있습니다.

/shared파일 시스템은 사설 IP를 마운트하며 용량은 1.2TB입니다. 이 파일은 자습서 FSx 앞부분에서 만든 Lustre 파일 시스템용 파일 시스템입니다. 여기에 작성된 모든 파일은 클러스터의 모든 /shared 노드에서 사용할 수 있습니다.

Slurm과 상호작용하세요

주제

대기열 및 노드 목록

대기열과 해당 대기열이 사용하는 노드를 나열할 수 있습니다. sinfo 클러스터의 출력은 다음과 비슷해야 합니다.

```
[ec2-user@ip-10-3-6-103 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
demo         up   infinite      4  idle~ compute-1-[1-4]
[ec2-user@ip-10-3-6-103 ~]$
```

이름이 지정된 demo 파티션을 기록해 두십시오. 상태는 up 이며 최대 4개의 노드가 있습니다. compute-1노드 그룹의 노드와 연결됩니다. 컴퓨팅 노드 그룹을 편집하고 최대 인스턴스 수를 8개로 늘리면 노드 수가 읽고 8 노드 목록이 읽힙니다compute-1-[1-8]. 노드 test 4개로 명명된 두 번째 컴퓨팅 노드 그룹을 생성하여 demo 대기열에 추가하면 해당 노드도 노드 목록에 표시됩니다.

작업 보기

를 사용하여 시스템의 모든 작업을 어떤 상태로든 나열할 수 squeue 있습니다. 클러스터의 출력은 다음과 비슷해야 합니다.

```
[ec2-user@ip-10-3-6-103 ~]$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

나중에 Slurm 작업이 보류 중이거나 실행 중일 때 squeue 다시 실행해 보십시오.

에서 단일 노드 작업 실행 AWS PCS

Slurm을 사용하여 작업을 실행하려면 작업 요구 사항을 지정하는 제출 스크립트를 준비하고 명령을 사용하여 큐에 제출합니다sbatch. 일반적으로 이 작업은 공유 디렉터리에서 수행되므로 로그인 노드와 컴퓨팅 노드에 파일에 액세스할 수 있는 공통 공간이 있습니다.

클러스터의 로그인 노드에 연결하고 셸 프롬프트에서 다음 명령을 실행합니다.

- 기본 사용자가 되십시오. 공유 디렉터리로 변경합니다.

```
sudo su - ec2-user
cd /shared
```

- 다음 명령을 사용하여 예제 작업 스크립트를 만들 수 있습니다.

```
cat << EOF > job.sh
#!/bin/bash
#SBATCH -J single
#SBATCH -o single.%j.out
#SBATCH -e single.%j.err

echo "This is job \${SLURM_JOB_NAME} [\${SLURM_JOB_ID}] running on \
${SLURMD_NODENAME}, submitted from \${SLURM_SUBMIT_HOST}" && sleep 60 && echo "Job
 complete"
EOF
```

- Slurm 스케줄러에 작업 스크립트를 제출하십시오.

```
sbatch -p demo job.sh
```

- 작업이 제출되면 작업 ID가 숫자로 반환됩니다. 이 ID를 사용하여 작업 상태를 확인할 수 있습니다. Replace *job-id* 에서 반환된 번호를 사용하여 다음 명령에서 입력합니다sbatch.

```
squeue --job job-id
```

Example

```
squeue --job 1
```

이 squeue 명령은 다음과 비슷한 출력을 반환합니다.

```
JOBID PARTITION NAME USER     ST TIME NODES NODELIST(REASON)
1     demo      test ec2-user CF 0:47 1     compute-1
```

- 작업이 R (실행 중) 상태에 도달할 때까지 작업 상태를 계속 확인합니다. 아무 것도 squeue 반환되지 않으면 작업이 완료됩니다.
- /shared디렉토리의 내용을 검사하십시오.

```
ls -alth /shared
```

명령 출력은 다음과 비슷합니다.

```
-rw-rw-r- 1 ec2-user ec2-user 107 Mar 19 18:33 single.1.out
-rw-rw-r- 1 ec2-user ec2-user 0 Mar 19 18:32 single.1.err
-rw-rw-r- 1 ec2-user ec2-user 381 Mar 19 18:29 job.sh
```

이름이 지정된 single.1.out 파일은 클러스터의 컴퓨팅 노드 중 하나에서 single.1.err 작성되었습니다. 작업은 공유 디렉터리 (/shared) 에서 실행되었으므로 로그인 노드에서도 사용할 수 있습니다. 이것이 이 FSx 클러스터에 대해 Lustre용 파일 시스템을 구성한 이유입니다.

- 파일 내용을 검사하십시오. single.1.out

```
cat /shared/single.1.out
```

출력은 다음과 유사합니다.

```
This is job test [1] running on compute-1, submitted from ip-10-3-13-181
Job complete
```

Slurm을 사용하여 다중 노드 MPI 작업 실행 AWS PCS

이 지침은 Slurm을 사용하여 에서 메시지 전달 인터페이스 () 작업을 실행하는 방법을 보여줍니다. MPI AWS PCS

로그인 노드의 셸 프롬프트에서 다음 명령을 실행합니다.

- 기본 사용자가 되세요. 홈 디렉터리로 변경합니다.

```
sudo su - ec2-user
```

```
cd ~/
```

- C 프로그래밍 언어로 소스 코드를 생성합니다.

```
cat > hello.c << EOF
// * mpi-hello-world - https://www.mpitutorial.com
// Released under MIT License
//
// Copyright (c) 2014 MPI Tutorial.
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to
// deal in the Software without restriction, including without limitation the
// rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
// sell copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.

#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();
}
EOF
```

- Open MPI 모듈을 로드합니다.

```
module load openmpi
```

- C 프로그램을 컴파일합니다.

```
mpicc -o hello hello.c
```

- Slurm 작업 제출 스크립트를 작성하세요.

```
cat > hello.sh << EOF
#!/bin/bash
#SBATCH -J multi
#SBATCH -o multi.out
#SBATCH -e multi.err
#SBATCH --exclusive
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1

srun $HOME/hello
EOF
```

- 공유 디렉터리로 변경합니다.

```
cd /shared
```

- 작업 스크립트를 제출합니다.

```
sbatch -p demo ~/hello.sh
```

- 작업이 완료될 때까지 작업을 모니터링하는 squeue 데 사용합니다.
- multi.out다음 내용을 확인하세요.

```
cat multi.out
```

출력 결과는 다음과 비슷합니다. 참고로 각 랭크는 다른 노드에서 실행되었으므로 고유한 IP 주소를 가집니다.

```
Hello world from processor ip-10-3-133-204, rank 0 out of 4 processors
Hello world from processor ip-10-3-128-219, rank 2 out of 4 processors
Hello world from processor ip-10-3-141-26, rank 3 out of 4 processors
Hello world from processor ip-10-3-143-52, rank 1 out of 4 processor
```

에 대한 AWS 리소스 삭제 AWS PCS

이 자습서에서 생성한 클러스터 및 노드 그룹을 모두 사용한 후에는 생성한 리소스를 삭제해야 합니다.

> **Important**
> 내에서 실행 중인 모든 리소스에 대해 청구 요금이 청구됩니다. AWS 계정

이 튜토리얼을 위해 생성한 AWS PCS 리소스를 삭제하려면

- AWS PCS콘솔을 엽니다.
- get-started라는 이름의 클러스터로 이동합니다.
- 대기열 섹션으로 이동합니다.
- 이름이 demo인 대기열을 선택합니다.
- Delete(삭제)를 선택합니다.

> **Important**
> 대기열이 삭제될 때까지 기다린 다음 진행하십시오.

- Compute 노드 그룹 섹션으로 이동합니다.
- compute-1이라는 컴퓨팅 노드 그룹을 선택합니다.
- Delete(삭제)를 선택합니다.
- login이라는 컴퓨팅 노드 그룹을 선택합니다.
- Delete(삭제)를 선택합니다.

> **Important**
> 계속하려면 두 컴퓨팅 노드 그룹이 모두 삭제될 때까지 기다리십시오.

- 시작을 위한 클러스터 세부 정보 페이지에서 삭제를 선택합니다.

> **Important**
> 클러스터가 삭제될 때까지 기다린 다음 단계를 진행하십시오.

이 자습서를 위해 만든 다른 AWS 리소스를 삭제하려면

- IAM콘솔을 엽니다.
  - 역할을 선택합니다.
  - 이름이 AWSPCS-getstarted-role인 역할을 선택한 다음 삭제를 선택합니다.
  - 역할을 삭제한 후 정책을 선택합니다.
  - 이름이 AWSPCS-getstarted-policy인 정책을 선택한 다음 삭제를 선택합니다.
- AWS CloudFormation 콘솔을 엽니다.
  - 이름이 getstarted-lt인 스택을 선택합니다.
  - Delete(삭제)를 선택합니다.

> **Important**
> 스택이 삭제될 때까지 기다린 후 진행하십시오.

- Amazon EFS 콘솔을 엽니다.
  - 파일 시스템을 선택합니다.
  - 이름이 getstarted-efs인 파일 시스템을 선택합니다.

- Delete(삭제)를 선택합니다.

> **Important**
>
> 계속하려면 파일 시스템이 삭제될 때까지 기다리십시오.

- Amazon FSx 콘솔을 엽니다.
  - 파일 시스템을 선택합니다.
  - 이름이 getstarted-fsx인 파일 시스템을 선택합니다.
  - Delete(삭제)를 선택합니다.

> **Important**
>
> 계속하기 전에 파일 시스템이 삭제될 때까지 기다리십시오.

- AWS CloudFormation 콘솔을 엽니다.
  - 이름이 getstarted-sg인 스택을 선택합니다.
  - Delete(삭제)를 선택합니다.
- AWS CloudFormation 콘솔을 엽니다.
  - hpc-네트워킹이라는 이름의 스택을 선택합니다.
  - Delete(삭제)를 선택합니다.

함께 일하기 AWS PCS

이 장에서는 사용에 도움이 되는 정보와 지침을 제공합니다 AWS PCS.

주제

AWS PCS클러스터

AWS PCS클러스터는 다음 구성 요소로 구성됩니다.

- Slurm control 데몬 () 과 같은 HPC 시스템 스케줄러 소프트웨어의 관리형 인스턴스. `slurmctld`
- Amazon EC2 인스턴스를 프로비저닝하고 관리하기 위해 HPC 시스템 스케줄러와 통합되는 구성 요소.
- HPC시스템 스케줄러와 통합되어 CloudWatch Amazon으로 로그와 지표를 전송하는 구성 요소.

이러한 구성 요소는 에서 관리하는 AWS계정에서 실행됩니다. 이들은 함께 작동하여 고객 계정의 Amazon EC2 인스턴스를 관리합니다. AWS PCSAmazon VPC 서브넷에 엘라스틱 네트워크 인터페이스를 프로비저닝하여 스케줄러 소프트웨어에서 Amazon EC2 인스턴스로의 연결을 제공합니다 (예: 인스턴스에 대한 배치 작업 스케줄링을 지원하고 사용자가 스케줄러 명령을 실행하여 해당 작업을 나열하고 관리할 수 있도록 지원).

주제

- AWS PCS클러스터 크기 선택
- 에서 클러스터 시크릿 사용하기 AWS PCS

AWS 병렬 컴퓨팅 서비스에서 클러스터 생성

이 항목에서는 사용 가능한 옵션에 대한 개요를 제공하고 AWS 병렬 컴퓨팅 서비스 (AWS PCS) 에서 클러스터를 생성할 때 고려해야 할 사항에 대해 설명합니다. AWS PCS클러스터를 처음 생성하는 경우 다음과 같이 하는 것이 좋습니다시작하기 AWS PCS. 자습서를 통해 사용 가능한 모든 옵션과 HPC 시스템 아키텍처를 확장하지 않고도 제대로 작동하는 시스템을 만들 수 있습니다.

사전 조건

- 요구 사항을 충족하는 AWS PCS네트워킹 기존 VPC 및 서브넷. 프로덕션 용도로 클러스터를 배포하기 전에 서브넷 VPC 및 서브넷 요구 사항을 철저히 이해하는 것이 좋습니다. AND VPC 서브넷을 만들려면 을 참조하십시오. VPC AWS PCS클러스터용 생성
- AWS PCS리소스를 만들고 관리할 권한이 있는 IAM주도자. 자세한 내용은 AWS 병렬 컴퓨팅을 위한 Identity 및 Access Management 서비스 단원을 참조하십시오.

AWS PCS클러스터 생성

AWS Management Console AWS CLI OR를 사용하여 클러스터를 만들 수 있습니다.

AWS Management Console

클러스터 생성

1. https://console.aws.amazon.com/pcs/home#/clusters에서 AWS PCS 콘솔을 열고 클러스터 생성을 선택합니다.
2. 클러스터 설정 섹션에 다음 필드를 입력합니다.
   - 클러스터 이름 - 클러스터의 이름입니다. 이름에는 영숫자(대소문자 구분)와 하이픈만 사용할 수 있습니다. 영문자로 시작해야 하며 40자를 초과할 수 없습니다. 이름은 클러스터를 만드는 데 사용하는 AWS 리전 밴드 AWS 계정 내에서 고유해야 합니다.
   - 스케줄러 - 스케줄러와 버전을 선택합니다. AWS PCS현재 Slurm 23.11을 지원합니다. 자세한 내용은 슬럼 버전 입력 AWS PCS 단원을 참조하십시오.
   - 컨트롤러 크기 — 컨트롤러 크기를 선택합니다. 이에 따라 AWS PCS 클러스터에서 관리할 수 있는 동시 작업 및 컴퓨팅 노드 수가 결정됩니다. 클러스터를 생성할 때만 컨트롤러 크기

를 설정할 수 있습니다. 크기 조정에 대한 자세한 내용은 을 참조하십시오AWS PCS클러스터 크기 선택.

3. 네트워킹 섹션에서 다음 필드의 값을 선택합니다.

   - VPC— AWS PCS 요구 사항을 VPC 충족하는 기존 제품을 선택합니다. 자세한 내용은 AWS PCSVPC및 서브넷 요구 사항 및 고려 사항 단원을 참조하십시오. 클러스터를 생성한 후에는 클러스터를 변경할 수 없습니다VPC. 목록에 VPCs 없는 경우 먼저 만들어야 합니다.
   - 서브넷 - 선택한 서브넷에서 사용 가능한 모든 서브넷이 VPC 나열됩니다. 서로 다른 가용 영역에서 두 개를 선택합니다. 각 서브넷은 서브넷 요구 사항을 AWS PCS 충족해야 합니다. 자세한 내용은 AWS PCSVPC및 서브넷 요구 사항 및 고려 사항 단원을 참조하십시오. 스케줄러 엔드포인트가 퍼블릭 인터넷에 노출되지 않도록 하려면 프라이빗 서브넷을 선택하는 것이 좋습니다.
   - 보안 그룹 - 클러스터용으로 생성한 네트워크 인터페이스에 AWS PCS 연결할 보안 그룹을 지정합니다. 클러스터와 해당 컴퓨팅 노드 간의 통신을 허용하는 보안 그룹을 하나 이상 선택해야 합니다. 자세한 내용은 보안 그룹 요구 사항 및 고려 사항 단원을 참조하십시오.

4. (선택 사항) 암호화에서 다음 필드를 설정하여 컨트롤러 데이터를 암호화하는 사용자 지정 키를 정의할 수 있습니다.

   - KMS키 ID — PCS 생성한 KMS 키는 그대로 `aws/pcs` 사용하십시오. 사용자 지정 KMS KMS 키를 사용하려면 기존 키 별칭을 선택합니다. 클러스터를 생성하는 데 사용된 계정에는 사용자 지정 KMS 키에 대한 `kms:Decrypt` 권한이 있어야 한다는 점에 유의하십시오.

5. (선택 사항) Slurm 구성 섹션에서 다음과 같이 설정된 기본값을 재정의하는 Slurm 구성 옵션을 지정할 수 있습니다. AWS PCS

   - 유휴 시간 축소 — 동적으로 프로비저닝된 컴퓨팅 노드에 배치된 작업이 완료되거나 종료된 후 활성 상태를 유지하는 시간을 제어합니다. 이 값을 더 길게 설정하면 노드에서 후속 작업을 실행할 가능성이 높아지지만 비용이 증가할 수 있습니다. 값이 짧을수록 비용은 줄어들지만 HPC 시스템에서 작업을 실행하는 데 걸리는 시간과 비교하여 노드를 프로비저닝하는 데 소비하는 시간의 비율이 증가할 수 있습니다.
   - Prolog - 컴퓨팅 노드 그룹 인스턴스의 프롤로그 스크립트 디렉터리로 연결되는 완전한 경로입니다. 이는 Slurm의 Prolog 설정에 해당합니다. 참고로 이 디렉토리는 특정 실행 파일의 경로가 아니라 디렉토리여야 합니다.
   - 에필로그 — 컴퓨팅 노드 그룹 인스턴스의 에필로그 스크립트 디렉터리에 대한 완전한 경로입니다. 이는 Slurm의 에필로그 설정에 해당합니다. 참고로 이 디렉토리는 특정 실행 파일의 경로가 아니라 디렉토리여야 합니다.

- 유형 매개 변수 선택 — 이는 Slurm에서 사용하는 리소스 선택 알고리즘을 제어하는 데 도움이 됩니다. 이 값을 로 CR_CPU_Memory 설정하면 메모리 인식 스케줄링이 활성화되고, 이 값을 로 CR_CPU 설정하면 스케줄링만 활성화됩니다. CPU 이 파라미터는 by로 설정된 Slurm의 SelectTypeParameters설정에 해당합니다. SelectType select/cons_tres AWS PCS

6. (선택 사항) 태그에서 클러스터에 태그를 추가합니다. AWS PCS
7. 클러스터 생성을 선택합니다. 클러스터를 AWS PCS 생성하는 Creating 동안 상태 필드가 표시됩니다. 이 프로세스는 몇 분 정도 걸릴 수 있습니다.

> **Important**
>
> 한 Creating AWS 리전 상태당 클러스터가 1개만 있을 수 AWS 계정있습니다. AWS PCS 클러스터를 만들려고 할 때 해당 Creating 상태에 이미 클러스터가 있는 경우 오류가 반환됩니다.

AWS CLI

클러스터 생성

1. 다음 명령을 사용하여 클러스터를 생성합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

   - Replace *region* 클러스터를 AWS 리전 생성하려는 ID와 함께 입력합니다 (예:)us-east-1.
   - Replace *my-cluster* 클러스터 이름과 함께 이름에는 영숫자(대소문자 구분)와 하이픈만 사용할 수 있습니다. 알파벳 문자로 시작해야 하며 40자를 초과할 수 없습니다. 이름은 클러스터를 생성하는 AWS 계정 위치 AWS 리전 및 클러스터 내에서 고유해야 합니다.
   - Replace *23.11* 지원되는 모든 Slurm 버전과 함께 사용할 수 있습니다.

     > **Note**
     >
     > AWS PCS현재 슬럼 23.11을 지원합니다.

   - Replace *SMALL* 지원되는 모든 클러스터 크기와 함께 사용할 수 있습니다. 이에 따라 AWS PCS 클러스터에서 관리할 수 있는 동시 작업 및 컴퓨팅 노드 수가 결정됩니다. 클러스터

를 생성할 때만 설정할 수 있습니다. 크기 조정에 대한 자세한 내용은 을 참조하십시오AWS PCS클러스터 크기 선택.

- 의 값을 자신의 값으로 subnetIds 바꾸십시오. 스케줄러 엔드포인트가 퍼블릭 인터넷에 노출되지 않도록 하려면 프라이빗 서브넷을 선택하는 것이 좋습니다.
- 클러스터용으로 securityGroupIds 생성하는 네트워크 AWS PCS 인터페이스에 연결할 항목을 지정하십시오. 보안 그룹은 VPC 클러스터와 동일해야 합니다. 클러스터와 컴퓨팅 노드 간의 통신을 허용하는 보안 그룹을 하나 이상 선택해야 합니다. 자세한 내용은 보안 그룹 요구 사항 및 고려 사항 단원을 참조하십시오.
- 선택적으로 옵션을 추가하여 Slurm 동작을 세밀하게 조정할 수 있습니다. --slurm-configration 예를 들어 를 사용하여 스케일 다운 유휴 시간을 60분 (3600초) 으로 설정할 수 있습니다. --slurm configuration scaleDownIdeTime=3600
- 원하는 경우 를 사용하여 컨트롤러의 데이터를 암호화하는 사용자 지정 KMS 키를 제공할 수 있습니다. --kms-key-id *kms-key* 기존 KMS ARN 키 ID 또는 *kms-key* 별칭으로 바꾸십시오. 단, 클러스터를 생성하는 데 사용된 계정에는 사용자 지정 KMS 키에 대한 kms:Decrypt 권한이 있어야 합니다.

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=23.11 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

2. 클러스터를 프로비저닝하는 데 몇 분 정도 걸릴 수 있습니다. 다음 명령을 사용하여 클러스터의 상태를 쿼리할 수 있습니다. 클러스터의 상태 필드가 표시되기 전까지는 대기열 또는 컴퓨팅 노드 그룹 생성을 진행하지 마십시오. ACTIVE

```
aws pcs get-cluster --region region --cluster-identifier my-cluster
```

> **Important**
>
> 한 Creating 상태에 있는 클러스터는 한 개당 AWS 리전 1개만 있을 수 있습니다. AWS 계정 AWS PCS클러스터를 만들려고 할 때 해당 Creating 상태에 이미 클러스터가 있는 경우 오류가 반환됩니다.

클러스터에 권장되는 다음 단계

- 컴퓨팅 노드 그룹을 추가합니다.
- 대기열을 추가합니다.
- 로깅을 활성화합니다.

에서 클러스터 삭제 AWS PCS

이 항목에서는 AWS PCS 클러스터를 삭제하는 방법에 대한 개요를 제공합니다.

AWS PCS클러스터를 삭제할 때 고려할 사항

- 클러스터를 삭제하려면 먼저 클러스터와 관련된 모든 대기열을 삭제해야 합니다. 자세한 내용은 에서 대기열 삭제 AWS PCS 단원을 참조하십시오.
- 클러스터를 삭제하려면 먼저 클러스터와 연결된 모든 컴퓨팅 노드 그룹을 삭제해야 합니다. 자세한 내용은 에서 컴퓨팅 노드 그룹 삭제 AWS PCS 단원을 참조하십시오.

클러스터를 삭제합니다.

AWS Management Console AWS CLI OR를 사용하여 클러스터를 삭제할 수 있습니다.

AWS Management Console

클러스터를 삭제하려면

1. AWS PCS콘솔을 엽니다.
2. 삭제할 클러스터를 선택합니다.
3. Delete(삭제)를 선택합니다.
4. 클러스터 상태 필드가 표시됩니다`Deleting`. 완료되는 데 몇 분 정도 걸릴 수 있습니다.

AWS CLI

클러스터를 삭제하려면

1. 다음 명령을 사용하여 클러스터를 삭제하십시오. 클러스터는 다음과 같이 대체됩니다.
   - Replace *`region-code`* 클러스터가 AWS 리전 들어 있는 상태입니다.

- Replace *my-cluster* 클러스터 이름 또는 ID와 함께

```
aws pcs delete-cluster --region region-code --cluster-identifier my-cluster
```

2. 클러스터를 삭제하는 데 몇 분 정도 걸릴 수 있습니다. 다음 명령을 사용하여 클러스터의 상태를 확인할 수 있습니다.

```
aws pcs get-cluster --region region-code --cluster-identifier my-cluster
```

AWS PCS클러스터 크기 선택

AWS PCS가용성이 높고 안전한 클러스터를 제공하는 동시에 패치, 노드 프로비저닝, 업데이트와 같은 주요 작업을 자동화합니다.

클러스터를 생성할 때는 다음 두 가지 요소를 기준으로 클러스터 크기를 선택합니다.

- 관리할 컴퓨팅 노드의 수
- 클러스터에서 실행될 것으로 예상되는 활성 및 대기 작업 수

Slurm 클러스터 크기	관리되는 인스턴스 수	활성 및 대기 중인 작업 수
작은	최대 32개	최대 256개
중간	최대 512개	최대 8192까지
라지	최대 2048개	최대 16384까지

예시

- 클러스터에 최대 24개의 관리형 인스턴스가 있고 최대 100개의 작업을 실행하는 경우 스몰을 선택하십시오.
- 클러스터에 최대 24개의 관리형 인스턴스가 있고 최대 1000개의 작업을 실행할 수 있는 경우 [Medium] 을 선택합니다.
- 클러스터에 최대 1000개의 관리형 인스턴스가 있고 최대 100개의 작업을 실행할 경우 Large를 선택합니다.

- 클러스터에 최대 1000개의 관리형 인스턴스가 있고 최대 10,000개의 작업을 실행할 경우 Large를 선택합니다.

에서 클러스터 시크릿 사용하기 AWS PCS

클러스터를 생성하는 과정에서 클러스터의 작업 스케줄러에 연결하는 데 필요한 클러스터 암호를 AWS PCS 생성합니다. 또한 확장 이벤트에 대한 응답으로 시작할 인스턴스 세트를 정의하는 AWS PCS 컴퓨팅 노드 그룹을 생성합니다. AWS PCS해당 컴퓨팅 노드 그룹에서 시작한 인스턴스를 클러스터 암호로 구성하여 작업 스케줄러에 연결할 수 있도록 합니다. Slurm 클라이언트를 수동으로 구성해야 하는 경우가 있을 수 있습니다. 영구 로그인 노드를 구축하거나 작업 관리 기능을 갖춘 워크플로 관리자를 설정하는 경우를 예로 들 수 있습니다.

AWS PCS클러스터 암호를 접두사가 포함된 관리 암호로 저장합니다`pcs!`. AWS Secrets Manager암호 비용은 사용 AWS PCS 요금에 포함됩니다.

> **Warning**
>
> 클러스터 비밀번호를 수정하지 마세요. AWS PCS클러스터 암호를 수정하면 클러스터와 통신할 수 없습니다. AWS PCS클러스터 암호 교체를 지원하지 않습니다. 클러스터 암호를 수정해야 하는 경우 새 클러스터를 생성해야 합니다.

목차

Slurm 클러스터 시크릿 찾기

콘솔을 사용하거나API, AWS Secrets Manager 콘솔에서 AWS PCS 직접 또는 태그를 사용하여 AWS PCS -managed 암호를 찾을 수 있습니다.

클러스터 AWS Secrets Manager 암호를 찾는 데 사용합니다.

AWS Management Console

1. Secrets Manager 콘솔로 이동합니다.

2. Secrets를 선택한 다음 `pcs!` 접두사를 검색합니다.

> **Note**
>
> AWS PCS클러스터 비밀의 이름은 다음과 같습니다. `pcs!slurm-secret-`*`cluster-id`* 여기서 *`cluster-id`* 는 AWS PCS 클러스터 ID입니다.

AWS CLI

각 AWS PCS 클러스터 암호에도 태그가 지정됩니다. `aws:pcs:`*`cluster-id`* 다음 명령을 사용하여 클러스터의 비밀 ID를 가져올 수 있습니다. 명령을 실행하기 전에 다음과 같이 대체하십시오.

- *`region`*로 바꾸어 AWS 리전 클러스터를 생성하십시오 (예:). `us-east-1`
- 클러스터 암호를 찾으려면 AWS PCS 클러스터의 *`cluster-id`* ID로 바꾸십시오.

```
aws secretsmanager list-secrets \
    --region region  \
    --filters Key=tag-key,Values=aws:pcs:cluster-id \
            Key=tag-value,Values=cluster-id
```

클러스터 암호를 찾는 AWS PCS 데 사용합니다.

를 사용하여 AWS PCS 클러스터 암호의 ARN 를 찾을 수 있습니다. AWS CLI 다음 명령을 입력하고 다음과 같이 대체하십시오.

- *`region`*로 바꾸어 AWS 리전 클러스터를 생성하십시오 (예:). `us-east-1`
- 클러스터의 이름 또는 *`my-cluster`* 식별자로 바꾸십시오.

```
aws pcs get-cluster --region region --cluster-identifier my-cluster
```

다음 예제 출력은 `get-cluster` 명령에서 나온 것입니다. `secretArn`및 를 `secretVersion` 함께 사용하여 암호를 가져올 수 있습니다.

```
{
    "cluster": {
```

```
        "name": "pcsdemo",
        "id": "s3431v9rx2",
        "arn": "arn:aws:pcs:us-east-1:012345678901:cluster/s3431v9rx2",
        "status": "ACTIVE",
        "createdAt": "2024-07-12T15:32:27.225136+00:00",
        "modifiedAt": "2024-07-12T15:32:27.225136+00:00",
        "scheduler": {
            "type": "SLURM",
            "version": "23.11"
        },
        "size": "SMALL",
        "networking": {
            "subnetIds": [
                "subnet-0123456789abcdef"
            ],
            "securityGroupIds": [
                "sg-0123456789abcde"
            ]
        },
        "endpoints": [
            {
                "type": "SLURMCTLD",
                "privateIpAddress": "127.0.0.1",
                "port": "6817"
            }
        ],
        "secretArn": "arn:aws:secretsmanager:us-east-1:012345678901:secret:pcs!slurm-
secret-s3431v9rx2-FN7tJF",
        "secretVersion": "ff58d1fd-070e-4bbc-98a0-64ef967cebcc"
    }
}
```

Slurm 클러스터 시크릿 확인하기

Secrets Manager를 사용하여 Slurm 클러스터 암호의 현재 base64로 인코딩된 버전을 가져올 수 있습니다. 다음 예에서는 를 사용합니다. AWS CLI명령을 실행하기 전에 다음과 같이 대체하십시오.

- *region*로 바꾸어 AWS 리전 클러스터를 생성하십시오 (예:). `us-east-1`
- AWS PCS클러스터의 `secretArn` 것으로 *secret-arn* 교체하십시오.

```
aws secretsmanager get-secret-value \
    --region region \
```

```
    --secret-id 'secret-arn' \
    --version-stage AWSCURRENT \
    --query 'SecretString' \
    --output text
```

Slurm 클러스터 시크릿을 사용하는 방법에 대한 자세한 내용은 을 참조하십시오. 독립형 인스턴스를 AWS PCS 로그인 노드로 사용

권한

IAM보안 주체를 사용하여 Slurm 클러스터 암호를 가져옵니다. IAM보안 주체는 암호를 읽을 권한이 있어야 합니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 역할 용어 및 개념을 참조하십시오.

다음 샘플 IAM 정책은 예제 클러스터 암호에 대한 액세스를 허용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSecretValueRetrievalAndVersionListing",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:012345678901:secret:pcs!
slurm-secret-s3431v9rx2-FN7tJF"
        }
    ]
}
```

AWS PCS컴퓨팅 노드 그룹

AWS PCS컴퓨팅 노드 그룹은 노드의 논리적 컬렉션 (Amazon EC2 인스턴스) 입니다. 이러한 노드를 사용하여 컴퓨팅 작업을 실행할 수 있을 뿐만 아니라 시스템에 대한 대화형 셸 기반 액세스를 제공할 수도 HPC 있습니다. 컴퓨팅 노드 그룹은 사용할 Amazon EC2 인스턴스 유형, 실행할 인스턴스 수, 스팟 인스턴스 또는 온디맨드 인스턴스 사용 여부, 사용할 서브넷 및 보안 그룹, 시작 시 각 인스턴스를 구성하는 방법 등 노드 생성 규칙으로 구성됩니다. 이러한 규칙이 AWS PCS 업데이트되면 컴퓨팅 노드 그룹과 관련된 리소스가 일치하도록 업데이트됩니다.

주제

에서 컴퓨팅 노드 그룹 생성 AWS PCS

이 항목에서는 사용 가능한 옵션에 대한 개요를 제공하고 AWS 병렬 컴퓨팅 서비스 (AWS PCS) 에서 컴퓨팅 노드 그룹을 생성할 때 고려해야 할 사항에 대해 설명합니다. 에서 AWS PCS 처음으로 컴퓨팅 노드 그룹을 생성하는 경우 의 자습서를 따르는 것이 좋습니다시작하기 AWS PCS. 이 자습서를 통해 사용 가능한 모든 옵션과 HPC 시스템 아키텍처를 확장하지 않고도 제대로 작동하는 시스템을 만들 수 있습니다.

사전 조건

- 원하는 수의 EC2 인스턴스를 원하는 만큼 시작할 수 있을 만큼 충분한 서비스 할당량 AWS 리전를 AWS Management Console사용하여 서비스 할당량 증가를 확인하고 요청할 수 있습니다.
- 네트워킹 요구 사항을 충족하는 AWS PCS 기존 VPC 및 서브넷 프로덕션 용도로 클러스터를 배포하기 전에 이러한 요구 사항을 완전히 이해하는 것이 좋습니다. 자세한 내용은 AWS PCSVPC및 서브넷 요구 사항 및 고려 사항 단원을 참조하십시오. CloudFormation 템플릿을 사용하여 VPC a와 서브넷을 만들 수도 있습니다. AWS CloudFormation 템플릿의 HPC 레시피를 제공합니다. 자세한 내용은 aws-hpc-recipeson을 참조하십시오 GitHub.
- AWS PCS`RegisterComputeNodeGroupInstance`API작업을 호출하고 노드 그룹 IAM 인스턴스에 필요한 기타 AWS 리소스에 액세스할 수 있는 권한이 있는 인스턴스 프로필입니다. 자세한 내용은 IAM AWS 병렬 컴퓨팅 서비스의 인스턴스 프로필 단원을 참조하십시오.
- 노드 그룹 인스턴스의 시작 템플릿입니다. 자세한 내용은 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS 단원을 참조하십시오.
- Amazon EC2 스팟 인스턴스를 사용하는 컴퓨팅 노드 그룹을 생성하려면 AWSServiceRoleForEC2Spot서비스 연결 역할이 있어야 합니다. AWS 계정자세한 내용은 아마존 EC2 스팟 역할 AWS PCS 단원을 참조하십시오.

에서 컴퓨팅 노드 그룹을 생성하십시오. AWS PCS

AWS Management Console 또는 를 사용하여 컴퓨팅 노드 그룹을 생성할 수 AWS CLI있습니다.

AWS Management Console

콘솔을 사용하여 컴퓨팅 노드 그룹을 만들려면

1. AWS PCS콘솔을 엽니다.
2. 컴퓨팅 노드 그룹을 생성할 클러스터를 선택합니다. Compute 노드 그룹으로 이동하여 [Create] 를 선택합니다.
3. Compute 노드 그룹 설정 섹션에서 노드 그룹의 이름을 입력합니다. 이름에는 대소문자를 구분하는 영숫자 문자와 하이픈만 사용할 수 있습니다. 영문자로 시작해야 하며 25자를 초과할 수 없습니다. 이름은 클러스터 내에서 고유해야 합니다.
4. 컴퓨팅 구성에서 다음 값을 입력하거나 선택합니다.

   a. EC2시작 템플릿 - 이 노드 그룹에 사용할 사용자 지정 시작 템플릿을 선택합니다. 시작 템플릿을 사용하여 서브넷, 보안 그룹, 모니터링 구성, 인스턴스 수준 스토리지와 같은 네트워크 설정을 사용자 지정할 수 있습니다. 시작 템플릿을 준비하지 않은 경우 시작 템플릿을 만드는 방법을 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS 알아보려면 을 참조하십시오.

   > **Important**
   >
   > AWS PCS각 컴퓨팅 노드 그룹에 대한 관리형 시작 템플릿을 생성합니다. 이름이 지정됩니다`pcs-identifier-do-not-delete`. 컴퓨팅 노드 그룹을 생성하거나 업데이트할 때 이러한 항목을 선택하지 마십시오. 그렇지 않으면 노드 그룹이 제대로 작동하지 않습니다.

   b. EC2시작 템플릿 버전 - 사용자 지정 시작 템플릿의 버전을 선택합니다. 재현성을 높일 수 있는 특정 버전을 선택할 수 있습니다. 나중에 버전을 변경하는 경우 컴퓨팅 노드 그룹을 업데이트하여 시작 템플릿의 변경 사항을 감지해야 합니다. 자세한 내용은 AWS PCS컴퓨팅 노드 그룹 업데이트 단원을 참조하십시오.

   c. AMIID — 시작 템플릿에 AMI ID가 포함되어 있지 않거나 시작 템플릿의 값을 재정의하려는 경우 여기에 AMI ID를 입력합니다. 단, 노드 그룹에 AMI 사용되는 것은 호환 AWS PCS 가능해야 합니다. 에서 AMI 제공한 샘플을 선택할 수도 AWS있습니다. 이 주제에 대한 자세한 내용은 을 참조하십시오아마존 머신 이미지 (AMIs) 용 AWS PCS.

   d. IAM인스턴스 프로필 - 노드 그룹의 인스턴스 프로필을 선택합니다. 인스턴스 프로필은 인스턴스에 AWS 리소스와 서비스에 안전하게 액세스할 수 있는 권한을 부여합니다. 아직 준비하지 않은 경우, 생성 방법을 IAM AWS 병렬 컴퓨팅 서비스의 인스턴스 프로필 보려면 을 참조하십시오.

e. 서브넷 - AWS PCS 클러스터가 배포된 VPC 위치에서 하나 이상의 서브넷을 선택합니다. 서브넷을 여러 개 선택하면 노드 간에 EFA 통신이 불가능하고 서로 다른 서브넷에 있는 노드 간 통신 지연 시간이 길어질 수 있습니다. 여기서 지정하는 서브넷이 시작 템플릿에서 정의한 서브넷과 일치하는지 확인하십시오. EC2

f. 인스턴스 - 노드 그룹의 조정 요청을 처리할 인스턴스 유형을 하나 이상 선택합니다. 모든 인스턴스 유형의 프로세서 아키텍처 (x864_64 또는 arm64) 와 개수가 같아야 합니다. vCPUs 인스턴스에 있는 GPUs 경우 모든 인스턴스 유형의 개수가 같아야 합니다. GPUs

g. 조정 구성 - 노드 그룹의 최소 및 최대 인스턴스 수를 지정합니다. 고정된 수의 노드가 실행되는 정적 구성을 정의하거나 최대 노드 수까지 실행할 수 있는 동적 구성을 정의할 수 있습니다. 정적 구성의 경우 최소값과 최대값을 동일하게, 0보다 큰 값으로 설정합니다. 동적 구성의 경우 최소 인스턴스를 0으로 설정하고 최대 인스턴스를 0보다 큰 수로 설정합니다. AWS PCS정적 인스턴스와 동적 인스턴스가 혼합된 컴퓨팅 노드 그룹을 지원하지 않습니다.

5. (선택 사항) 추가 설정에서 다음을 지정합니다.

   a. 구매 옵션 - 스팟 인스턴스와 온디맨드 인스턴스 중에서 선택합니다.

   b. 할당 전략 - 스팟 구매 옵션을 선택한 경우 노드 그룹에서 인스턴스를 시작할 때 스팟 용량 풀을 선택하는 방법을 지정할 수 있습니다. 자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 스팟 인스턴스 할당 전략을 참조하십시오. 온디맨드 구매 옵션을 선택한 경우에는 이 옵션이 적용되지 않습니다.

6. (선택 사항) Slurm사용자 지정 설정 섹션에서 다음 값을 입력합니다.

   a. 가중치 - 이 값은 스케줄링을 위해 그룹 내 노드의 우선 순위를 설정합니다. 가중치가 낮은 노드일수록 우선 순위가 높으며 단위는 임의적입니다. 자세한 내용은 설명서의 가중치를 참조하십시오. Slurm

   b. 실제 메모리 - 이 값은 노드 그룹 내 노드의 실제 메모리 크기 (GB) 를 설정합니다. 이 값은 의 클러스터 Slurm 구성 `CR_CPU_Memory` 옵션과 함께 사용하기 위한 AWS PCS 것입니다. 자세한 내용은 RealMemorySlurm설명서를 참조하십시오.

7. (선택 사항) 태그에서 컴퓨팅 노드 그룹에 태그를 추가합니다.

8. 컴퓨팅 노드 그룹 생성을 선택합니다. 노드 그룹을 `Creating` AWS PCS 프로비저닝하는 동안 상태 필드가 표시됩니다. 몇 분 정도 걸릴 수 있습니다.

권장되는 다음 단계

- 노드 그룹을 대기열에 AWS PCS 추가하여 작업을 처리할 수 있도록 합니다.

AWS CLI

를 사용하여 컴퓨팅 노드 그룹을 생성하려면 AWS CLI

다음 명령을 사용하여 대기열을 생성합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

1. Replace *region*클러스터를 생성할 때 사용할 AWS 리전 ID로 입력합니다 (예:)`us-east-1`.
2. Replace *my-cluster* `clusterId`클러스터의 이름 또는 이름과 함께
3. Replace *my-node-group*컴퓨팅 노드 그룹의 이름과 함께 이름에는 영숫자(대소문자 구분)와 하이픈만 사용할 수 있습니다. 영문자로 시작해야 하며 25자를 초과할 수 없습니다. 이름은 클러스터 내에서 고유해야 합니다.
4. Replace *subnet-ExampleID1* 클러스터에 하나 이상의 IDs 서브넷이 있어야 VPC 합니다.
5. Replace *lt-ExampleID1* 사용자 지정 시작 템플릿의 ID와 함께 아직 준비하지 않은 경우 템플릿을 만드는 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS 방법을 알아보려면 을 참조하십시오.

   > **Important**
   >
   > AWS PCS각 컴퓨팅 노드 그룹에 대한 관리형 시작 템플릿을 생성합니다. 이름이 지정됩니다`pcs-`*identifier*`-do-not-delete`. 컴퓨팅 노드 그룹을 생성하거나 업데이트할 때 이러한 항목을 선택하지 마십시오. 그렇지 않으면 노드 그룹이 제대로 작동하지 않습니다.

6. Replace *launch-template-version* 노드 그룹을 특정 버전과 연결하려는 경우 특정 시작 템플릿 버전을 사용하십시오.
7. Replace *arn:InstanceProfile*IAM인스턴스 ARN 프로필과 함께. 아직 준비하지 않은 경우 지침을 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS 참조하십시오.
8. Replace *min-instances* 그리고 *max-instances* 정수 값 포함. 고정된 수의 노드가 실행되는 정적 구성이나 최대 노드 수까지 실행할 수 있는 동적 구성을 정의할 수 있습니다. 정적 구성의 경우 최소값과 최대값을 동일하게, 0보다 큰 값으로 설정합니다. 동적 구성의 경우 최소 인스턴스를 0으로 설정하고 최대 인스턴스를 0보다 큰 수로 설정합니다. AWS PCS정적 인스턴스와 동적 인스턴스가 혼합된 컴퓨팅 노드 그룹을 지원하지 않습니다.
9. Replace *t3.large* 다른 인스턴스 유형과 함께 사용할 수 있습니다. `instanceType`설정 목록을 지정하여 더 많은 인스턴스 유형을 추가할 수 있습니다. 예: *--instance-configs instanceType=c6i.16xlarge,instanceType=c6a.16xlarge*. 모든 인스턴스 유형의 프로세서 아키텍처 (x864_64 또는 arm64) 와 개수가 같아야 합니다. vCPUs 인스턴스에 있는 GPUs 경우 모든 인스턴스 유형의 개수가 같아야 합니다. GPUs

```
aws pcs create-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-name my-node-group \
    --subnet-ids subnet-ExampleID1 \
    --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
    --iam-instance-profile arn=arn:InstanceProfile \
    --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
    --instance-configs instanceType=t3.large
```

create-compute-node-group명령에 추가할 수 있는 몇 가지 선택적 구성 설정이 있습니다.

- 사용자 지정 시작 템플릿에 에 에 대한 참조가 포함되지 않는지 또는 해당 값을 재정의할지 지정할 --amiId 수 있습니다. AMI 단, 노드 그룹에 AMI 사용되는 것은 호환 AWS PCS 가능해야 합니다. 에서 AMI 제공한 샘플을 선택할 수도 AWS있습니다. 이 주제에 대한 자세한 내용은 을 참조하십시오아마존 머신 이미지 (AMIs) 용 AWS PCS.
- 를 사용하여 --purchase-option on-demand (ONDEMAND) 인스턴스와 Spot (SPOT) 인스턴스 중에서 선택할 수 있습니다. 온디맨드가 기본값입니다. 스팟 인스턴스를 선택하면 노드 그룹에서 인스턴스를 시작할 때 스팟 용량 AWS PCS 풀을 선택하는 방법을 정의하는 --allocation-strategy 데 사용할 수도 있습니다. 자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 스팟 인스턴스 할당 전략을 참조하십시오.
- 를 사용하여 노드 그룹의 노드에 Slurm 구성 옵션을 제공할 수 --slurm-configuration 있습니다. 가중치 (스케줄링 우선 순위) 와 실제 메모리를 설정할 수 있습니다. 가중치가 낮은 노드일수록 우선 순위가 높으며 단위는 임의적입니다. 자세한 내용은 설명서의 가중치를 참조하십시오. Slurm 실제 메모리는 노드 그룹 내 노드의 실제 메모리 크기 (GB) 를 설정합니다. 이는 AWS PCS Slurm 구성의 클러스터 CR_CPU_Memory 옵션과 함께 사용하기 위한 것입니다. 자세한 내용은 RealMemorySlurm설명서를 참조하십시오.

> **Important**
>
> 컴퓨팅 노드 그룹을 생성하는 데 몇 분 정도 걸릴 수 있습니다.

다음 명령을 사용하여 노드 그룹의 상태를 쿼리할 수 있습니다. 상태가 도달할 때까지는 노드 그룹을 대기열에 연결할 수 없습니다ACTIVE.

```
aws pcs get-compute-node-group --region region \
    --cluster-identifier my-cluster \
```

```
    --compute-node-group-identifier my-node-group
```

AWS PCS컴퓨팅 노드 그룹 업데이트

이 항목에서는 사용 가능한 옵션에 대한 개요를 제공하고 AWS PCS 컴퓨팅 노드 그룹을 업데이트할 때 고려해야 할 사항을 설명합니다.

AWSPCS컴퓨팅 노드 그룹 업데이트 옵션

AWSPCS컴퓨팅 노드 그룹을 업데이트하면 에서 시작한 인스턴스의 속성과 해당 인스턴스가 시작되는 방식에 대한 규칙을 변경할 수 있습니다. AWS PCS 예를 들어 노드 AMI 그룹용 인스턴스를 다른 소프트웨어가 설치된 다른 인스턴스로 교체할 수 있습니다. 또는 보안 그룹을 업데이트하여 인바운드 또는 아웃바운드 네트워크 연결을 변경할 수 있습니다. 조정 구성을 변경하거나 스팟 인스턴스로 또는 스팟 인스턴스에서 선호하는 구매 옵션을 변경할 수도 있습니다.

다음 노드 그룹 설정은 생성 후에는 변경할 수 없습니다.

- 명칭
- 인스턴스

AWS PCS컴퓨팅 노드 그룹 업데이트 시 고려 사항

Compute 노드 그룹은 작업 처리, 대화형 셸 액세스 및 기타 작업에 사용되는 EC2 인스턴스를 정의합니다. 이들은 종종 하나 이상의 AWS PCS 대기열과 연결됩니다. 컴퓨팅 노드 그룹을 업데이트하여 해당 동작 (또는 해당 노드의 동작) 을 변경할 때는 다음 사항을 고려하십시오.

- 컴퓨팅 노드 그룹 속성에 대한 변경 사항은 컴퓨팅 노드 그룹 상태가 업데이트에서 활성으로 변경될 때 적용됩니다. 새 인스턴스는 업데이트된 속성과 함께 시작됩니다.
- 특정 노드의 구성에 영향을 주지 않는 업데이트는 실행 중인 노드에 영향을 주지 않습니다. 서브넷 추가 및 할당 전략 변경 등을 예로 들 수 있습니다.
- 컴퓨팅 노드 그룹의 시작 템플릿을 업데이트하는 경우 새 버전을 사용하도록 컴퓨팅 노드 그룹을 업데이트해야 합니다.
- 컴퓨팅 노드 그룹의 노드에서 보안 그룹을 추가하거나 제거하려면 시작 템플릿을 편집하고 컴퓨팅 노드 그룹을 업데이트하십시오. 새 인스턴스는 업데이트된 보안 그룹 세트와 함께 시작됩니다.
- 컴퓨팅 노드 그룹에서 사용하는 보안 그룹을 직접 편집하면 실행 중인 인스턴스와 향후 인스턴스에 즉시 적용됩니다.

- 컴퓨팅 노드 그룹이 사용하는 IAM 인스턴스 프로필에서 권한을 추가하거나 제거하면 실행 중인 인스턴스 및 향후 인스턴스에 즉시 적용됩니다.
- 컴퓨팅 노드 그룹 인스턴스에서 AMI 사용되는 항목을 변경하려면 새 인스턴스를 사용하도록 컴퓨팅 노드 그룹 (또는 시작 템플릿) 을 AMI 업데이트하고 인스턴스가 교체될 때까지 기다리십시오 AWS PCS.
- AWS PCS노드 그룹 업데이트 작업 후 노드 그룹의 기존 인스턴스를 대체합니다. 노드에 실행 중인 작업이 있는 경우 노드를 AWS PCS 교체하기 전에 해당 작업을 완료할 수 있습니다. 대화형 사용자 프로세스 (예: 로그인 노드 인스턴스) 가 종료됩니다. 노드 그룹 상태는 인스턴스를 교체용으로 AWS PCS 표시하는 시점으로 `Active` 돌아가지만 실제 교체는 인스턴스가 유휴 상태일 때 발생합니다.
- 컴퓨팅 노드 그룹에 허용되는 최대 인스턴스 수를 줄이면 Slurm에서 노드가 AWS PCS 제거되어 새 최대값에 도달하게 됩니다. AWS PCS제거된 Slurm 노드와 관련된 실행 중인 인스턴스를 종료합니다. 제거된 노드에서 실행 중인 작업이 실패하고 해당 대기열로 돌아갑니다.
- AWS PCS각 컴퓨팅 노드 그룹에 대한 관리형 시작 템플릿을 생성합니다. 이름이 지정되어 `pcs-`*`identifier`*`-do-not-delete` 있습니다. 컴퓨팅 노드 그룹을 생성하거나 업데이트할 때 이를 선택하지 마십시오. 그렇지 않으면 노드 그룹이 제대로 작동하지 않습니다.
- 구매 옵션에 스팟을 사용하도록 컴퓨팅 노드 그룹을 업데이트하는 경우 계정에 AWSServiceRoleForEC2Spot서비스 연결 역할이 있어야 합니다. 자세한 내용은 아마존 EC2 스팟 역할 AWS PCS 단원을 참조하십시오.

AWSPCS컴퓨팅 노드 그룹을 업데이트하려면

AWS관리 콘솔 또는 를 사용하여 노드 그룹을 업데이트할 수 AWS CLI 있습니다.

AWS Management Console

컴퓨팅 노드 그룹을 업데이트하려면

1. 다음 위치에서 AWS PCS 콘솔을 엽니다. `https://console.aws.amazon.com/pcs/home#/clusters`
2. 컴퓨팅 노드 그룹을 업데이트하려는 클러스터를 선택합니다.
3. Compute 노드 그룹으로 이동하여 업데이트하려는 노드 그룹으로 이동한 다음 편집을 선택합니다.
4. 컴퓨팅 구성, 추가 설정 및 Slurm사용자 지정 설정 섹션에서 다음을 제외한 모든 값을 업데이트합니다.
   - 인스턴스 - 컴퓨팅 노드 그룹의 인스턴스는 변경할 수 없습니다.

5. 업데이트를 선택합니다. 상태 필드에는 변경 사항이 적용되는 동안 업데이트 중인 내용이 표시됩니다.

> **Important**
>
> 컴퓨팅 노드 그룹 업데이트는 몇 분 정도 걸릴 수 있습니다.

AWS CLI

컴퓨팅 노드 그룹을 업데이트하려면

1. 다음 명령으로 컴퓨팅 노드 그룹을 업데이트하십시오. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

   a. Replace *region-code* 클러스터를 생성하려는 AWS 지역과 함께

   b. Replace *my-node-group* 이름과 함께 또는 컴퓨팅 노드 `computeNodeGroupId` 그룹용입니다.

   c. Replace *my-cluster* `clusterId`클러스터의 이름 또는 이름과 함께

```
aws pcs update-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group
```

2. 를 제외한 모든 노드 그룹 매개변수를 `--instance-configs` 업데이트하십시오. 예를 들어, 새 AMI ID를 설정하려면 다음과 같이 `--amiId my-custom-ami-id` 전달하십시오.*my-custom-ami-id* 사용자가 선택한 AMI 것으로 대체됩니다.

> **Important**
>
> 컴퓨팅 노드 그룹을 업데이트하는 데 몇 분 정도 걸릴 수 있습니다.

다음 명령을 사용하여 노드 그룹의 상태를 쿼리할 수 있습니다.

```
aws pcs get-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
```

```
    --compute-node-group-identifier my-node-group
```

에서 컴퓨팅 노드 그룹 삭제 AWS PCS

이 항목에서는 사용 가능한 옵션에 대한 개요를 제공하고 에서 컴퓨팅 노드 그룹을 삭제할 때 고려할 사항을 설명합니다 AWS PCS.

컴퓨팅 노드 그룹 삭제 시 고려 사항

컴퓨팅 노드 그룹은 작업 처리, 대화형 셸 액세스 및 기타 작업에 사용되는 EC2 인스턴스를 정의합니다. 이들은 종종 하나 이상의 AWS PCS 대기열과 연결됩니다. 컴퓨팅 노드 그룹을 삭제하기 전에 다음 사항을 고려하십시오.

- 컴퓨팅 노드 그룹에서 시작한 모든 EC2 인스턴스는 종료됩니다. 이렇게 하면 해당 인스턴스에서 실행 중인 작업이 취소되고 실행 중인 대화형 프로세스가 종료됩니다.
- 컴퓨팅 노드 그룹을 삭제하려면 먼저 모든 대기열에서 연산 노드 그룹을 분리해야 합니다. 자세한 내용은 AWS PCS대기열 업데이트 단원을 참조하십시오.

컴퓨팅 노드 그룹을 삭제합니다.

AWS Management Console 또는 AWS CLI 를 사용하여 컴퓨팅 노드 그룹을 삭제할 수 있습니다.

AWS Management Console

컴퓨팅 노드 그룹을 삭제하려면

1. AWS PCS콘솔을 엽니다.
2. 컴퓨팅 노드 그룹의 클러스터를 선택합니다.
3. Compute 노드 그룹으로 이동하여 삭제할 컴퓨팅 노드 그룹을 선택합니다.
4. Delete(삭제)를 선택합니다.
5. 상태 필드가 표시됩니다`Deleting`. 완료되는 데 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 스케줄러에 기본 제공되는 명령을 사용하여 컴퓨팅 노드 그룹이 삭제되었는지 확인할 수 있습니다. 예를 들어 Slurm에는 `sinfo` 또는 `squeue` 를 사용하십시오.

AWS CLI

컴퓨팅 노드 그룹을 삭제하려면

- 다음 명령을 사용하여 다음과 같이 대체된 컴퓨팅 노드 그룹을 삭제하십시오.
  - Replace *region-code* 클러스터가 AWS 리전 들어 있는 상태입니다.
  - Replace *my-node-group* 컴퓨팅 노드 그룹의 이름 또는 ID와 함께
  - Replace *my-cluster* 클러스터의 이름 또는 ID와 함께.

```
aws pcs delete-compute-node-group --region region-code \
      --compute-node-group-identifier my-node-group \
      --cluster-identifier my-cluster
```

컴퓨팅 노드 그룹을 삭제하는 데 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 스케줄러 고유의 명령을 사용하여 컴퓨팅 노드 그룹이 삭제되었는지 확인할 수 있습니다. 예를 들어 Slurm에는 `sinfo` 또는 `squeue` 를 사용하십시오.

에서 컴퓨팅 노드 그룹 인스턴스 찾기 AWS PCS

각 AWS PCS 컴퓨팅 노드 그룹은 공유 구성으로 EC2 인스턴스를 시작할 수 있습니다. EC2태그를 사용하여 AWS Management Console 또는 와 함께 컴퓨팅 노드 그룹에서 인스턴스를 찾을 수 AWS CLI 있습니다.

AWS Management Console

컴퓨팅 노드 그룹 인스턴스를 찾으려면

1. AWS PCS콘솔을 엽니다.
2. 클러스터를 선택합니다.
3. Compute 노드 그룹을 선택합니다.
4. 생성한 로그인 노드 그룹의 ID를 찾으십시오.
5. EC2콘솔로 이동하여 [Instances] 를 선택합니다.

6. 다음 태그가 있는 인스턴스를 검색합니다. Replace *node-group-id* 컴퓨팅 노드 그룹의 ID (이름 아님) 를 사용하십시오.

```
aws:pcs:compute-node-group-id=node-group-id
```

7. (선택 사항) 검색 필드의 인스턴스 상태 값을 변경하여 구성 중이거나 최근에 종료된 인스턴스를 찾을 수 있습니다.
8. 태그가 지정된 인스턴스 목록에서 각 인스턴스의 인스턴스 ID와 IP 주소를 찾을 수 있습니다.

AWS CLI

노드 그룹 인스턴스를 찾으려면 다음 명령을 사용하십시오. 명령을 실행하기 전에 다음과 같이 바꾸십시오.

- AWS 리전 클러스터의 것으로 *region-code* 교체하십시오. 예제: `us-east-1`
- 컴퓨팅 노드 그룹의 ID (이름 아님) *node-group-id* 로 바꾸십시오.
- 다른 상태의 인스턴스를 `terminated` 찾으려면 `pending` 또는 와 같은 다른 EC2 인스턴스 상태로 `running` 바꾸십시오.

```
aws ec2 describe-instances \
     --region region-code --filters \
     "Name=tag:aws:pcs:compute-node-group-id,Values=node-group-id" \
     "Name=instance-state-name,Values=running" \
     --query 'Reservations[*].Instances[*].
{InstanceID:InstanceId,State:State.Name,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress}
```

이 명령은 다음과 비슷한 출력을 반환합니다. 의 `PublicIP` 값은 인스턴스가 프라이빗 서브넷에 있는 `null` 경우입니다.

```
[
    [
        {
            "InstanceID": "i-0123456789abcdefa",
            "State": "running",
            "PublicIP": "18.189.32.188",
            "PrivateIP": "10.0.0.1"
        }
    ]
```

```
]
```

> **Note**
>
> 많은 수의 인스턴스를 `describe-instances` 반환하려는 경우 여러 페이지에 대한 옵션을 사용해야 합니다. 자세한 내용은 Amazon Elastic Compute 클라우드 API 레퍼런스를 참조하십시오 DescribeInstances.

Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS

EC2Amazon에서는 시작 템플릿에 기본 설정 세트를 저장할 수 있으므로 인스턴스를 시작할 때 개별적으로 지정할 필요가 없습니다. AWS PCS컴퓨팅 노드 그룹을 구성하는 유연한 방법으로 시작 템플릿을 통합합니다. 노드 그룹을 생성할 때 시작 템플릿을 제공합니다. AWS PCS이 템플릿에서 파생된 시작 템플릿을 생성합니다. 이 템플릿에는 서비스와 함께 작동하도록 하는 데 도움이 되는 변형이 포함되어 있습니다.

사용자 정의 시작 템플릿을 작성할 때 고려해야 할 옵션과 고려 사항을 이해하면 함께 사용할 템플릿을 작성하는 데 도움이 될 수 있습니다. AWS PCS 시작 템플릿에 대한 자세한 내용은 Amazon EC2 사용 설명서의 시작 템플릿에서 시작 인스턴스에서 인스턴스 시작을 참조하십시오.

주제

개요

EC2시작 템플릿에 포함할 수 있는 30개 이상의 파라미터가 있어 인스턴스 구성 방식의 여러 측면을 제어할 수 있습니다. 대부분은 완벽하게 AWS PCS 호환되지만 몇 가지 예외가 있습니다.

EC2Launch 템플릿의 다음 매개 변수는 서비스에서 직접 관리해야 AWS PCS 하므로 무시됩니다.

- 인스턴스 유형/인스턴스 유형 속성 지정 (`InstanceRequirements`) — 속성 기반 인스턴스 선택을 AWS PCS 지원하지 않습니다.

- 인스턴스 유형 (InstanceType) — 노드 그룹을 생성할 때 인스턴스 유형을 지정합니다.
- 고급 세부 정보/ IAM 인스턴스 프로필 (IamInstanceProfile) — 노드 그룹을 생성하거나 업데이트할 때 이 정보를 제공합니다.
- 고급 세부 API 정보/종료 비활성화 (DisableApiTermination) — 시작하는 노드 그룹 인스턴스의 수명 주기를 AWS PCS 제어해야 합니다.
- 고급 세부 정보/disable API stop (DisableApiStop) — 시작하는 노드 그룹 인스턴스의 수명 주기를 AWS PCS 제어해야 합니다.
- 고급 세부 정보/중지 — 하이버네이트 동작 () **HibernationOptions** — 인스턴스 하이버네이션을 지원하지 않습니다. AWS PCS
- 어드밴스드 디테일/엘라스틱 GPU () ElasticGpuSpecifications — 아마존 엘라스틱 그래픽스는 2024년 1월 8일에 수명을 다했습니다.
- 고급 세부 정보/탄력적 추론 (ElasticInferenceAccelerators) — Amazon Elastic Inference는 신규 고객이 더 이상 사용할 수 없습니다.
- AAdvanced세부 CPU 정보/옵션 지정/코어당 스레드 수 (**ThreadsPerCore**) — 코어당 스레드 수를 1로 설정합니다. AWS PCS

이러한 매개변수에는 다음과 호환성을 지원하는 특수 요구 사항이 있습니다. AWS PCS

- 사용자 데이터 (UserData) — 여러 부분으로 인코딩되어야 합니다. Amazon EC2 사용자 데이터 다루기을 참조하세요.
- 애플리케이션 및 OS 이미지 (ImageId) — 이를 포함할 수 있습니다. 하지만 노드 그룹을 만들거나 업데이트할 때 AMI ID를 지정하면 시작 템플릿의 값이 무시됩니다. AMI제공하는 항목은 AWS PCS 호환되어야 합니다. 자세한 내용은 “을 참조하십시오아마존 머신 이미지 (AMIs) 용 AWS PCS.
- 네트워크 설정/방화벽 (보안 그룹) (SecurityGroups) - 시작 템플릿에서 보안 그룹 이름 목록을 설정할 수 없습니다. AWS PCS 시작 템플릿에서 네트워크 인터페이스를 정의하지 않는 한 보안 그룹 목록 IDs (SecurityGroupIds) 을 설정할 수 있습니다. 그런 다음 각 인터페이스의 보안 그룹을 IDs 지정해야 합니다. 자세한 내용은 의 보안 그룹 AWS PCS 단원을 참조하십시오.
- 네트워크 설정/고급 네트워크 구성 (NetworkInterfaces) - 단일 네트워크 카드가 있는 EC2 인스턴스를 사용하고 특별한 네트워킹 구성이 필요하지 않은 경우 인스턴스 네트워킹을 구성할 AWS PCS 수 있습니다. 여러 네트워크 카드를 구성하거나 인스턴스에서 Elastic Fabric Adapter를 활성화하려면 를 사용하십시오. NetworkInterfaces 각 네트워크 인터페이스에는 보안 그룹 목록이 있어야 IDs 합니다Groups. 자세한 내용은 다중 네트워크 인터페이스 입력 AWS PCS 단원을 참조하십시오.

- 고급 세부 정보/용량 예약 (`CapacityReservationSpecification`) - 설정할 수 있지만 작업할 `CapacityReservationId` 때 특정 항목을 참조할 수는 없습니다. AWS PCS 하지만 용량 예약 그룹을 참조할 수 있습니다. 이 그룹에는 하나 이상의 용량 예약이 포함되어 있습니다. 자세한 내용은 수용 인원 예약 AWS PCS 단원을 참조하십시오.

기본 시작 템플릿 생성

AWS Management Console 또는 를 사용하여 시작 템플릿을 생성할 수 AWS CLI있습니다.

AWS Management Console

시작 템플릿 생성

1. Amazon EC2 콘솔을 열고 시작 템플릿을 선택합니다.
2. Create launch template(시작 템플릿 생성)을 선택합니다.
3. 시작 템플릿 이름 및 설명에서 Launch 템플릿 이름의 고유하고 고유한 이름을 입력합니다.
4. 키 페어 이름의 키 페어 (로그인) 에서 관리 대상 EC2 인스턴스에 로그인하는 데 사용할 SSH 키 페어를 선택합니다 AWS PCS. 이는 선택 사항이며, 권장 사항은 아닙니다.
5. 네트워크 설정, 방화벽 (보안 그룹) 에서 네트워크 인터페이스에 연결할 보안 그룹을 선택합니다. 시작 템플릿의 모든 보안 그룹은 AWS PCS 클러스터의 보안 그룹이어야 합니다VPC. 최소한 다음을 선택하십시오.
   - AWS PCS클러스터와의 통신을 허용하는 보안 그룹
   - 에서 시작한 EC2 인스턴스 간 통신을 허용하는 보안 그룹입니다. AWS PCS
   - (선택 사항) 대화형 인스턴스에 대한 인바운드 SSH 액세스를 허용하는 보안 그룹
   - (선택 사항) 컴퓨팅 노드가 인터넷에 발신 연결을 할 수 있도록 허용하는 보안 그룹
   - (선택 사항) 공유 파일 시스템 또는 데이터베이스 서버와 같은 네트워크 리소스에 대한 액세스를 허용하는 보안 그룹.
6. Amazon EC2 콘솔의 시작 템플릿에서 새 시작 템플릿 ID에 액세스할 수 있습니다. 시작 템플릿 ID의 양식은 다음과 같습니다`lt-0123456789abcdef01`.

권장되는 다음 단계

- 새 시작 템플릿을 사용하여 AWS PCS 컴퓨팅 노드 그룹을 만들거나 업데이트하십시오.

AWS CLI

시작 템플릿 생성

다음 명령을 사용하여 시작 템플릿을 생성합니다.

- 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

  a. Replace *region-code* 작업 중인 AWS 리전 위치와 함께 사용하세요. AWS PCS

  b. Replace *my-launch-template-name* 템플릿 이름을 넣으세요. 사용 중인 과 (AWS 계정 과 AWS 리전) 고유해야 합니다.

  c. Replace *my-ssh-key-name* 선호하는 SSH 키의 이름과 함께.

  d. Replace *sg-ExampleID1* 그리고 *sg-ExampleID2* 인스턴스와 스케줄러 간 통신 및 EC2 EC2 인스턴스 간 통신을 IDs 허용하는 보안 그룹을 사용합니다. 이 모든 트래픽을 활성화하는 보안 그룹이 하나뿐인 경우, `sg-ExampleID2` 및 그 앞의 쉼표를 제거할 수 있습니다. 보안 그룹을 더 추가할 수도 있습니다. IDs 시작 템플릿에 포함하는 모든 보안 그룹은 AWS PCS 클러스터의 것이어야 합니다VPC.

```
aws ec2 create-launch-template --region region-code \
    --launch-template-name my-template-name \
    --launch-template-data '{"KeyName":"my-ssh-key-name","SecurityGroupIds":
 ["sg-ExampleID1","sg-ExampleID2"]}'
```

AWS CLI 는 다음과 비슷한 텍스트를 출력합니다. 시작 템플릿 ID는 에서 찾을 수 있습니다. `LaunchTemplateId`

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0123456789abcdef01",
        "LaunchTemplateName": "my-launch-template-name",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-04-30T18:16:06.000Z"
    }
}
```

권장되는 다음 단계

- 새 시작 템플릿을 사용하여 AWS PCS 컴퓨팅 노드 그룹을 만들거나 업데이트하십시오.

Amazon EC2 사용자 데이터 다루기

인스턴스 시작 시 `cloud-init` 실행되는 시작 템플릿에 EC2 사용자 데이터를 제공할 수 있습니다. 콘텐츠 유형의 사용자 데이터 블록은 인스턴스가 에 등록되기 전에 `cloud-config` 실행되는 반면 AWS PCSAPI, 콘텐츠 유형의 사용자 데이터 블록은 등록이 완료된 후 Slurm 데몬이 시작되기 전에 `text/x-shellscript` 실행됩니다. 콘텐츠 유형에 대한 자세한 내용은 cloud-init 문서를 참조하세요.

사용자 데이터는 다음을 포함하되 이에 국한되지 않는 일반적인 구성 시나리오를 수행할 수 있습니다.

- 사용자 또는 그룹 포함
- 패키지 설치
- 파티션 및 파일 시스템 생성
- 네트워크 파일 시스템 마운팅

시작 템플릿의 사용자 데이터는 MIME멀티파트 아카이브 형식이어야 합니다. 이는 사용자 데이터가 노드 그룹에서 노드를 구성하는 데 필요한 다른 AWS PCS 사용자 데이터와 병합되기 때문입니다. 여러 사용자 데이터 블록을 하나의 MIME 멀티파트 파일로 결합할 수 있습니다.

MIME멀티파트 파일은 다음과 같은 구성 요소로 구성됩니다.

- 콘텐츠 유형 및 부분 경계 선언: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- MIME버전 선언: `MIME-Version: 1.0`
- 다음 구성 요소를 포함하는 하나 이상의 사용자 데이터 블록:
  - 사용자 데이터 블록의 시작을 나타내는 시작 경계: `--==BOUNDARY==` 이 경계 앞의 라인은 비워 두어야 합니다.
  - 블록의 콘텐츠 유형 선언: `Content-Type: text/cloud-config; charset="us-ascii"` 또는`Content-Type: text/x-shellscript; charset="us-ascii"`. 콘텐츠 유형 선언 뒤의 라인은 비워 두어야 합니다.
  - 쉘 명령 또는 `cloud-config` 지시어 목록 등의 사용자 데이터 콘텐츠
- MIME여러 부분으로 구성된 파일의 끝을 나타내는 종료 경계:. `--==BOUNDARY==--` 종료 경계 앞의 라인은 비워 두어야 합니다.

> **Note**
>
> Amazon EC2 콘솔의 시작 템플릿에 사용자 데이터를 추가하는 경우 일반 텍스트로 붙여넣을 수 있습니다. 또는 파일에서 업로드할 수도 있습니다. AWS CLI 또는 AWS SDK an을 사용하는 경우 이 JSON 파일에 표시된 대로 먼저 사용자 데이터를 base64로 인코딩하고 호출 CreateLaunchTemplate시 해당 문자열을 `UserData` 파라미터 값으로 제출해야 합니다.

```
{
    "LaunchTemplateName": "base64-user-data",
    "LaunchTemplateData": {
        "UserData":
 "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
    }
}
```

예시

- 예: 패키지 리포지토리에서 소프트웨어 설치
- 예: S3 버킷에서 스크립트 실행
- 예: 글로벌 환경 변수 설정
- 네트워크 파일 시스템 사용 AWS PCS
- 예: EFS 파일 시스템을 공유 홈 디렉터리로 사용

예: 패키지 리포지토리에서 소프트웨어 설치 AWS PCS

이 스크립트를 시작 `"userData"` 템플릿의 값으로 제공하십시오. 자세한 내용은 Amazon EC2 사용자 데이터 다루기 단원을 참조하십시오.

이 스크립트는 시작 시 cloud-config를 사용하여 노드 그룹 인스턴스에 소프트웨어 패키지를 설치합니다. 자세한 내용은 cloud-init 설명서의 사용자 데이터 형식을 참조하십시오. 이 예제는 및 를 설치합니다`curl`. `llvm`

> **Note**
>
> 인스턴스는 구성된 패키지 리포지토리에 연결할 수 있어야 합니다.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- python3-devel
- rust
- golang

--==MYBOUNDARY==--
```

예: S3 AWS PCS 버킷에서 에 대한 추가 스크립트 실행

이 스크립트를 시작 "userData" 템플릿의 값으로 제공하십시오. 자세한 내용은 Amazon EC2 사용자 데이터 다루기 단원을 참조하십시오.

이 스크립트는 cloud-config를 사용하여 S3 버킷에서 스크립트를 가져와서 시작 시 노드 그룹 인스턴스에서 실행합니다. 자세한 내용은 cloud-init 설명서의 사용자 데이터 형식을 참조하십시오.

이 스크립트의 다음 값을 원하는 세부 정보로 바꾸십시오.

- *my-bucket-name* — 계정에서 읽을 수 있는 S3 버킷의 이름.
- *path* — S3 버킷 루트를 기준으로 한 경로.
- *shell* — 스크립트를 실행하는 데 사용할 Linux 셸 (예:bash.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- aws s3 cp s3://my-bucket-name/path /tmp/script.sh
- /usr/bin/shell /tmp/script.sh

--==MYBOUNDARY==--
```

노드 그룹의 IAM 인스턴스 프로필에 버킷에 대한 액세스 권한이 있어야 합니다. 다음 IAM 정책은 위 사용자 데이터 스크립트의 버킷 예제입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket-name",
                "arn:aws:s3:::my-bucket-name/path/*"
            ]
        }
    ]
}
```

예: 에 대한 글로벌 환경 변수 설정 AWS PCS

이 스크립트를 시작 "userData" 템플릿의 값으로 제공하십시오. 자세한 내용은 Amazon EC2 사용자 데이터 다루기 단원을 참조하십시오.

다음 예제는 노드 그룹 인스턴스에 글로벌 변수를 설정하는 /etc/profile.d 데 사용합니다.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
touch /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR1=100 >> /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR2=abc >> /etc/profile.d/awspcs-userdata-vars.sh

--==MYBOUNDARY==--
```

예: EFS 파일 시스템을 공유 홈 디렉토리로 사용 AWS PCS

이 스크립트를 시작 "userData" 템플릿의 값으로 제공하십시오. 자세한 내용은 Amazon EC2 사용자 데이터 다루기 단원을 참조하십시오.

이 예제는 EFS 마운트 인 예제를 네트워크 파일 시스템 사용 AWS PCS 확장하여 공유 홈 디렉터리를 구현합니다. /home의 내용은 EFS 파일 시스템이 마운트되기 전에 백업됩니다. 그러면 마운트가 완료된 후 콘텐츠가 공유 스토리지의 제자리에 빠르게 복사됩니다.

이 스크립트의 다음 값을 사용자 세부 정보로 바꾸십시오.

- *`/mount-point-directory`* — EFS 파일 시스템을 마운트하려는 인스턴스의 경로.
- *`filesystem-id`* — 파일 시스템의 EFS 파일 시스템 ID.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /tmp/home
  - rsync -a /home/ /tmp/home
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults
  - rsync -a --ignore-existing /tmp/home/ /home
  - rm -rf /tmp/home/

--==MYBOUNDARY==--
```

비밀번호 없이 활성화 SSH

공유 홈 디렉터리 예제를 기반으로 빌드하여 키를 사용하여 SSH 클러스터 인스턴스 간 SSH 연결을 구현할 수 있습니다. 공유 홈 파일 시스템을 사용하는 각 사용자에 대해 다음과 유사한 스크립트를 실행합니다.

```
#!/bin/bash
```

```
mkdir -p $HOME/.ssh && chmod 700 $HOME/.ssh
touch $HOME/.ssh/authorized_keys
chmod 600 $HOME/.ssh/authorized_keys

if [ ! -f "$HOME/.ssh/id_rsa" ]; then
    ssh-keygen -t rsa -b 4096 -f $HOME/.ssh/id_rsa -N ""
    cat ~/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
fi
```

> **Note**
>
> 인스턴스는 클러스터 노드 간 SSH 연결을 허용하는 보안 그룹을 사용해야 합니다.

수용 인원 예약 AWS PCS

온디맨드 EC2 용량 예약 또는 용량 블록을 사용하여 특정 가용 영역에서 특정 기간 동안 Amazon EC2 용량을 예약하여 필요할 때 필요한 컴퓨팅 파워를 사용할 수 있도록 할 수 있습니다.

> **Note**
>
> AWS PCS온디맨드 용량 예약 (ODCR) 을 지원하지만 ML용 용량 블록은 현재 지원하지 않습니다.

와 함께 사용 ODCRs AWS PCS

예약 AWS PCS 인스턴스의 소비 방식을 선택할 수 있습니다. ODCR오픈을 생성하면 계정의 다른 AWS PCS 프로세스나 다른 프로세스에서 실행한 일치하는 모든 인스턴스가 예약에 포함됩니다. 타겟팅된 ODCR 경우 특정 예약 ID로 시작된 인스턴스만 예약에 포함됩니다. 시간에 민감한 워크로드의 경우 ODCRs 타겟팅이 더 일반적입니다.

시작 템플릿에 대상을 ODCR 추가하여 대상을 사용하도록 AWS PCS 컴퓨팅 노드 그룹을 구성할 수 있습니다. 이를 위한 단계는 다음과 같습니다.

1. 대상 온디맨드 용량 예약 (ODCR) 을 생성하십시오.
2. 용량 예약 그룹에 추가합니다. ODCR
3. 용량 예약 그룹을 시작 템플릿에 연결합니다.

4. 시작 템플릿을 사용할 AWS PCS 컴퓨팅 노드 그룹을 생성하거나 업데이트하십시오.

예: 타겟팅된 hpc6a.48xlarge 인스턴스를 예약하고 사용하십시오. ODCR

이 예제 명령은 32개의 hpc6a.48xlarge 인스턴스를 대상으로 생성합니다. ODCR 배치 그룹에서 예약 인스턴스를 시작하려면 명령에 추가하십시오. `--placement-group-arn` 중지 날짜를 `--end-date` 및 로 정의할 수 있습니다. `--end-date-type` 그렇지 않으면 예약은 수동으로 종료될 때까지 계속됩니다.

```
aws ec2 create-capacity-reservation \
    --instance-type hpc6a.48xlarge \
    --instance-platform Linux/UNIX \
    --availability-zone us-east-2a \
    --instance-count 32 \
    --instance-match-criteria targeted
```

이 명령의 결과는 새 ARN ODCR 명령의 결과가 됩니다. ODCRwith AWS PCS 를 사용하려면 용량 예약 그룹에 추가해야 합니다. 개인을 지원하지 AWS PCS 않기 때문입니다ODCRs. 자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 용량 예약 그룹을 참조하십시오.

라는 이름의 용량 예약 그룹에 ODCR 를 추가하는 방법은 다음과 같습니다`EXAMPLE-CR-GROUP`.

```
aws resource-groups group-resources --group EXAMPLE-CR-GROUP \
    --resource-arns arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/
cr-1234567890abcdef1
```

용량 예약 그룹을 ODCR 생성하여 추가했으므로 이제 시작 템플릿에 추가하여 AWS PCS 컴퓨팅 노드 그룹에 연결할 수 있습니다. 다음은 용량 예약 그룹을 참조하는 예제 시작 템플릿입니다.

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-2:123456789012:group/EXAMPLE-CR-GROUP"
  }
}
```

마지막으로 hpc6a.48xlarge 인스턴스를 사용하고 해당 용량 예약 그룹에서 를 참조하는 시작 템플릿을 사용하도록 AWS PCS 컴퓨팅 노드 그룹을 생성하거나 업데이트하십시오ODCR. 정적 노드 그룹의 경우 최소 및 최대 인스턴스를 예약 크기 (32) 로 설정합니다. 동적 노드 그룹의 경우 최소 인스턴스를 0으로 설정하고 최대 인스턴스를 예약 크기까지 설정합니다.

이 예는 하나의 컴퓨팅 노드 그룹에 ODCR 프로비저닝된 단일 노드를 간단하게 구현한 것입니다. 하지만 다른 디자인도 많이 AWS PCS 지원합니다. 예를 들어 대규모 ODCR 또는 용량 예약 그룹을 여러 컴퓨팅 노드 그룹으로 세분화할 수 있습니다. 또는 다른 AWS 계정에서 생성하여 공유한 계정을 사용할 ODCRs 수도 있습니다. 주요 제약 조건은 ODCRs 항상 용량 예약 그룹에 포함되어야 한다는 것입니다.

자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 ML용 온디맨드 용량 예약 및 용량 블록을 참조하십시오.

유용한 시작 템플릿 매개변수

이 섹션에서는 광범위하게 유용할 수 있는 몇 가지 시작 템플릿 파라미터에 대해 설명합니다. AWS PCS

세부 CloudWatch 모니터링 켜기

시작 템플릿 파라미터를 사용하여 더 짧은 간격으로 CloudWatch 지표 수집을 활성화할 수 있습니다.

AWS Management Console

시작 템플릿을 만들거나 편집하기 위한 콘솔 페이지의 고급 세부 정보 섹션에서 이 옵션을 찾을 수 있습니다. 세부 CloudWatch 모니터링을 활성화로 설정합니다.

YAML

```
Monitoring:
    Enabled: True
```

JSON

```
{"Monitoring": {"Enabled": "True"}}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스에 대한 세부 모니터링 활성화 또는 해제를 참조하십시오.

인스턴스 메타데이터 서비스 버전 2 (IMDSv2)

IMDSv2를 EC2 인스턴스와 함께 사용하면 보안이 크게 향상되고 환경의 인스턴스 메타데이터 액세스와 관련된 잠재적 위험을 완화하는 데 도움이 됩니다. AWS

AWS Management Console

시작 템플릿을 만들거나 편집하기 위한 콘솔 페이지의 고급 세부 정보 섹션에서 이 옵션을 찾을 수 있습니다. 메타데이터 액세스를 활성화로 설정하고, 메타데이터 버전을 V2만 (토큰 필요) 으로 설정하고, 메타데이터 응답 홉 제한을 4로 설정합니다.

YAML

```
MetadataOptions:
  HttpEndpoint: enabled
  HttpTokens: required
  HttpPutResponseHopLimit: 4
```

JSON

```
{
    "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpPutResponseHopLimit": 4,
        "HttpTokens": "required"
    }
}
```

.

AWS PCS대기열

AWS PCS대기열은 스케줄러의 기본 작업 대기열 구현을 단순하게 추상화한 것입니다. Slurm의 경우 AWS PCS 대기열은 Slurm 파티션과 동일합니다.

사용자는 하나 이상의 컴퓨팅 노드 그룹에서 제공하는 노드에서 실행되도록 스케줄링할 수 있을 때까지 대기열에 작업을 제출합니다. AWS PCS클러스터에는 작업 대기열이 여러 개 있을 수 있습니다. 예를 들어 우선 순위가 높은 작업에는 Amazon EC2 On-demand Instances를 사용하는 대기열을 생성하고 우선 순위가 낮은 작업에는 Amazon EC2 스팟 인스턴스를 사용하는 대기열을 하나 더 생성할 수 있습니다.

주제

- 에서 대기열 만들기 AWS PCS
- AWS PCS대기열 업데이트

에서 대기열 만들기 AWS PCS

이 항목에서는 사용 가능한 옵션에 대한 개요를 제공하고 에서 대기열을 만들 때 고려해야 할 사항에 대해 설명합니다 AWS PCS.

사전 조건

- AWSPCS클러스터 - 대기열은 특정 PCS 클러스터와 관련해서만 생성할 수 있습니다.
- 하나 이상의 AWS PCS 컴퓨팅 노드 그룹 - 대기열은 하나 이상의 PCS 컴퓨팅 노드 그룹과 연결되어야 합니다.

에서 대기열을 만들려면 AWS PCS

AWS Management Console 또는 를 사용하여 대기열을 만들 수 AWS CLI있습니다.

AWS Management Console

콘솔을 사용하여 대기열을 만들려면

1. 다음 위치에서 AWS PCS 콘솔을 엽니다. `https://console.aws.amazon.com/pcs/home#/clusters`
2. 대기열을 생성하려는 클러스터를 선택합니다. 대기열로 이동하여 대기열 생성을 선택합니다.
3. 큐 구성 섹션에서 다음 값을 입력합니다.
    a. 대기열 이름 - 대기열의 이름입니다. 이름에는 영숫자(대소문자 구분)와 하이픈만 사용할 수 있습니다. 영문자로 시작해야 하며 25자를 초과할 수 없습니다. 이름은 클러스터 내에서 고유해야 합니다.
    b. 컴퓨팅 노드 그룹 - 이 대기열에 서비스를 제공할 컴퓨팅 노드 그룹을 하나 이상 선택합니다. 컴퓨팅 노드 그룹은 둘 이상의 대기열과 연결될 수 있습니다.
4. (선택 사항) 태그에서 AWS PCS 대기열에 태그를 추가합니다.
5. 대기열 생성을 선택합니다. 대기열을 설정하는 동안 상태 필드에 작성 중이라는 메시지가 표시됩니다. 대기열 생성에는 몇 분이 걸릴 수 있습니다.

권장되는 다음 단계

- 새 대기열에 작업 제출

AWS CLI

를 사용하여 대기열을 만들려면 AWS CLI

다음 명령을 사용하여 대기열을 생성합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

1. Replace *region-code* 클러스터를 생성하려는 AWS 지역과 함께
2. Replace *my-queue* 대기열 이름과 함께 이름에는 영숫자(대소문자 구분)와 하이픈만 사용할 수 있습니다. 알파벳 문자로 시작해야 하며 25자를 초과할 수 없습니다. 이름은 클러스터 내에서 고유해야 합니다.
3. Replace *my-cluster* clusterId 클러스터의 이름 또는 이름과 함께
4. 의 값을 computeNodeGroupId 자체 컴퓨팅 노드 그룹 식별자로 바꾸십시오. 대기열을 생성할 때는 컴퓨팅 노드 그룹 이름을 지정할 수 없다는 점에 유의하십시오.

```
aws pcs create-queue --region region-code \
    --queue-name my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=computeNodeGroupExampleID1
```

대기열을 생성하는 데 몇 분 정도 걸릴 수 있습니다. 다음 명령을 사용하여 대기열 상태를 쿼리할 수 있습니다. 상태가 될 때까지는 대기열에 작업을 제출할 수 없습니다ACTIVE.

```
aws pcs get-queue --region region-code \
    --cluster-identifier my-cluster \
    --queue-identifier my-queue
```

권장되는 다음 단계

- 새 대기열에 작업 제출

AWS PCS대기열 업데이트

이 항목에서는 사용 가능한 옵션에 대한 개요를 제공하고 AWS PCS 대기열을 업데이트할 때 고려할 사항을 설명합니다.

AWSPCS대기열 업데이트 시 고려 사항

큐 업데이트는 실행 중인 작업에 영향을 주지 않지만 큐가 업데이트되는 동안에는 클러스터가 새 작업을 수락하지 못할 수 있습니다.

AWSPCS컴퓨팅 노드 그룹을 업데이트하려면

AWS관리 콘솔 또는 를 사용하여 노드 그룹을 업데이트할 수 AWS CLI 있습니다.

AWS Management Console

대기열을 업데이트하려면

1. 다음 위치에서 AWS PCS 콘솔을 엽니다. `https://console.aws.amazon.com/pcs/home#/clusters`
2. 대기열을 업데이트하려는 클러스터를 선택합니다.
3. 대기열로 이동하여 업데이트하려는 대기열로 이동한 다음 편집을 선택합니다.
4. 대기열 구성 섹션에서 다음 값 중 하나를 업데이트하십시오.
   - 노드 그룹 - 대기열과의 연결에서 컴퓨팅 노드 그룹을 추가하거나 제거합니다.
   - 태그 - 대기열에 태그를 추가하거나 제거합니다.
5. 업데이트를 선택합니다. 상태 필드에는 변경 사항이 적용되는 동안 업데이트 중이라는 내용이 표시됩니다.

> **Important**
> 대기열 업데이트는 몇 분 정도 걸릴 수 있습니다.

AWS CLI

대기열을 업데이트하려면

1. 다음 명령을 사용하여 대기열을 업데이트하십시오. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

   a. Replace *region-code* 클러스터를 AWS 리전 생성하려는 항목으로

   b. Replace *my-queue* 이름을 붙이거나 computeNodeGroupId 대기열에 넣을 수 있습니다.

   c. Replace *my-cluster* clusterId클러스터의 이름 또는 이름과 함께

   d. 컴퓨팅 노드 그룹 연결을 변경하려면 에 대한 업데이트된 목록을 제공하십시오--compute-node-group-configurations.

      - 예를 들어, 두 번째 컴퓨팅 노드 그룹을 추가하려면computeNodeGroupExampleID2:

        ```
        --compute-node-group-configurations
         computeNodeGroupId=computeNodeGroupExampleID1,computeNodeGroupId=computeNodeGro
        ```

   ```
   aws pcs update-queue --region region-code \
       --queue-identifier my-queue \
       --cluster-identifier my-cluster \
       --compute-node-group-configurations \
       computeNodeGroupId=computeNodeGroupExampleID1
   ```

2. 대기열을 업데이트하는 데 몇 분이 걸릴 수 있습니다. 다음 명령을 사용하여 대기열 상태를 쿼리할 수 있습니다. 상태가 될 때까지는 대기열에 작업을 제출할 수 없습니다ACTIVE.

   ```
   aws pcs get-queue --region region-code \
       --cluster-identifier my-cluster \
       --queue-identifier my-queue
   ```

권장되는 다음 단계

- 업데이트된 대기열에 작업을 제출하십시오.

에서 대기열 삭제 AWS PCS

이 항목에서는 에서 대기열을 삭제하는 방법에 대한 개요를 제공합니다 AWS PCS.

대기열을 삭제할 때 고려할 사항

- 큐에서 실행 중인 작업이 있는 경우 큐가 삭제되면 스케줄러가 해당 작업을 종료합니다. 대기열에 있는 보류 중인 작업은 취소됩니다. 대기열에 있는 작업이 완료될 때까지 기다리거나 스케줄러의 기본 명령 (예: Slurm) 을 사용하여 작업을 수동으로 중지/취소하는 것을 고려해 보십시오. `scancel`

대기열 삭제

OR를 사용하여 대기열을 삭제할 수 있습니다. AWS Management Console AWS CLI

AWS Management Console

대기열을 삭제하려면

1. AWS PCS콘솔을 엽니다.
2. 큐의 클러스터를 선택합니다.
3. 대기열로 이동하여 삭제할 대기열을 선택합니다.
4. Delete(삭제)를 선택합니다.
5. 상태 필드가 표시됩니다. `Deleting` 완료되는 데 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 스케줄러에 기본 제공되는 명령을 사용하여 대기열이 삭제되었는지 확인할 수 있습니다. 예를 들어 Slurm에는 `sinfo` 또는 `squeue` 를 사용하십시오.

AWS CLI

대기열을 삭제하려면

- 다음 명령을 사용하여 대기열을 삭제하십시오. 대기열은 다음과 같이 대체됩니다.
  - Replace *`region-code`* 클러스터가 AWS 리전 연결되어 있는 상태입니다.
  - Replace *`my-queue`* 대기열의 이름 또는 ID와 함께

- Replace *my-cluster* 클러스터의 이름 또는 ID와 함께

```
aws pcs delete-queue --region region-code \
       --queue-identifier my-queue \
       --cluster-identifier my-cluster
```

대기열을 삭제하는 데 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 스케줄러에 기본 제공되는 명령을 사용하여 대기열이 삭제되었는지 확인할 수 있습니다. 예를 들어 Slurm에는 `sinfo` 또는 `squeue` 를 사용하십시오.

AWS PCS로그인 노드

일반적으로 AWS PCS 클러스터에는 대화형 액세스 및 작업 관리를 지원하는 로그인 노드가 1개 이상 필요합니다. 이 작업을 수행하는 방법은 로그인 노드 기능을 위해 구성된 정적 AWS PCS 컴퓨팅 노드 그룹을 사용하는 것입니다. 독립 실행형 EC2 인스턴스가 로그인 노드로 작동하도록 구성할 수도 있습니다.

주제

- AWS PCS컴퓨팅 노드 그룹을 사용하여 로그인 노드 제공
- 독립형 인스턴스를 AWS PCS 로그인 노드로 사용

AWS PCS컴퓨팅 노드 그룹을 사용하여 로그인 노드 제공

이 항목에서는 제안된 구성 옵션에 대한 개요를 제공하고 AWS PCS 컴퓨팅 노드 그룹을 사용하여 클러스터에 대한 지속적인 대화형 액세스를 제공할 때 고려해야 할 사항을 설명합니다.

로그인 노드용 AWS PCS 컴퓨팅 노드 그룹 생성

운영상 이는 일반 컴퓨팅 노드 그룹을 생성하는 것과 크게 다르지 않습니다. 하지만 다음과 같은 몇 가지 주요 구성 선택 사항이 있습니다.

- 컴퓨팅 노드 그룹에 있는 하나 이상의 EC2 인스턴스에 대한 정적 확장 구성을 설정합니다.
- 인스턴스가 회수되지 않도록 하려면 온디맨드 구매 옵션을 선택하십시오.

- 컴퓨팅 노드 그룹의 정보를 제공하는 이름 (예: 로그인) 을 선택합니다.
- 외부에서 로그인 노드 인스턴스에 액세스할 수 있게 하려면 퍼블릭 서브넷을 사용하는 것이 좋습니다. VPC
- SSH액세스를 허용하려면 선택한 IP 주소에 SSH 포트를 노출시키는 보안 그룹이 시작 템플릿에 있어야 합니다.
- IAM인스턴스 프로필에는 최종 사용자에게 AWS 부여하려는 권한만 있어야 합니다. 세부 정보는 IAM AWS 병렬 컴퓨팅 서비스의 인스턴스 프로필를 참조하세요.
- AWSSystems Manager 세션 관리자가 로그인 인스턴스를 관리하도록 허용하는 것을 고려해 보십시오.
- 인스턴스 AWS 자격 증명에 대한 액세스를 관리자만 사용할 수 있도록 제한하는 방안을 고려해 보십시오.
- 로그인 노드가 계속 실행되므로 일반 컴퓨팅 노드 그룹보다 비용이 저렴한 인스턴스 유형을 선택하십시오.
- 다른 컴퓨팅 노드 AMI 그룹과 동일 (또는 파생) 을 사용하면 모든 인스턴스에 동일한 소프트웨어가 설치되도록 할 수 있습니다. 사용자 AMIs 지정에 대한 자세한 내용은 을 참조하십시오. 아마존 머신 이미지 (AMIs) 용 AWS PCS
- 컴퓨팅 인스턴스와 동일한 네트워크 파일 시스템 (AmazonEFS, Amazon FSx for Lustre 등) 마운트를 로그인 노드에 구성합니다. 자세한 내용은 네트워크 파일 시스템 사용 AWS PCS 단원을 참조하십시오.

로그인 노드에 액세스하세요.

새 컴퓨팅 노드 그룹이 ACTIVE 상태에 도달하면 생성된 EC2 인스턴스를 찾아 로그인할 수 있습니다. 자세한 내용은 에서 컴퓨팅 노드 그룹 인스턴스 찾기 AWS PCS 단원을 참조하십시오.

로그인 노드의 AWS PCS 컴퓨팅 노드 그룹 업데이트

를 사용하여 로그인 노드 그룹을 업데이트할 수 UpdateComputeNodeGroup 있습니다. 노드 그룹 업데이트 프로세스의 일부로 실행 중인 인스턴스가 교체됩니다. 이렇게 하면 인스턴스의 활성 사용자 세션이나 프로세스가 모두 중단된다는 점에 유의하십시오. 실행 중이거나 대기 중인 Slurm 작업은 영향을 받지 않습니다. 자세한 내용은 AWS PCS컴퓨팅 노드 그룹 업데이트 단원을 참조하십시오.

컴퓨팅 노드 그룹에서 사용하는 시작 템플릿을 편집할 수도 있습니다. 를 UpdateComputeNodeGroup 사용하여 업데이트된 시작 템플릿을 컴퓨팅 노드 그룹에 적용해야 합니다. 컴퓨팅 노드 그룹에서 시작된 새 EC2 인스턴스는 업데이트된 시작 템플릿을 사용합니다. 자세한 내용은 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS 단원을 참조하십시오.

로그인 노드의 AWS PCS 컴퓨팅 노드 그룹 삭제

에서 컴퓨팅 노드 그룹 삭제 메커니즘을 사용하여 로그인 노드 그룹을 업데이트할 수 AWS PCS 있습니다. 노드 그룹 삭제의 일환으로 실행 중인 인스턴스가 종료됩니다. 이렇게 하면 인스턴스의 활성 사용자 세션이나 프로세스가 모두 중단된다는 점에 유의하십시오. 실행 중이거나 대기 중인 Slurm 작업은 영향을 받지 않습니다. 자세한 내용은 에서 컴퓨팅 노드 그룹 삭제 AWS PCS 단원을 참조하십시오.

독립형 인스턴스를 AWS PCS 로그인 노드로 사용

AWS PCS클러스터의 Slurm 스케줄러와 상호 작용하도록 독립 EC2 인스턴스를 설정할 수 있습니다. 이는 AWS PCS 클러스터와 함께 작동하지만 관리 외부에서 작동하는 로그인 노드, 워크스테이션 또는 전용 워크플로 관리 호스트를 만드는 데 유용합니다. AWS PCS 이를 위해서는 각 독립형 인스턴스가 다음을 수행해야 합니다.

1. 호환되는 Slurm 소프트웨어 버전을 설치해야 합니다.
2. AWS PCS클러스터의 Slurmctld 엔드포인트에 연결할 수 있어야 합니다.
3. 클러스터의 엔드포인트 및 시크릿으로 Slurm Auth 및 Cred Kiosk 데몬 () 데몬 () 을 적절하게 구성하십시오. `sackd` AWS PCS 자세한 내용은 Slurm 설명서의 sackd를 참조하십시오.

이 자습서는 클러스터에 연결되는 독립 인스턴스를 구성하는 데 도움이 됩니다. AWS PCS

목차

1단계 — 대상 AWS PCS 클러스터의 주소 및 암호 검색

다음 AWS CLI 명령과 함께 를 사용하여 대상 AWS PCS 클러스터에 대한 세부 정보를 검색합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.

- Replace *`region-code`* 대상 클러스터가 실행되고 AWS 리전 있는 위치와 함께
- Replace *`cluster-ident`* 대상 클러스터의 이름 또는 식별자와 함께

```
aws pcs get-cluster --region region-code --cluster-identifier cluster-ident
```

이 명령은 이 예제와 비슷한 출력을 반환합니다.

```
{
    "cluster": {
        "name": "independent-instance-demo",
        "id": "s3431v9rx2",
        "arn": "arn:aws:pcs:us-east-1:012345678901:cluster/s3431v9rx2",
        "status": "ACTIVE",
        "createdAt": "2024-07-12T15:32:27.225136+00:00",
        "modifiedAt": "2024-07-12T15:32:27.225136+00:00",
        "scheduler": {
            "type": "SLURM",
            "version": "23.11"
        },
        "size": "SMALL",
        "networking": {
            "subnetIds": [
                "subnet-0123456789abdef"
            ],
            "securityGroupIds": [
                "sg-0123456789abdef"
            ]
        },
        "endpoints": [
            {
                "type": "SLURMCTLD",
                "privateIpAddress": "10.3.149.220",
                "port": "6817"
            }
        ],
        "authKey": {
            "secretArn": "arn:aws:secretsmanager:us-east-1:123456789012:secret:pcs!
slurm-secret-s3431v9rx2-FN7tJFf",
            "secretVersion": "ff58d1fd-070e-4bbc-98a0-64ef967cebcc"
        }
    }
}
```

이 샘플에서 클러스터 Slurm 컨트롤러 엔드포인트는 IP 주소가 `10.3.149.220` 1이고 포트에서 실행되고 있습니다. 6817 `secretArn`는 이후 단계에서 클러스터 암호를 검색하는 데 사용됩니다. IP 주소 및 포트는 이후 단계에서 `sackd` 서비스를 구성하는 데 사용됩니다.

2단계 — EC2 인스턴스 시작

EC2 인스턴스를 시작하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택한 다음에 Launch Instances(인스턴스 시작)를 선택하여 새 인스턴스 시작 마법사를 엽니다.
3. (선택 사항) 이름 및 태그 섹션에서 인스턴스의 이름을 입력합니다 (예:)`PCS-LoginNode`. 이름은 인스턴스에 리소스 태그(`Name=PCS-LoginNode`)로 할당됩니다.
4. 애플리케이션 및 OS 이미지 섹션에서 에서 지원하는 운영 체제 중 하나를 선택합니다 AWS PCS. AMI 자세한 내용은 지원되는 운영 체제 단원을 참조하십시오.
5. 인스턴스 유형 섹션에서 지원되는 인스턴스 유형을 선택합니다. 자세한 내용은 지원되는 인스턴스 유형 단원을 참조하십시오.
6. Key pair 섹션에서 인스턴스에 사용할 SSH 키 페어를 선택합니다.
7. 네트워크 설정 섹션에서:
   - 편집을 선택합니다.
     i. AWS PCS클러스터를 VPC 선택합니다.
     ii. 방화벽(보안 그룹)에서 기존 보안 그룹 선택을 선택합니다.
       A. 인스턴스와 대상 AWS PCS 클러스터의 Slurm 컨트롤러 간의 트래픽을 허용하는 보안 그룹을 선택합니다. 자세한 내용은 보안 그룹 요구 사항 및 고려 사항 단원을 참조하십시오.
       B. (선택 사항) 인스턴스에 대한 인바운드 SSH 액세스를 허용하는 보안 그룹을 선택합니다.
8. 스토리지 섹션에서 필요에 따라 스토리지 볼륨을 구성합니다. 사용 사례를 지원하려면 애플리케이션과 라이브러리를 설치할 수 있는 충분한 공간을 구성해야 합니다.
9. 고급에서 클러스터 암호에 대한 액세스를 허용하는 IAM 역할을 선택합니다. 자세한 내용은 Slurm 클러스터 시크릿 확인하기 단원을 참조하십시오.
10. 요약 창에서 인스턴스 시작을 선택합니다.

3단계 — 인스턴스에 Slurm 설치

인스턴스가 시작되어 활성화되면 원하는 메커니즘을 사용하여 인스턴스에 연결합니다. 에서 제공하는 Slurm 설치 프로그램을 사용하여 인스턴스에 AWS Slurm을 설치합니다. 자세한 내용은 Slurm 인스톨러 단원을 참조하십시오.

Slurm 설치 프로그램을 다운로드하고 압축을 풀고 스크립트를 사용하여 Slurm을 설치합니다. `installer.sh` 자세한 내용은 3단계 — Slurm 설치 단원을 참조하십시오.

4단계 — 클러스터 시크릿 검색 및 저장

이 지침에는 가 필요합니다 AWS CLI. 자세한 내용은 버전 2용AWS Command Line Interface 사용 설명서의 최신 버전 설치 또는 업데이트를 참조하십시오. AWS CLI

다음 명령을 사용하여 클러스터 암호를 저장합니다.

- Slurm의 구성 디렉터리를 생성합니다.

```
sudo mkdir -p /etc/slurm
```

- 클러스터 암호를 검색, 디코딩 및 저장합니다. 이 명령을 실행하기 전에 다음을 대체하십시오.*region-code* 대상 클러스터가 실행되고 있는 지역으로 교체하십시오.*secret-arn* 1단계에서 `secretArn` 검색한 값과 함께

```
sudo aws secretsmanager get-secret-value \
    --region region-code \
    --secret-id 'secret-arn' \
    --version-stage AWSCURRENT \
    --query 'SecretString' \
    --output text | base64 -d > /etc/slurm/slurm.key
```

> **Warning**
>
> 다중 사용자 환경에서는 인스턴스에 액세스할 수 있는 모든 사용자가 인스턴스 메타데이터 서비스 () 에 액세스할 수 있으면 클러스터 암호를 가져올 수 있습니다. IMDS 이로 인해 다른 사용자로 가장할 수 있습니다. 루트 또는 관리자 사용자만 액세스할 수 IMDS 있도록 제한하는 방안을 고려해 보세요. 또는 인스턴스 프로필에 의존하지 않고 시크릿을 가져오고 구성하는 다른 메커니즘을 사용하는 것도 고려해 보세요.

- Slurm 키 파일에 소유권과 권한을 설정합니다.

```
sudo chmod 0600 /etc/slurm/slurm.key
sudo chown slurm:slurm /etc/slurm/slurm.key
```

> **Note**
>
> Slurm 키는 서비스를 실행하는 사용자 및 그룹이 소유해야 합니다. sackd

5단계 — 클러스터에 대한 연결 구성 AWS PCS

AWS PCS클러스터에 연결하려면 다음 단계에 따라 시스템 sackd 서비스로 시작하십시오.

1. 다음 명령을 사용하여 sackd 서비스의 환경 파일을 설정합니다. 명령을 실행하기 전에 다음을 대체하십시오.*ip-address* 그리고 *port* 1단계에서 엔드포인트에서 검색한 값으로

   ```
   sudo echo "SACKD_OPTIONS='--conf-server=ip-address:port'" > /etc/sysconfig/sackd
   ```

2. 프로세스 관리를 위한 systemd 서비스 파일을 생성합니다. sackd

   ```
   sudo cat << EOF > /etc/systemd/system/sackd.service
   [Unit]
   Description=Slurm auth and cred kiosk daemon
   After=network-online.target remote-fs.target
   Wants=network-online.target
   ConditionPathExists=/etc/sysconfig/sackd

   [Service]
   Type=notify
   EnvironmentFile=/etc/sysconfig/sackd
   User=slurm
   Group=slurm
   RuntimeDirectory=slurm
   RuntimeDirectoryMode=0755
   ExecStart=/opt/aws/pcs/scheduler/slurm-23.11/sbin/sackd --systemd \$SACKD_OPTIONS
   ExecReload=/bin/kill -HUP \$MAINPID
   KillMode=process
   LimitNOFILE=131072
   LimitMEMLOCK=infinity
   LimitSTACK=infinity
   ```

```
[Install]
WantedBy=multi-user.target
EOF
```

3. sackd서비스 파일의 소유권을 설정합니다.

```
sudo chown root:root /etc/systemd/system/sackd.service && \
    sudo chmod 0644 /etc/systemd/system/sackd.service
```

4. sackd서비스를 활성화합니다.

```
sudo systemctl daemon-reload && sudo systemctl enable sackd
```

5. sackd 서비스를 시작합니다.

```
sudo systemctl start sackd
```

6단계 — (선택 사항) 연결 테스트

sackd서비스가 실행 중인지 확인합니다. 샘플 출력은 다음과 같습니다. 오류가 있는 경우 일반적으로 여기에 표시됩니다.

```
[root@ip-10-3-27-112 ~]# systemctl status sackd
[x] sackd.service - Slurm auth and cred kiosk daemon
   Loaded: loaded (/etc/systemd/system/sackd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-07-16 16:34:55 UTC; 8s ago
 Main PID: 9985 (sackd)
   CGroup: /system.slice/sackd.service
           ##9985 /opt/aws/pcs/scheduler/slurm-23.11/sbin/sackd --systemd --conf-
server=10.3.149.220:6817

Jul 16 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Starting Slurm auth and cred
 kiosk daemon...
Jul 16 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Started Slurm auth and cred
 kiosk daemon.
Jul 16 16:34:55 ip-10-3-27-112.ec2.internal sackd[9985]: sackd: running
```

및 와 같은 sinfo Slurm 클라이언트 명령을 사용하여 클러스터에 대한 연결이 작동하는지 확인합니다. squeue 다음은 의 출력 예제입니다. sinfo

```
[root@ip-10-3-27-112 ~]# /opt/aws/pcs/scheduler/slurm-23.11/bin/sinfo
```

```
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
all up infinite 4 idle~ compute-[1-4]
```

또한 작업을 제출할 수 있어야 합니다. 예를 들어, 이 예제와 비슷한 명령은 클러스터의 한 노드에서 대화형 작업을 시작합니다.

```
/opt/aws/pcs/scheduler/slurm-23.11/bin/srun --nodes=1 -p all --pty bash -i
```

AWS PCS네트워킹

AWS PCS클러스터는 Amazon에서 생성됩니다VPC. 이 장에는 클러스터 스케줄러 및 노드의 네트워킹에 관한 다음 주제가 포함되어 있습니다.

인스턴스를 시작할 서브넷을 선택하는 경우를 제외하고, EC2 시작 템플릿을 사용하여 AWS PCS 컴퓨팅 노드 그룹의 네트워킹을 구성해야 합니다. 시작 템플릿에 대한 자세한 내용은 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS의 내용을 참조하세요.

주제

- AWS PCSVPC및 서브넷 요구 사항 및 고려 사항
- VPC AWS PCS클러스터용 생성
- 의 보안 그룹 AWS PCS
- 다중 네트워크 인터페이스 입력 AWS PCS
- EC2인스턴스의 배치 그룹 AWS PCS
- 탄성 패브릭 어댑터 (EFA) 와 함께 사용 AWS PCS

AWS PCSVPC및 서브넷 요구 사항 및 고려 사항

AWS PCS클러스터를 생성할 때 해당 클러스터에 VPC 서브넷을 지정합니다. VPC 이 항목에서는 클러스터에서 사용하는 및 서브넷의 AWS PCS 특정 요구 사항 VPC 및 고려 사항에 대한 개요를 제공합니다. 와 함께 사용할 VPC 필요가 없는 경우 AWS제공된 AWS PCS AWS CloudFormation 템플릿을 사용하여 새로 만들 수 있습니다. 에 대한 VPCs 자세한 내용은 Amazon VPC 사용 설명서의 가상 사설 클라우드 (VPC) 를 참조하십시오.

VPC요구 사항 및 고려 사항

클러스터를 생성할 때 지정하는 VPC 것은 다음 요구 사항 및 고려 사항을 충족해야 합니다.

- 에는 생성하려는 클러스터, 노드 및 기타 클러스터 리소스에 사용할 수 있는 충분한 수의 IP 주소가 VPC 있어야 합니다. 자세한 내용은 Amazon VPC 사용 설명서의 사용자 VPCs 및 서브넷의 IP 주소 지정을 참조하십시오.
- DNS호스트 이름 및 DNS 해상도 지원이 VPC 있어야 합니다. 그렇지 않으면 노드가 고객 클러스터를 등록할 수 없습니다. 자세한 내용은 Amazon VPC 사용 설명서에서 사용자의 DNS 속성을 참조하십시오. VPC
- 에 VPC 연락하려면 VPC 엔드포인트를 사용해야 AWS PrivateLink 할 수 있습니다. AWS PCS API 자세한 내용은 Amazon 사용 VPC 설명서를 VPC사용하여 AWS PrivateLink서비스에 연결을 참조하십시오.

서브넷 요구 사항 및 고려 사항

Slurm 클러스터를 생성하면 지정한 서브넷에 엘라스틱 네트워크 인터페이스 (ENI) 가 AWS PCS 생성됩니다. 이 네트워크 인터페이스를 통해 스케줄러 컨트롤러와 고객 간의 통신이 가능합니다. VPC 또한 Slurm은 네트워크 인터페이스를 통해 고객 계정에 배포된 구성 요소와 통신할 수 있습니다. 클러스터의 서브넷은 생성 시에만 지정할 수 있습니다.

클러스터의 서브넷 요구 사항

클러스터를 생성할 때 지정하는 서브넷은 다음 요구 사항을 충족해야 합니다.

- 에서 사용할 수 있는 서브넷에는 하나 이상의 IP 주소가 있어야 합니다. AWS PCS
- 서브넷은 AWS Outposts AWS Wavelength, 또는 로컬 영역에 있을 수 없습니다. AWS
- 서브넷은 퍼블릭 또는 프라이빗일 수 있습니다. 가능하면 프라이빗 서브넷을 지정하는 것이 좋습니다. 퍼블릭 서브넷은 인터넷 게이트웨이에 대한 경로가 포함된 라우팅 테이블이 있는 서브넷이고, 프라이빗 서브넷은 인터넷 게이트웨이에 대한 경로가 포함되지 않은 라우팅 테이블이 있는 서브넷입니다.

노드의 서브넷 요구 사항

노드 및 기타 클러스터 리소스를 클러스터를 생성할 때 지정한 서브넷과 동일한 서브넷에 배포할 수 있습니다 AWS PCS. VPC

노드와 클러스터 리소스를 배포하는 모든 서브넷은 다음 요구 사항을 충족해야 합니다.

- 서브넷에 모든 노드와 클러스터 리소스를 배포할 수 있는 충분한 IP 주소가 있는지 확인해야 합니다.
- 퍼블릭 서브넷에 노드를 배포하려는 경우 해당 서브넷은 퍼블릭 주소를 자동 IPv4 할당해야 합니다.

- 노드를 배포하는 서브넷이 사설 서브넷이고 해당 라우팅 테이블에 네트워크 주소 변환 (NAT) 장치 (IPv4) 에 대한 경로가 포함되어 있지 않은 경우 고객에게 사용하는 VPC 엔드포인트를 추가하십시오. AWS PrivateLink VPC VPC엔드포인트는 노드가 연결하는 모든 AWS 서비스에 필요합니다. 필요한 유일한 엔드포인트는 노드가 `registerNodeGroupInstances` API 작업을 AWS PCS 호출하도록 허용하는 것입니다.
- 퍼블릭 또는 프라이빗 서브넷 상태는 영향을 받지 않으므로 필요한 엔드포인트에 연결할 수 있어야 합니다 AWS PCS.

VPC AWS PCS클러스터용 생성

AWS 병렬 컴퓨팅 서비스 (VPC) 내에서 클러스터용 Amazon Virtual Private Cloud (Amazon AWS PCS) 를 생성할 수 있습니다.

VPCAmazon을 사용하여 사용자가 정의한 가상 네트워크로 VPC 리소스를 시작할 수 있습니다. 이 가상 네트워크는 사용자의 자체 데이터 센터에서 운영하는 기존 네트워크와 매우 유사합니다. 그러나 Amazon Web Services Services의 확장 가능한 인프라를 사용하면 얻을 수 있는 이점이 있습니다. 프로덕션 VPC 클러스터를 배포하기 전에 Amazon VPC 서비스를 완전히 이해하는 것이 좋습니다. 자세한 내용은 Amazon이란 무엇입니까VPC? 를 참조하십시오. 작성자 비주얼 모드에서. 아마존 VPC 사용 설명서.

PCS클러스터, 노드 및 지원 리소스 (예: 파일 시스템 및 디렉터리 서비스) 가 Amazon 내에 배포됩니다VPC. 기존 Amazon을 VPC 와 함께 PCS 사용하려면 에 설명된 요건을 충족해야 합니다AWS PCSVPC및 서브넷 요구 사항 및 고려 사항. 이 주제에서는 AWS—제공된 AWS CloudFormation 템플릿을 사용하여 PCS 요건을 VPC 충족하는 템플릿을 생성하는 방법을 설명합니다. 템플릿을 배포했다면 템플릿에서 생성된 리소스를 보면서 생성된 리소스와 해당 리소스의 구성을 정확히 알 수 있습니다.

사전 조건

Amazon VPC for PCS 를 생성하려면 Amazon VPC 리소스를 생성하는 데 필요한 IAM 권한이 있어야 합니다. 이러한 리소스는 서브넷VPCs, 보안 그룹, 라우팅 테이블 및 경로, 인터넷 및 NAT 게이트웨이입니다. 자세한 내용은 Amazon VPC 사용 설명서의 퍼블릭 VPC 서브넷으로 생성을 참조하십시오. EC2Amazon의 전체 목록을 검토하려면 서비스 승인 EC2 참조의 Amazon 작업, 리소스 및 조건 키를 참조하십시오.

아마존 만들기 VPC

사용할 AWS 리전 PCS 위치에 URL 적합한 것을 VPC 복사하여 붙여넣어 생성하십시오. AWS CloudFormation 템플릿을 다운로드하여 AWS CloudFormation 콘솔에 직접 업로드할 수도 있습니다.

- 미국 동부(버지니아 북부)(us-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 미국 동부(오하이오)(us-east-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 미국 서부(오레곤)(us-west-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-west-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- 템플릿만

```
https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/
assets/main.yaml
```

VPCAmazon을 만들려면 PCS

1. AWS CloudFormation 콘솔에서 템플릿을 엽니다.

   > **Note**
   >
   > 템플릿에 미리 채워져 있으므로 기본값으로 두기만 하면 됩니다.

2. 스택 이름 제공에서 스택 이름을 입력한 다음 입력합니다. `hpc-networking`
3. 매개 변수에 다음 세부 정보를 입력합니다.

   a. 아래에 VPCCidrBlock다음을 입력합니다. `10.3.0.0/16`

   b. 서브넷 A에서:

      i. 그런 다음 CidrPublicSubnetA, 다음을 입력하십시오. `10.3.0.0/20`

      ii. 그럼 CidrPrivateSubnetA, 엔터 `10.3.128.0/20`

c. 서브넷 B에서:

i. 그런 다음 CidrPublicSubnetB, 다음을 입력하십시오. `10.3.16.0/20`

ii. 그럼 CidrPrivateSubnetA, 엔터 `10.3.144.0/20`

d. 서브넷 C에서:

i. ProvisionSubnetsC의 경우 선택합니다`True`.

> **Note**
>
> 가용 영역이 3개 미만인 VPC 지역에서 만드는 경우 이 옵션을 로 설정하면 무시됩니다`True`.

ii. 그런 다음 CidrPublicSubnetB, 다음을 입력하십시오. `10.3.32.0/20`

iii. 그럼 CidrPrivateSubnetA, 엔터 `10.3.160.0/20`

4. 기능에서 IAM 리소스를 생성할 AWS CloudFormation 수 있음을 인정함 체크박스를 선택합니다.

AWS CloudFormation 스택 상태를 모니터링합니다. `CREATE_COMPLETE`도달하면 VPC 리소스를 사용할 준비가 된 것입니다.

> **Note**
>
> AWS CloudFormation 템플릿에서 생성한 모든 리소스를 보려면 AWS CloudFormation 콘솔을 여십시오. `hpc-networking` 스택을 선택한 다음 리소스(Resources) 탭을 선택합니다.

의 보안 그룹 AWS PCS

Amazon의 보안 그룹은 인스턴스로의 인바운드 및 아웃바운드 트래픽을 제어하는 가상 방화벽 EC2 역할을 합니다. AWS PCS컴퓨팅 노드 그룹의 시작 템플릿을 사용하여 인스턴스에 보안 그룹을 추가하거나 제거합니다. 시작 템플릿에 네트워크 인터페이스가 없는 경우 `SecurityGroupIds` 를 사용하여 보안 그룹 목록을 제공하십시오. 시작 템플릿이 네트워크 인터페이스를 정의하는 경우 `Groups` 파라미터를 사용하여 각 네트워크 인터페이스에 보안 그룹을 할당해야 합니다. 시작 템플릿에 대한 자세한 내용은 Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS의 내용을 참조하세요.

> **Note**
>
> 시작 템플릿의 보안 그룹 구성 변경 사항은 컴퓨팅 노드 그룹이 업데이트된 후 시작된 새 인스턴스에만 영향을 줍니다.

보안 그룹 요구 사항 및 고려 사항

AWS PCS클러스터를 생성할 때 지정한 서브넷에 계정 간 엘라스틱 네트워크 인터페이스 (ENI) 를 생성합니다. 이렇게 하면 에서 관리하는 계정에서 실행되는 HPC 스케줄러가 에서 시작한 EC2 인스턴스와 통신할 수 있는 경로가 제공됩니다. AWS AWS PCS ENI스케줄러와 클러스터 인스턴스 간의 양방향 통신을 ENI 허용하는 보안 그룹을 제공해야 합니다. EC2

이를 수행하는 간단한 방법은 그룹의 모든 구성원 간에 모든 포트에서 TCP /IP 트래픽을 허용하는 허용적인 자체 참조 보안 그룹을 만드는 것입니다. 이를 클러스터와 노드 그룹 인스턴스 모두에 연결할 수 있습니다. EC2

허용적 보안 그룹 구성의 예

규칙 타입	프로토콜	포트	소스	대상
인바운드	모두	모두	본인	
아웃바운드	모두	모두		0.0.0.0/0
아웃바운드	모두	모두		본인

이 규칙은 Slurm 컨트롤러와 노드 간에 모든 트래픽이 자유롭게 흐르도록 허용하고, 모든 아웃바운드 트래픽을 모든 목적지로 허용하고, 트래픽을 활성화합니다. EFA

제한적 보안 그룹 구성의 예

클러스터와 해당 컴퓨팅 노드 사이의 열린 포트를 제한할 수도 있습니다. Slurm 스케줄러의 경우 클러스터에 연결된 보안 그룹이 다음 포트를 허용해야 합니다.

- 6817 - 인스턴스로부터의 인바운드 연결을 활성화합니다. `slurmctld` EC2
- 6818 — 인스턴스에서 `slurmctld` 실행되도록 아웃바운드 연결을 활성화합니다. `slurmd` EC2

컴퓨팅 노드에 연결된 보안 그룹은 다음 포트를 허용해야 합니다.

- 6817 - 인스턴스와의 아웃바운드 연결을 활성화합니다`slurmctld`. EC2
- 6818 — 노드 그룹 인스턴스와의 인바운드 및 아웃바운드 연결을 활성화합니다`slurmd`. `slurmctld` `slurmd`
- 60001—63000 — 노드 그룹 인스턴스 간의 인바운드 및 아웃바운드 연결 지원 `srun`
- EFA노드 그룹 인스턴스 간 트래픽. 자세한 내용은 Linux 인스턴스용 사용 설명서의 EFA -enabled 보안 그룹 준비를 참조하십시오.
- 워크로드에 필요한 기타 모든 노드 간 트래픽

다중 네트워크 인터페이스 입력 AWS PCS

일부 EC2 인스턴스에는 여러 네트워크 카드가 있습니다. 이를 통해 100Gbps 이상의 대역폭 용량 및 향상된 패킷 처리 등 더 높은 네트워크 성능을 제공할 수 있습니다. 여러 네트워크 카드가 있는 인스턴스에 대한 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 엘라스틱 네트워크 인터페이스를 참조하십시오.

EC2시작 템플릿에 네트워크 인터페이스를 추가하여 AWS PCS 컴퓨팅 노드 그룹의 인스턴스에 대한 추가 네트워크 카드를 구성하십시오. 다음은 `hpc7a.96xlarge` 인스턴스에서 찾을 수 있는 것과 같은 두 개의 네트워크 카드를 사용할 수 있는 예제 시작 템플릿입니다. 다음 세부 정보를 참고하십시오.

- 각 네트워크 인터페이스의 서브넷은 시작 템플릿을 사용할 AWS PCS 컴퓨팅 노드 그룹을 구성할 때 선택한 것과 동일해야 합니다.
- SSH및 HTTPS 트래픽과 같은 일상적인 네트워크 통신이 발생하는 기본 네트워크 디바이스는 `DeviceIndex` a를 설정하여 설정됩니다. `0` 다른 네트워크 인터페이스에는 a가 `DeviceIndex` 있습니다1. 기본 네트워크 인터페이스는 하나만 있을 수 있고 다른 인터페이스는 모두 보조 인터페이스입니다.
- 모든 네트워크 인터페이스는 고유해야 합니다. `NetworkCardIndex` 시작 템플릿에 정의된 대로 순차적으로 번호를 매기는 것이 좋습니다.
- 각 네트워크 인터페이스의 보안 그룹은 를 사용하여 `Groups` 설정됩니다. 이 예에서는 인바운드 SSH 보안 그룹 (sg-*`SshSecurityGroupId`*) 이 기본 네트워크 인터페이스에 추가되고 클러스터 내 통신을 가능하게 하는 보안 그룹 () 이 추가됩니다. sg-*`ClusterSecurityGroupId`* 마지막으로 인터넷으로의 아웃바운드 연결을 허용하는 보안 그룹 (sg-*`InternetOutboundSecurityGroupId`*) 이 기본 인터페이스와 보조 인터페이스 모두에 추가됩니다.

```
{
```

```
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "NetworkCardIndex": 0,
            "SubnetId": "subnet-SubnetId",
            "Groups": [
               "sg-SshSecurityGroupId",
               "sg-ClusterSecurityGroupId",
               "sg-InternetOutboundSecurityGroupId"
            ]
        },
        {
            "DeviceIndex": 1,
            "NetworkCardIndex": 1,
            "SubnetId": "subnet-SubnetId",
            "Groups": ["sg-InternetOutboundSecurityGroupId"]
        }
    ]
}
```

EC2인스턴스의 배치 그룹 AWS PCS

배치 그룹을 사용하여 인스턴스에서 실행되는 워크로드의 요구 사항에 맞게 EC2 인스턴스 배치에 영향을 줄 수 있습니다.

배치 그룹 유형

- 클러스터 — 가용 영역에서 인스턴스를 서로 가깝게 패킹하여 지연 시간이 짧은 통신을 최적화합니다.
- 파티션 - 인스턴스를 논리적 파티션에 분산하여 복원력을 극대화합니다.
- 분산 — 소수의 인스턴스가 별도의 하드웨어에서 시작되도록 엄격하게 적용하므로 복원력에도 도움이 될 수 있습니다.

자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 Amazon EC2 인스턴스용 배치 그룹을 참조하십시오.

Elastic Fabric Adapter (EFA) 를 사용하도록 AWS PCS 컴퓨팅 노드 그룹을 구성할 때 클러스터 배치 그룹을 포함하는 것이 좋습니다.

다음과 함께 작동하는 클러스터 배치 그룹을 만들려면 EFA

1. 컴퓨팅 노드 그룹의 클러스터 유형을 사용하여 배치 그룹을 생성합니다.
   - 다음 AWS CLI 명령을 사용하세요.

     ```
     aws ec2 create-placement-group --strategy cluster --group-name PLACEMENT-GROUP-NAME
     ```

   - CloudFormation 템플릿을 사용하여 배치 그룹을 만들 수도 있습니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation템플릿 작업을 참조하십시오. URL다음에서 템플릿을 다운로드하여 CloudFormation 콘솔에 업로드하십시오.

     ```
     https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-placement-group.yaml
     ```

2. AWS PCS컴퓨팅 노드 그룹의 EC2 시작 템플릿에 배치 그룹을 포함시키십시오.

탄성 패브릭 어댑터 (EFA) 와 함께 사용 AWS PCS

Elastic Fabric Adapter (EFA) 는 EC2 인스턴스에 연결하여 고성능 컴퓨팅 (HPC) 및 기계 학습 애플리케이션을 가속화할 수 AWS 있는 고성능 고급 네트워킹 상호 연결입니다. 를 사용하여 AWS PCS 클러스터에서 애플리케이션을 실행하도록 EFA 하려면 AWS PCS 컴퓨팅 노드 그룹 인스턴스를 다음과 EFA 같이 사용하도록 구성해야 합니다.

목차

호환되는 EFA AWS PCS 서버에 설치 AMI

AWS PCS컴퓨팅 노드 그룹에서 AMI 사용되는 드라이버에는 EFA 드라이버가 설치 및 로드되어 있어야 합니다. EFA소프트웨어가 설치된 AMI 상태에서 사용자 지정을 구축하는 방법에 대한 자세한 내용은 을 참조하십시오사용자 지정 Amazon 머신 이미지 (AMIs) 에 대한 AWS PCS.

EFA활성화된 인스턴스 EC2 식별

사용하려면 EFA AWS PCS 컴퓨팅 그룹에 허용되는 모든 인스턴스 유형이 EFA 지원되어야 하고 개수 vCPUs (해당하는 GPUs 경우) 가 같아야 합니다. EFA활성화된 인스턴스 목록은 Amazon Elastic Compute Cloud 사용 설명서의 Amazon EC2 기반 엘라스틱 패브릭 어댑터 HPC 및 ML 워크로드를 참조하십시오. 를 AWS CLI 사용하여 지원하는 인스턴스 유형 목록을 볼 수도 있습니다. EFA Replace *region-code* 사용하는 AWS 리전 곳 (예:) 과 함께 사용하십시오 AWS PCSus-east-1.

```
aws ec2 describe-instance-types \
   --region region-code \
   --filters Name=network-info.efa-supported,Values=true \
   --query "InstanceTypes[*].[InstanceType]" \
   --output text | sort
```

사용 가능한 네트워크 인터페이스 수를 결정하십시오.

일부 EC2 인스턴스에는 네트워크 카드가 여러 개 있습니다. 이렇게 하면 여러 개를 가질 수 EFAs 있습니다. 자세한 내용은 다중 네트워크 인터페이스 입력 AWS PCS 단원을 참조하십시오.

EFA통신을 지원하는 보안 그룹을 만드세요.

AWS CLI

다음 AWS CLI 명령을 사용하여 지원하는 보안 그룹을 생성할 수 EFA 있습니다. 이 명령은 보안 그룹 ID를 출력합니다. 다음과 같이 교체하십시오.

- *region-code*— 사용 AWS 리전 AWS PCS 위치를 지정하십시오 (예:). us-east-1
- *vpc-id*— 사용하는 ID의 VPC ID를 지정합니다 AWS PCS.
- *efa-group-name*— 보안 그룹에 대해 선택한 이름을 입력합니다.

```
aws ec2 create-security-group \
    --group-name efa-group-name \
```

```
    --description "Security group to enable EFA traffic" \
    --vpc-id vpc-id \
    --region region-code
```

다음 명령을 사용하여 인바운드 및 아웃바운드 보안 그룹 규칙을 연결합니다. 다음과 같이 교체하십시오.

- *efa-secgroup-id*— 방금 생성한 EFA 보안 그룹의 ID를 입력합니다.

```
aws ec2 authorize-security-group-ingress \
    --group-id efa-secgroup-id \
    --protocol -1 \
    --source-group efa-secgroup-id

aws ec2 authorize-security-group-egress \
    --group-id efa-secgroup-id \
    --protocol -1 \
    --source-group efa-secgroup-id
```

CloudFormation template

CloudFormation 템플릿을 사용하여 지원하는 보안 그룹을 생성할 수 EFA 있습니다. URL다음에서 템플릿을 다운로드한 다음 AWS CloudFormation 콘솔에 업로드하십시오.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-sg.yaml
```

AWS CloudFormation 콘솔에서 템플릿을 연 상태에서 다음 옵션을 입력합니다.

- 스택 이름 제공에서
  - 스택 이름에 다음과 같은 이름을 입력합니다`efa-sg-stack`.
- 파라미터에서
  - 아래에 SecurityGroupName다음과 같은 이름을 입력합니다`efa-sg`.
  - 아래에서 VPC사용할 VPC 위치를 선택합니다 AWS PCS.

CloudFormation 스택 생성을 완료하고 상태를 모니터링합니다. EFA보안 그룹에 `CREATE_COMPLETE` 도달하면 사용할 준비가 된 것입니다.

(선택 사항) 배치 그룹 만들기

클러스터 배치 EFA 그룹에서 사용하는 모든 인스턴스를 시작하여 인스턴스 간의 물리적 거리를 최소화하는 것이 좋습니다. 사용할 각 컴퓨팅 노드 그룹에 대해 배치 그룹을 생성하는 것이 좋습니다EFA. 컴퓨팅 노드 그룹을 위한 배치 그룹을 EC2인스턴스의 배치 그룹 AWS PCS 생성하려면 을 참조하십시오.

EC2시작 템플릿 생성 또는 업데이트

EFA네트워크 인터페이스는 AWS PCS 컴퓨팅 노드 그룹의 EC2 시작 템플릿에 설정됩니다. 네트워크 카드가 여러 개 있는 경우 여러 개를 구성할 EFAs 수 있습니다. EFA보안 그룹과 선택적 배치 그룹도 시작 템플릿에 포함됩니다.

다음은 hpc7a.96xlarge와 같이 네트워크 카드가 두 개 있는 인스턴스에 대한 예제 시작 템플릿입니다. 인스턴스는 클러스터 배치 그룹에서 시작됩니다. `subnet-`*SubnetID1* `pg-`*PlacementGroupId1*

보안 그룹은 각 EFA 인터페이스에 특별히 추가해야 합니다. 모든 EFA 제품에는 EFA 트래픽을 활성화하는 보안 그룹이 필요합니다 (`sg-`*EfaSecGroupId*). 기타 보안 그룹, 특히 SSH HTTPS OR와 같은 일반 트래픽을 처리하는 보안 그룹은 기본 네트워크 인터페이스 (`DeviceIndex`a로 지정`0`) 에만 연결하면 됩니다. 네트워크 인터페이스가 정의된 시작 템플릿은 `SecurityGroupIds` 매개 변수를 사용한 보안 그룹 설정을 지원하지 않으므로 구성하는 각 네트워크 인터페이스에서 값을 설정해야 합니다. `Groups`

```
{
    "Placement": {
        "GroupId": "pg-PlacementGroupId1"
    },
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "InterfaceType": "efa",
            "NetworkCardIndex": 0,
            "SubnetId": "subnet-SubnetId1",
            "Groups": [
                "sg-SecurityGroupId1",
                "sg-EfaSecGroupId"
            ]
        },
        {
            "DeviceIndex": 1,
            "InterfaceType": "efa",
            "NetworkCardIndex": 1,
```

```
            "SubnetId": "subnet-SubnetId1"
            "Groups": ["sg-EfaSecGroupId"]
        }
    ]
}
```

컴퓨팅 노드 그룹 생성 또는 업데이트

개수가 같고 프로세서 아키텍처가 같고 모두 지원하는 인스턴스로 AWS PCS 컴퓨팅 노드 그룹을 만들거나 EFA 업데이트하십시오. vCPUs EFA소프트웨어가 설치된 AMI 상태에서 를 사용하고 EFA 지원 네트워크 인터페이스를 설정하는 시작 템플릿을 사용하도록 컴퓨팅 노드 그룹을 구성합니다.

(선택 사항) 테스트 EFA

EFA소프트웨어 설치에 포함된 `fi_pingpong` 프로그램을 실행하여 컴퓨팅 노드 그룹의 두 노드 간 EFA -지원 통신을 시연할 수 있습니다. 이 테스트가 성공하면 제대로 구성된 것일 수 있습니다. EFA

시작하려면 컴퓨팅 노드 그룹에서 두 개의 실행 중인 인스턴스가 필요합니다. 컴퓨팅 노드 그룹에서 고정 용량을 사용하는 경우 사용 가능한 인스턴스가 이미 있어야 합니다. 동적 용량을 사용하는 컴퓨팅 노드 그룹의 경우 `salloc` 명령을 사용하여 두 노드를 시작할 수 있습니다. 다음은 이름이 지정된 대기열과 hpc7g 연결된 동적 노드 그룹이 있는 클러스터의 예입니다`all`.

```
% salloc --nodes 2 -p all
salloc: Granted job allocation 6
salloc: Waiting for resource configuration
... a few minutes pass ...
salloc: Nodes hpc7g-[1-2] are ready for job
```

를 사용하여 할당된 두 노드의 IP 주소를 확인합니다`scontrol`. 다음 예제에서 주소는 `10.3.140.69` `10.3.132.211` for hpc7g-1 및 hpc7g-2 for입니다.

```
% scontrol show nodes hpc7g-[1-2]
NodeName=hpc7g-1 Arch=aarch64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=hpc7g
   ActiveFeatures=hpc7g
   Gres=(null)
   NodeAddr=10.3.140.69 NodeHostName=ip-10-3-140-69 Version=23.11.8
   OS=Linux 5.10.218-208.862.amzn2.aarch64 #1 SMP Tue Jun 4 16:52:10 UTC 2024
   RealMemory=124518 AllocMem=0 FreeMem=110763 Sockets=64 Boards=1
   State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
```

```
   Partitions=efa
   BootTime=2024-07-02T19:00:09 SlurmdStartTime=2024-07-08T19:33:25
   LastBusyTime=2024-07-08T19:33:25 ResumeAfterTime=None
   CfgTRES=cpu=64,mem=124518M,billing=64
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
   Reason=Maintain Minimum Number Of Instances [root@2024-07-02T18:59:00]
   InstanceId=i-04927897a9ce3c143 InstanceType=hpc7g.16xlarge

NodeName=hpc7g-2 Arch=aarch64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=hpc7g
   ActiveFeatures=hpc7g
   Gres=(null)
   NodeAddr=10.3.132.211 NodeHostName=ip-10-3-132-211 Version=23.11.8
   OS=Linux 5.10.218-208.862.amzn2.aarch64 #1 SMP Tue Jun 4 16:52:10 UTC 2024
   RealMemory=124518 AllocMem=0 FreeMem=110759 Sockets=64 Boards=1
   State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=efa
   BootTime=2024-07-02T19:00:09 SlurmdStartTime=2024-07-08T19:33:25
   LastBusyTime=2024-07-08T19:33:25 ResumeAfterTime=None
   CfgTRES=cpu=64,mem=124518M,billing=64
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
   Reason=Maintain Minimum Number Of Instances [root@2024-07-02T18:59:00]
   InstanceId=i-0a2c82623cb1393a7 InstanceType=hpc7g.16xlarge
```

(또는SSM) 를 사용하여 SSH 노드 중 하나 (이 예에서는hpc7g-1) 에 연결합니다. 참고로 이 주소는 내부 IP 주소이므로 사용할 경우 로그인 노드 중 하나에서 연결해야 할 수도 SSH 있습니다. 또한 컴퓨팅 노드 그룹 시작 템플릿을 통해 SSH 키를 사용하여 인스턴스를 구성해야 한다는 점도 유의하십시오.

```
% ssh ec2-user@10.3.140.69
```

이제 서버 `fi_pingpong` 모드에서 시작하세요.

```
/opt/amazon/efa/bin/fi_pingpong -p efa
```

두 번째 인스턴스 (hpc7g-2) 에 연결합니다.

```
% ssh ec2-user@10.3.132.211
```

클라이언트 fi_pingpong 모드에서 실행하고 서버에 연결합니다hpc7g-1. 아래 예와 비슷한 출력이 표시될 것입니다.

```
% /opt/amazon/efa/bin/fi_pingpong -p efa 10.3.140.69

bytes   #sent   #ack     total       time     MB/sec    usec/xfer   Mxfers/sec
64      10      =10      1.2k        0.00s      3.08      20.75       0.05
256     10      =10      5k          0.00s     21.24      12.05       0.08
1k      10      =10      20k         0.00s     82.91      12.35       0.08
4k      10      =10      80k         0.00s    311.48      13.15       0.08
[error] util/pingpong.c:1876: fi_close (-22) fid 0
```

(선택 사항) CloudFormation 템플릿을 사용하여 EFA 활성화된 시작 템플릿을 생성합니다.

설정에는 여러 종속성이 있으므로 컴퓨팅 노드 그룹을 구성하는 데 사용할 수 있는 CloudFormation 템플릿이 제공되었습니다. EFA 최대 4개의 네트워크 카드가 있는 인스턴스를 지원합니다. 여러 네트워크 카드가 있는 인스턴스에 대해 자세히 알아보려면 Amazon Elastic Compute Cloud 사용 설명서의 엘라스틱 네트워크 인터페이스를 참조하십시오.

URL다음에서 CloudFormation 템플릿을 다운로드한 다음 사용하는 CloudFormation 콘솔에 AWS 리전 AWS PCS 업로드하십시오.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/pcs-lt-efa.yaml
```

AWS CloudFormation 콘솔에서 템플릿을 연 상태에서 다음 값을 입력합니다. 템플릿에는 몇 가지 기본 매개변수 값이 제공되므로 이 값을 기본값으로 둘 수 있습니다.

- 스택 이름 제공에서
  - 스택 이름에 설명이 포함된 이름을 입력합니다. AWS PCS컴퓨팅 노드 그룹에 대해 선택할 이름 (예:) 을 통합하는 것이 좋습니다. *NODEGROUPNAME*-efa-lt
- 파라미터에서
  - 에서 노드 그룹에 포함될 인스턴스의 네트워크 카드 수를 선택합니다. NumberOfNetworkCards
  - 에서 VpcId AWS PCS클러스터가 VPC 배포되는 위치를 선택합니다.

- 에서 EFA -enabled 인스턴스가 시작될 VPC 클러스터의 서브넷을 선택합니다. NodeGroupSubnetId
- 에서 PlacementGroupName필드를 비워 두고 노드 그룹을 위한 새 클러스터 배치 그룹을 생성합니다. 사용하려는 기존 배치 그룹이 있는 경우 여기에 해당 이름을 입력합니다.
- 에서 ClusterSecurityGroupId클러스터의 다른 인스턴스 및 에 대한 액세스를 허용하기 위해 사용하는 보안 그룹을 선택합니다 AWS PCSAPI. 많은 고객이 클러스터에서 기본 보안 그룹을 선택합니다VPC.
- 에서 SshSecurityGroupId클러스터의 노드에 대한 인바운드 SSH 액세스를 허용하는 데 사용하는 보안 그룹의 ID를 입력합니다.
- 의 경우 SshKeyName, 클러스터의 노드에 액세스하기 위한 SSH 키페어를 선택합니다.
- 의 LaunchTemplateName경우 시작 템플릿을 설명하는 이름 (예:) 을 입력합니다. *NODEGROUPNAME*`-efa-lt` 이름은 사용할 AWS 계정 AWS 리전 AWS PCS 위치에 따라 고유한 이름이어야 합니다.

- 기능 아래서
  - IAM리소스가 AWS CloudFormation 생성될 수 있음을 인정한다는 확인란에 체크하세요.

CloudFormation 스택 상태를 모니터링하세요. 시작 `CREATE_COMPLETE` 템플릿에 도달하면 사용할 준비가 된 것입니다. 위의 설명에 따라 AWS PCS 컴퓨팅 노드 그룹과 함께 사용하십시오컴퓨팅 노드 그룹 생성 또는 업데이트.

네트워크 파일 시스템 사용 AWS PCS

AWS 병렬 컴퓨팅 서비스 (AWS PCS) 컴퓨팅 노드 그룹에서 시작된 노드에 네트워크 스토리지 볼륨을 연결하여 데이터와 파일을 쓰고 액세스할 수 있는 영구 위치를 제공할 수 있습니다. AWS 서비스에서 제공하는 볼륨을 사용할 수 있습니다. 볼륨에는 아마존 Elastic File System (아마존EFS), 아마존 FSx 포 오픈 NetApp ONTAP, 아마존 FSx 포 오픈 ZFS, 아마존 FSx 포 러스터, 아마존 파일 캐시가 포함됩니다. 서버와 같은 자체 관리형 볼륨을 사용할 수도 있습니다. NFS

이 항목에서는 네트워크 파일 시스템 사용에 대한 고려 사항 및 사용 예제를 다룹니다. AWS PCS

네트워크 파일 시스템 사용 고려 사항

다양한 파일 시스템의 구현 세부 사항은 다르지만 몇 가지 일반적인 고려 사항이 있습니다.

- 관련 파일 시스템 소프트웨어를 인스턴스에 설치해야 합니다. 예를 들어 Amazon FSx for Lustre를 사용하려면 적절한 Lustre 패키지가 있어야 합니다. 이를 컴퓨팅 노드 그룹에 AMI 포함시키거나 인스턴스 부팅 시 실행되는 스크립트를 사용하여 이 작업을 수행할 수 있습니다.
- 공유 스토리지 볼륨과 컴퓨팅 노드 그룹 인스턴스 간에는 네트워크 경로가 있어야 합니다.
- 공유 스토리지 볼륨과 컴퓨팅 노드 그룹 인스턴스 모두의 보안 그룹 규칙은 관련 포트에 대한 연결을 허용해야 합니다.
- 파일 시스템에 액세스하는 리소스 전체에서 일관된 POSIX 사용자 및 그룹 네임스페이스를 유지해야 합니다. 그렇지 않으면 PCS 클러스터에서 실행되는 작업과 대화형 프로세스에서 권한 오류가 발생할 수 있습니다.
- 파일 시스템 마운트는 EC2 시작 템플릿을 사용하여 수행됩니다. 네트워크 파일 시스템을 마운트할 때 오류가 발생하거나 제한 시간이 초과되면 인스턴스를 작업 실행에 사용할 수 없게 될 수 있습니다. 이로 인해 예상치 못한 비용이 발생할 수 있습니다. 시작 템플릿 디버깅에 대한 자세한 내용은 을 참조하십시오Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS.

네트워크 마운트 예시

Amazon, Amazon FSx for LustreEFS, Amazon FSx for Open ZFS 및 Amazon 파일 캐시를 사용하여 파일 시스템을 생성할 수 있습니다. 아래 관련 섹션을 확장하여 각 네트워크 마운트의 예를 확인하십시오.

아마존 EFS

파일 시스템 설정

Amazon EFS 파일 시스템을 생성합니다. PCS컴퓨팅 노드 그룹 인스턴스를 시작할 각 가용 영역에 탑재 대상이 있는지 확인하십시오. 또한 각 탑재 대상이 PCS 컴퓨팅 노드 그룹 인스턴스로부터의 인바운드 및 아웃바운드 액세스를 허용하는 보안 그룹과 연결되어 있는지 확인하십시오. 자세한 내용은 Amazon Elastic File System 사용 설명서의 탑재 대상 및 보안 그룹을 참조하십시오.

시작 템플릿

파일 시스템 설정의 보안 그룹을 컴퓨팅 노드 그룹에 사용할 시작 템플릿에 추가합니다.

`cloud-config`메커니즘을 사용하여 Amazon EFS 파일 시스템을 마운트하는 사용자 데이터를 포함하십시오. 이 스크립트의 다음 값을 사용자 세부 정보로 바꾸십시오.

- *`mount-point-directory`*— Amazon을 마운트할 각 인스턴스의 경로 EFS

- *filesystem-id*— 파일 시스템의 EFS 파일 시스템 ID

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /mount-point-directory
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults

--==MYBOUNDARY==--
```

아마존 FSx 포 러스터

파일 시스템 설정

사용할 VPC 위치에 FSx Lustre용 파일 시스템을 생성하십시오. AWS PCS 영역 간 전송을 최소화하려면 대부분의 PCS 컴퓨팅 노드 그룹 인스턴스를 시작할 동일한 가용 영역의 서브넷에 배포하십시오. 파일 시스템이 PCS 컴퓨팅 노드 그룹 인스턴스로부터의 인바운드 및 아웃바운드 액세스를 허용하는 보안 그룹과 연결되어 있는지 확인하십시오. 보안 그룹에 대한 자세한 내용은 Amazon VPC FSx for Lustre 사용 설명서의 Amazon을 통한 파일 시스템 액세스 제어를 참조하십시오.

시작 템플릿

FSxLustre용 파일 시스템을 `cloud-config` 마운트하는 데 사용하는 사용자 데이터를 포함하십시오. 이 스크립트의 다음 값을 사용자 세부 정보로 바꾸십시오.

- *mount-point-directory*— FSx Lustre용으로 마운트하려는 인스턴스의 경로
- *filesystem-id*— Lustre 파일 시스템의 파일 시스템 ID FSx
- *mount-name*— Lustre 파일 시스템의 FSx 마운트 이름
- *region-code*— FSx Lustre용 파일 시스템이 AWS 리전 배포되는 위치 (시스템과 동일해야 함) AWS PCS
- (선택 사항) *latest* — Lustre에서 Lustre 지원하는 FSx 모든 버전

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- amazon-linux-extras install -y lustre=latest
- mkdir -p /mount-point-directory
- mount -t lustre filesystem-id.fsx.region-code.amazonaws.com@tcp:/mount-name /mount-point-directory

--==MYBOUNDARY==
```

아마존 FSx 포 오픈 ZFS

파일 시스템 설정

사용할 VPC 위치에 Open ZFS 파일 FSx 시스템용 파일을 AWS PCS 생성하십시오. 영역 간 전송을 최소화하려면 대부분의 AWS PCS 컴퓨팅 노드 그룹 인스턴스를 시작할 동일한 가용 영역의 서브넷에 배포하십시오. 파일 시스템이 AWS PCS 컴퓨팅 노드 그룹 인스턴스로부터의 인바운드 및 아웃바운드 액세스를 허용하는 보안 그룹과 연결되어 있는지 확인하십시오. 보안 그룹에 대한 자세한 내용은 for Open ZFS User VPC Guide의 Amazon을 FSx 통한 파일 시스템 액세스 관리를 참조하십시오.

시작 템플릿

오픈 ZFS 파일 시스템용 루트 볼륨을 `cloud-config` 마운트하는 데 사용하는 사용자 데이터를 포함하십시오. FSx 이 스크립트의 다음 값을 원하는 세부 정보로 바꾸십시오.

- *`mount-point-directory`*— 공개 ZFS 공유를 위해 마운트하려는 인스턴스의 경로 FSx
- *`filesystem-id`*— 오픈 파일 시스템의 ZFS 파일 시스템 ID FSx
- *`region-code`*- 오픈 ZFS 파일 시스템이 AWS 리전 배포되는 위치 (사용 중인 AWS PCS 시스템과 동일해야 함) FSx

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"
```

```
runcmd:
- mkdir -p /mount-point-directory
- mount -t nfs -o noatime,nfsvers=4.2,sync,rsize=1048576,wsize=1048576 filesystem-
id.fsx.region-code.amazonaws.com:/fsx/ /mount-point-directory

--==MYBOUNDARY==
```

Amazon File Cache

파일 시스템 설정

사용할 VPC 위치에 Amazon 파일 캐시를 생성합니다 AWS PCS. 영역 간 전송을 최소화하려면 대부분의 PCS 컴퓨팅 노드 그룹 인스턴스를 시작할 동일한 가용 영역에서 서브넷을 선택하십시오. 파일 캐시가 포트 988을 통해 PCS 인스턴스와 파일 캐시 간의 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹과 연결되어 있는지 확인하십시오. 보안 그룹에 대한 자세한 내용은 Amazon VPC File Cache 사용 설명서의 Amazon을 통한 캐시 액세스 제어를 참조하십시오.

시작 템플릿

파일 시스템 설정의 보안 그룹을 컴퓨팅 노드 그룹에 사용할 시작 템플릿에 추가합니다.

Amazon 파일 캐시를 `cloud-config` 마운트하는 데 사용하는 사용자 데이터를 포함하십시오. 이 스크립트의 다음 값을 사용자 세부 정보로 바꾸십시오.

- *mount-point-directory*— FSx Lustre용으로 마운트하려는 인스턴스의 경로
- *cache-dns-name*— 파일 캐시의 도메인 이름 시스템 (DNS) 이름
- *mount-name*— 파일 캐시의 마운트 이름

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- amazon-linux-extras install -y lustre=2.12
- mkdir -p /mount-point-directory
- mount -t lustre -o relatime,flock cache-dns-name@tcp:/mount-name /mount-point-
directory
```

```
--==MYBOUNDARY==
```

아마존 머신 이미지 (AMIs) 용 AWS PCS

AWS PCS제공하는 소프트웨어와 함께 AMIs 작동하므로 클러스터의 노드에 있는 소프트웨어 및 구성을 매우 유연하게 사용할 수 있습니다. AWS PCS시험해 보려면 에서 AMI 제공하고 에서 유지 관리하는 샘플을 사용할 수 있습니다. AWS프로덕션 AWS PCS 환경에서 사용하는 경우 직접 제작하는 것이 좋습니다AMIs. 이 항목에서는 샘플을 검색하고 사용하는 방법과 사용자 지정 샘플을 AMIs 직접 제작하여 사용하는 방법을 다룹니다AMIs.

주제

- 샘플 Amazon 머신 이미지 (AMIs) 를 다음과 함께 사용 AWS PCS
- 사용자 지정 Amazon 머신 이미지 (AMIs) 에 대한 AWS PCS
- 맞춤 AMIs 제작을 위한 소프트웨어 설치 프로그램 AWS PCS

샘플 Amazon 머신 이미지 (AMIs) 를 다음과 함께 사용 AWS PCS

AWS작업의 시작점으로 사용할 수 AMIs 있는 샘플을 제공합니다 AWS PCS.

> **Important**
>
> AMIs샘플은 데모용이며 프로덕션 워크로드에는 권장되지 않습니다.

현재 AWS PCS 샘플 찾기 AMIs

AWS Management Console

AWSPCSAMIs샘플의 명명 규칙은 다음과 같습니다.

```
aws-pcs-sample_ami-OS-architecture-schdeulder-scheduler-major-version
```

허용되는 값

- *OS* – amzn2

- *architecture* — x86_64 또는 arm64
- *scheduler* – slurm
- *scheduler-major-version* – 23.11

AWS PCS샘플을 찾으려면 AMIs

1. Amazon EC2 콘솔을 엽니다.
2. AMIs로 이동합니다.
3. 퍼블릭 이미지를 선택합니다.
4. 속성 또는 AMI 태그별 찾기에서 템플릿 이름을 AMI 사용하여 검색하십시오.

   예시

   - 그라비톤을 지원하는 슬럼 23.11 AMI

     ```
     aws-pcs-sample_ami-amzn2-arm64-slurm-23.11
     ```

   - x86 AMI 인스턴스용 샘플

     ```
     aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11
     ```

   > **Note**
   >
   > 여러 AMIs 개의 타임스탬프가 있는 경우 가장 최근의 타임스탬프와 AMI 함께 사용하십시오.

5. 컴퓨팅 노드 그룹을 생성하거나 업데이트할 때 AMI ID를 사용하십시오.

AWS CLI

다음 명령을 사용하여 최신 AWS PCS 샘플을 AMI 찾을 수 있습니다. Replace *region-code* 사용하는 AWS 리전 곳 (예:) 과 함께 사용하십시오 AWS PCSus-east-1.

- x86_64

```
aws ec2 describe-images --region region-code --owners amazon 533267220047
 654654292779 654654317195 975050324343 \
```

```
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11*' \
            'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

- Arm64

```
aws ec2 describe-images --region region-code --owners amazon 533267220047
 654654292779 654654317195 975050324343 \
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-arm64-slurm-23.11*' \
            'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

컴퓨팅 노드 그룹을 생성하거나 업데이트할 때 AMI ID를 사용하십시오.

AWS PCS샘플에 대해 자세히 알아보십시오. AMIs

AWS PCS샘플의 현재 및 이전 릴리스에 대한 내용AMIs, 구성 세부 정보를 보려면 을 참조하십시오AWS PCS샘플용 릴리스 노트 AMIs.

다음과 AMIs 호환되는 자체 제작을 구축하십시오. AWS PCS

함께 사용할 수 AMIs 있는 자체 제작 방법을 AWS PCS 알아보려면 을 참조하십시오사용자 지정 Amazon 머신 이미지 (AMIs) 에 대한 AWS PCS.

사용자 지정 Amazon 머신 이미지 (AMIs) 에 대한 AWS PCS

AWS PCS서비스로 가져오는 Amazon 머신 이미지 (AMI) 와 함께 작동하도록 설계되었습니다. AWS PCS에이전트와 호환 가능한 버전의 Slurm이 올바르게 설치 및 구성되어 있는 한, 이들 서버에 임의의 소프트웨어와 구성을 설치할 AMIs 수 있습니다. AWS-제공된 설치 프로그램을 사용하여 사용자 정의에 AWS PCS 소프트웨어를 설치해야 합니다. AMI AWS제공된 설치 프로그램을 사용하여 사용자 정의에 Slurm을 설치하는 것이 AMI 좋지만 원하는 경우 직접 Slurm을 설치할 수도 있습니다 (권장하지 않음).

> **Note**
>
> 사용자 정의를 AWS PCS AMI 만들지 않고 시도하고 싶다면 에서 제공하는 샘플을 사용할 수 있습니다. AMI AWS자세한 내용은 샘플 Amazon 머신 이미지 (AMIs) 를 다음과 함께 사용 AWS PCS 단원을 참조하십시오.

이 자습서는 PCS 컴퓨팅 노드 그룹과 함께 사용하여 AI/ML 워크로드를 AMI 지원하는 데 사용할 수 있는 모델을 HPC 만드는 데 도움이 됩니다.

주제

1단계 — 임시 인스턴스 시작

AWS PCS소프트웨어와 Slurm 스케줄러를 설치하고 구성하는 데 사용할 수 있는 임시 인스턴스를 시작합니다. 이 인스턴스를 사용하여 호환 가능한 인스턴스를 AMI 만들 수 있습니다. AWS PCS

임시 인스턴스를 실행합니다.

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 [Instances] 를 선택한 다음 [Launch instance] 를 선택하여 새 인스턴스 시작 마법사를 엽니다.
3. (선택 사항) 이름 및 태그 섹션에서 인스턴스의 이름을 입력합니다 (예:)`PCS-AMI-instance`. 이름은 인스턴스에 리소스 태그(`Name=PCS-AMI-instance`)로 할당됩니다.
4. 애플리케이션 및 OS 이미지 섹션에서 지원되는 운영 체제 중 하나를 선택합니다. AMI
5. 인스턴스 유형(Instance type) 섹션에서 지원되는 인스턴스 유형(supported instance type)을 선택합니다.
6. 키 페어(Key pair) 섹션에서 인스턴스에 사용할 키 페어를 선택합니다.
7. 네트워크 설정 섹션에서:
    - 방화벽 (보안 그룹) 의 경우 기존 보안 그룹 선택을 선택한 다음 인스턴스에 대한 인바운드 SSH 액세스를 허용하는 보안 그룹을 선택합니다.
8. 스토리지(Storage) 섹션에서 필요에 따라 볼륨을 구성합니다. 자체 애플리케이션과 라이브러리를 설치할 수 있도록 충분한 공간을 구성해야 합니다.
9. 요약(Summary) 패널에서 인스턴스 실행(Launch instance)을 선택합니다.

2단계 — AWS PCS 에이전트 설치

에서 시작한 인스턴스를 Slurm과 함께 AWS PCS 사용하도록 구성하는 에이전트를 설치합니다.

AWS PCS 에이전트를 설치하려면

1. 시작한 인스턴스에 연결합니다. 자세한 내용은 Linux 인스턴스에 연결을 참조하세요.
2. (선택 사항) 모든 소프트웨어 패키지를 최신 상태로 유지하려면 인스턴스에서 빠른 소프트웨어 업데이트를 수행하십시오. 이 프로세스는 몇 분 정도 걸릴 수 있습니다.

   - 아마존 리눅스 2, RHEL 9, 록키 리눅스 9

     ```
     sudo yum update -y
     ```

   - 우분투 22.04

     ```
     sudo apt-get update && sudo apt-get upgrade -y
     ```

3. 인스턴스를 재부팅하고 다시 연결합니다.
4. AWS PCS에이전트 설치 파일을 다운로드합니다. 설치 파일은 압축된 tarball (`.tar.gz`) 파일로 패키지됩니다. 최신 안정 버전을 다운로드하려면 다음 명령을 사용하십시오. 대체 *`region`* 임시 인스턴스를 시작한 AWS 리전 위치 (예:) 를 사용합니다`us-east-1`.

   ```
   curl https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz -o aws-pcs-agent-v1.0.0-1.tar.gz
   ```

   이전 명령의 버전 번호를 (예:`aws-pcs-agent-v1-latest.tar.gz`) 로 `latest` 대체하여 최신 버전을 가져올 수도 있습니다.

   > **Note**
   >
   > 이는 향후 AWS PCS 에이전트 소프트웨어 릴리스에서 변경될 수 있습니다.

5. (선택 사항) AWS PCS 소프트웨어 타르볼의 신뢰성 및 무결성을 확인합니다. 이 작업을 수행하여 소프트웨어 게시자의 자격 증명을 확인하고 파일이 게시된 이후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다.

a. 의 공개 GPG 키를 다운로드하여 AWS PCS 키링으로 가져오십시오. 대체 *region* 임시 인스턴스를 시작한 AWS 리전 위치를 사용합니다. 명령에서 키 값이 반환됩니다. 키 값을 기록해 두면 다음 단계에서 사용할 수 있습니다.

```
wget https://aws-pcs-repo-public-keys-region.s3.amazonaws.com/aws-pcs-public-
key.pub && \
    gpg --import aws-pcs-public-key.pub
```

b. 다음 명령을 실행하여 GPG 키의 핑거프린트를 확인합니다.

```
gpg --fingerprint 7EEF030EDDF5C21C
```

이 명령은 다음과 동일한 핑거프린트를 반환해야 합니다.

```
1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
```

> **Important**
>
> 지문이 일치하지 않는 경우 AWS PCS 에이전트 설치 스크립트를 실행하지 마십시오. AWS Support에 문의하세요.

c. 서명 파일을 다운로드하고 AWS PCS 소프트웨어 타르볼 파일의 서명을 확인합니다. Replace *region* 임시 인스턴스를 시작한 AWS 리전 위치 (예:) 와 함께. `us-east-1`

```
wget https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-
v1.0.0-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-agent-v1.0.0-1.tar.gz.sig
```

다음과 유사하게 출력됩니다.

```
gpg: assuming signed data in './aws-pcs-agent-v1.0.0-1.tar.gz'
gpg: Signature made Thu Aug  8 18:50:19 2024 CEST
gpg:                using RSA key 4BAA531875430EB0739E6D961BA7F0AF6E34C496
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   Subkey fingerprint: 4BAA 5318 7543 0EB0 739E  6D96 1BA7 F0AF 6E34 C496
```

결과에 지문이 포함되어 `Good signature` 있고 지문이 이전 단계에서 반환된 지문과 일치하면 다음 단계로 진행하십시오.

> **Important**
>
> 지문이 일치하지 않으면 AWS PCS 소프트웨어 설치 스크립트를 실행하지 마세요. AWS Support에 문의하세요.

6. 압축 `.tar.gz` 파일에서 파일을 추출하고 추출된 디렉토리로 이동합니다.

```
tar -xf aws-pcs-agent-v1.0.0-1.tar.gz && \
    cd aws-pcs-agent
```

7. AWS PCS 소프트웨어를 설치합니다.

```
sudo ./installer.sh
```

8. AWS PCS소프트웨어 버전 파일을 확인하여 성공적으로 설치되었는지 확인합니다.

```
cat /opt/aws/pcs/version
```

다음과 유사하게 출력됩니다.

```
AGENT_INSTALL_DATE='Mon Aug 12 12:28:43 UTC 2024'
AGENT_VERSION='1.0.0'
AGENT_RELEASE='1'
```

3단계 — Slurm 설치

와 호환되는 Slurm 버전을 설치합니다. AWS PCS

Slurm을 설치하려면

1. AWS PCS소프트웨어를 설치한 동일한 임시 인스턴스에 연결합니다.
2. Slurm 설치 프로그램 소프트웨어를 다운로드합니다. Slurm 인스톨러는 압축된 tarball () 파일로 패키지됩니다. `.tar.gz` 최신 안정 버전을 다운로드하려면 다음 명령을 사용하십시오. 대체 *`region`* 임시 인스턴스 AWS 리전 (예:) 를 사용하십시오`us-east-1`.

```
curl https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-slurm/aws-pcs-
slurm-23.11-installer-23.11.9-1.tar.gz \
    -o aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz
```

이전 명령의 버전 번호를 (예:aws-pcs-slurm-23.11-installer-latest.tar.gz) 로 latest 대체하여 최신 버전을 가져올 수도 있습니다.

> **Note**
>
> 이는 향후 Slurm 설치 프로그램 소프트웨어 릴리스에서 변경될 수 있습니다.

3. (선택 사항) Slurm 설치 프로그램 타르볼의 진위 여부와 무결성을 확인하십시오. 이 작업을 수행하여 소프트웨어 게시자의 자격 증명을 확인하고 파일이 게시된 이후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다.

   a. 의 공개 GPG 키를 다운로드하여 키링으로 AWS PCS 가져오십시오. 대체 *region* 임시 인스턴스를 시작한 AWS 리전 위치를 사용합니다. 명령에서 키 값이 반환됩니다. 키 값을 기록해 두면 다음 단계에서 사용할 수 있습니다.

   ```
   wget https://aws-pcs-repo-public-keys-region.s3.amazonaws.com/aws-pcs-public-
   key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. 다음 명령을 실행하여 GPG 키의 핑거프린트를 확인합니다.

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   이 명령은 다음과 동일한 핑거프린트를 반환해야 합니다.

   ```
   1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   ```

   > **Important**
   >
   > 지문이 일치하지 않으면 Slurm 설치 스크립트를 실행하지 마십시오. AWS Support에 문의하세요.

   c. 서명 파일을 다운로드하고 Slurm 설치 프로그램 타르볼 파일의 서명을 확인합니다. Replace *region* 임시 인스턴스를 시작한 AWS 리전 위치 (예:) 와 함께. us-east-1

```
wget https://aws-pcs-repo-region.s3.amazonaws.com/aws-pcs-slurm/aws-pcs-
slurm-23.11-installer-23.11.9-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz.sig
```

다음과 유사하게 출력됩니다.

```
gpg: assuming signed data in './aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz'
gpg: Signature made Thu Aug  8 14:23:38 2024 CEST
gpg:                using RSA key 4BAA531875430EB0739E6D961BA7F0AF6E34C496
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   Subkey fingerprint: 4BAA 5318 7543 0EB0 739E  6D96 1BA7 F0AF 6E34 C496
```

결과에 지문이 포함되어 `Good signature` 있고 지문이 이전 단계에서 반환된 지문과 일치하면 다음 단계로 진행하십시오.

> **Important**
>
> 지문이 일치하지 않으면 Slurm 설치 스크립트를 실행하지 마세요. AWS Support에 문의하세요.

4. 압축 `.tar.gz` 파일에서 파일을 추출하고 추출된 디렉터리로 이동하십시오.

```
tar -xf aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz && \
    cd aws-pcs-slurm-23.11-installer
```

5. Slurm을 설치하세요. 설치 프로그램은 Slurm과 해당 종속 항목을 다운로드, 컴파일 및 설치합니다. 선택한 임시 인스턴스의 사양에 따라 몇 분 정도 걸립니다.

```
sudo ./installer.sh -y
```

6. 스케줄러 버전 파일을 확인하여 설치를 확인합니다.

```
cat /opt/aws/pcs/scheduler/slurm-23.11/version
```

다음과 유사하게 출력됩니다.

```
SLURM_INSTALL_DATE='Mon Aug 12 12:38:56 UTC 2024'
SLURM_VERSION='23.11.9'
PCS_SLURM_RELEASE='1'
```

4단계 — (선택 사항) 추가 드라이버, 라이브러리 및 응용 프로그램 소프트웨어 설치

임시 인스턴스에 추가 드라이버, 라이브러리 및 애플리케이션 소프트웨어를 설치합니다. 설치 절차는 특정 응용 프로그램 및 라이브러리에 따라 달라집니다. AWS PCS이전에 사용자 지정 AMI 버전을 구축한 적이 없다면 먼저 AMI AWS PCS 소프트웨어와 Slurm만 설치하여 빌드하고 테스트한 다음 초기 성공을 확인한 후 자체 소프트웨어 및 구성을 점진적으로 추가하는 것이 좋습니다.

예시

- 엘라스틱 패브릭 어댑터 () 소프트웨어EFA. 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 Amazon HPC EC2 워크로드 시작하기 EFA 및 MPI Amazon 워크로드를 참조하십시오.
- 아마존 Elastic File System (아마존EFS) 클라이언트. 자세한 내용은 Amazon Elastic File System 사용 설명서의 Amazon EFS 클라이언트 수동 설치를 참조하십시오.
- Lustre 클라이언트 - Amazon FSx for Lustre와 아마존 파일 캐시를 사용하기 위한 것입니다. 자세한 내용은 for Lustre의 사용 설명서에서 Lustre 클라이언트 설치를 참조하십시오. FSx
- Amazon CloudWatch 에이전트, CloudWatch 로그 및 지표 사용 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트 설치를 참조하십시오.
- AWS Neuron, trn* 및 inf* 인스턴스 유형을 사용하려면 자세한 내용은 Neuron 설명서를 참조하십시오.AWS
- NVIDIAp* 또는 g* 인스턴스 유형을 사용하기 위한 드라이버 및 CUDADCGM,

5단계 — 호환 가능한 제품 만들기 AMI AWS PCS

필수 소프트웨어 구성 요소를 설치한 후 AWS PCS 컴퓨팅 노드 그룹에서 인스턴스를 시작하는 데 재사용할 수 있는 구성 요소를 생성합니다. AMI

임시 AMI 인스턴스에서 생성하려면

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 생성한 임시 인스턴스를 선택합니다. [액션], [이미지], [이미지 만들기] 를 선택합니다.

4. [이미지 생성(Create image)]에서 다음을 수행합니다.

   a. 이미지 이름에는 을 설명하는 이름을 입력합니다. AMI

   b. (선택 사항) 이미지 설명에는 용도에 대한 간략한 설명을 입력합니다. AMI

   c. 이미지 생성을 선택합니다.

5. 탐색 창에서 을 선택합니다 AMIs.

6. 목록에서 AMI 생성한 항목을 찾습니다. 상태가 [보류 중] 에서 [사용 가능] 으로 변경될 때까지 기다린 다음 AWS PCS 컴퓨팅 노드 그룹과 함께 사용하십시오.

6단계 — AWS PCS 컴퓨팅 노드 AMI 그룹과 함께 사용자 지정 사용

새 AWS PCS 컴퓨팅 노드 그룹 또는 기존 컴퓨팅 노드 AMI 그룹과 함께 사용자 지정을 사용할 수 있습니다.

New compute node group

사용자 지정을 사용하려면 AMI

1. AWS PCS콘솔을 엽니다.
2. 탐색 창에서 클러스터(Clusters)를 선택합니다.
3. 사용자 AMI 지정을 사용할 클러스터를 선택한 다음 Compute 노드 그룹을 선택합니다.
4. 새 컴퓨팅 노드 그룹을 생성합니다. 자세한 내용은 에서 컴퓨팅 노드 그룹 생성 AWS PCS 단원을 참조하십시오. AMIID에서 AMI 사용하려는 사용자 지정의 이름 또는 ID를 검색합니다. 컴퓨팅 노드 그룹 구성을 완료한 다음 컴퓨팅 노드 그룹 생성을 선택합니다.
5. (선택 사항) AMI 지원되는 인스턴스 시작을 확인합니다. 컴퓨팅 노드 그룹에서 인스턴스를 시작합니다. 단일 정적 인스턴스를 포함하도록 컴퓨팅 노드 그룹을 구성하거나 컴퓨팅 노드 그룹을 사용하는 대기열에 작업을 제출하여 이 작업을 수행할 수 있습니다.

   a. 인스턴스에 새 컴퓨팅 노드 그룹 ID 태그가 지정될 때까지 Amazon EC2 콘솔을 확인합니다. 이에 대한 자세한 내용은 다음을 참조하십시오에서 컴퓨팅 노드 그룹 인스턴스 찾기 AWS PCS.

   b. 인스턴스가 시작되고 부트스트랩 프로세스를 완료하면 예상대로 AMI 사용하고 있는지 확인하십시오. 이렇게 하려면 인스턴스를 선택한 다음 세부 정보에서 AMI ID를 검사하십시오. 컴퓨팅 노드 그룹 AMI 설정에서 구성한 것과 일치해야 합니다.

   c. (선택 사항) 컴퓨팅 노드 그룹 조정 구성을 원하는 값으로 업데이트하십시오.

Existing compute node group

사용자 지정을 사용하려면 AMI

1. AWS PCS콘솔을 엽니다.
2. 탐색 창에서 클러스터(Clusters)를 선택합니다.
3. 사용자 AMI 지정을 사용할 클러스터를 선택한 다음 Compute 노드 그룹을 선택합니다.
4. 구성하려는 노드 그룹을 선택하고 편집을 선택합니다. AMIID에서 AMI 사용하려는 사용자 지정의 이름 또는 ID를 검색합니다. 컴퓨팅 노드 그룹 구성을 완료한 다음 업데이트를 선택합니다. 컴퓨팅 노드 그룹에서 시작된 새 인스턴스는 업데이트된 AMI ID를 사용합니다. 기존 인스턴스는 AWS PCS 교체할 AMI 때까지 이전 인스턴스를 계속 사용합니다. 자세한 내용은 AWS PCS컴퓨팅 노드 그룹 업데이트 단원을 참조하십시오.
5. (선택 사항) AMI 지원되는 인스턴스 시작을 확인합니다. 컴퓨팅 노드 그룹에서 인스턴스를 시작합니다. 단일 정적 인스턴스를 포함하도록 컴퓨팅 노드 그룹을 구성하거나 컴퓨팅 노드 그룹을 사용하는 대기열에 작업을 제출하여 이 작업을 수행할 수 있습니다.

   a. 인스턴스에 새 컴퓨팅 노드 그룹 ID 태그가 지정될 때까지 Amazon EC2 콘솔을 확인합니다. 이에 대한 자세한 내용은 다음을 참조하십시오에서 컴퓨팅 노드 그룹 인스턴스 찾기 AWS PCS.

   b. 인스턴스가 시작되고 부트스트랩 프로세스를 완료하면 예상대로 AMI 사용하고 있는지 확인하십시오. 이렇게 하려면 인스턴스를 선택한 다음 세부 정보에서 AMI ID를 검사하십시오. 컴퓨팅 노드 그룹 AMI 설정에서 구성한 것과 일치해야 합니다.

   c. (선택 사항) 컴퓨팅 노드 그룹 조정 구성을 원하는 값으로 업데이트하십시오.

7단계 — 임시 인스턴스 종료

의도한 대로 AMI 작동하는지 확인한 후 임시 인스턴스를 종료하여 요금 발생을 중단할 수 있습니다. AWS PCS

임시 인스턴스 종료

1. Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 생성한 임시 인스턴스를 선택하고 작업, 인스턴스 상태, 인스턴스 종료를 선택합니다.
4. 확인 메시지가 표시되면 [Terminate] 를 선택합니다.

맞춤 AMIs 제작을 위한 소프트웨어 설치 프로그램 AWS PCS

AWS 인스턴스에 AWS PCS 소프트웨어를 설치할 수 있는 다운로드 가능한 파일을 제공합니다. AWS 관련 버전의 Slurm 및 해당 종속 항목을 다운로드, 컴파일 및 설치할 수 있는 소프트웨어도 제공합니다. 이 지침을 사용하여 함께 AWS PCS 사용할 사용자 정의를 만들거나 자체 방법을 AMIs 사용할 수 있습니다.

목차

AWS PCS소프트웨어 설치 프로그램

AWS PCS소프트웨어 설치 프로그램은 인스턴스 부트스트랩 프로세스 AWS PCS 중에 사용할 인스턴스를 구성합니다. 사용자 정의에 소프트웨어를 설치하려면 AWS-제공된 설치 프로그램을 사용해야 합니다. AWS PCS AMI

Slurm 인스톨러

Slurm 설치 프로그램은 관련 버전의 Slurm 및 해당 종속 항목을 다운로드, 컴파일 및 설치합니다. Slurm 설치 프로그램을 사용하여 사용자 지정 버전을 빌드할 수 있습니다. AMIs AWS PCS Slurm 설치 프로그램이 제공하는 소프트웨어 구성과 일치하는 경우 자체 메커니즘을 사용할 수도 있습니다.

AWS제공된 소프트웨어는 다음을 설치합니다.

- 요청된 메이저 및 유지 관리 버전 (현재 버전 23.11.8) 에서 Slurm - 라이선스 2 GPL
  - Slurm은 다음과 같이 설정되어 빌드됩니다. `--sysconfdir /etc/slurm`
  - Slurm은 다음과 같은 옵션으로 제작되었습니다. `--enable-pam --without-munge`
  - Slurm은 다음과 같은 옵션으로 제작됩니다. `--sharedstatedir=/run/slurm/`
  - Slurm은 다음과 같은 지원을 통해 제작되었습니다. PMIX JWT
  - Slurm은 다음에 설치됩니다. `/opt/aws/pcs/schedulers/slurm-23.11`

- 오픈 PMIX (버전 4.2.6) — 라이선스
  - PMIXOpen은 의 하위 디렉토리로 설치됩니다. `/opt/aws/pcs/scheduler/`
- libjwt (버전 1.15.3) — 라이선스 -2.0 MPL
  - libjwt는 다음의 하위 디렉토리로 설치됩니다. `/opt/aws/pcs/scheduler/`

AWS제공된 소프트웨어는 다음과 같이 시스템 구성을 변경합니다.

- 빌드에서 생성된 Slurm `systemd` 파일은 파일 이름과 `/etc/systemd/system/` 함께 복사됩니다. `slurmd-23.11.service`
- 존재하지 않는 경우 /of를 사용하여 Slurm 사용자 및 그룹 (`slurm:slurm`) 이 생성됩니다. UID GID `401`
- Amazon Linux 2 및 Rocky Linux 9에서 설치하면 Slurm 또는 해당 종속성을 빌드하는 데 필요한 소프트웨어를 설치할 수 있는 EPEL 리포지토리가 추가됩니다.
- 설치 시 RHEL9 Slurm 또는 해당 `epel-release-latest-9` 종속성을 `fedoraproject` 빌드하는 데 필요한 소프트웨어를 `codeready-builder-for-rhel-9-rhui-rpms` 활성화하고 설치할 수 있습니다.

지원되는 운영 체제

AWS PCS소프트웨어 및 Slurm 설치 프로그램은 다음 운영 체제를 지원합니다.

- Amazon Linux 2
- RedHat 엔터프라이즈 리눅스 9
- Rocky Linux 9
- Ubuntu 22.04

> **Note**
>
> AWS Deep Learning AMIs (DLAMI) Amazon Linux 2 및 우분투 22.04를 기반으로 하는 버전은 AWS PCS 소프트웨어 및 Slurm 설치 프로그램과 호환되어야 합니다. 자세한 내용은 개발자 안내서의 사용자 버전 선택을 참조하십시오. DLAMI AWS Deep Learning AMIs

지원되는 인스턴스 유형

AWS PCS소프트웨어 및 Slurm 설치 프로그램은 지원되는 운영 체제 중 하나를 실행할 수 있는 모든 x86_64 또는 arm64 인스턴스 유형을 지원합니다.

지원되는 Slurm 버전

다음과 같은 주요 버전의 Slurm이 지원됩니다.

- 슬럼 23.11

체크섬을 사용하여 인스톨러 확인하기

SHA256체크섬을 사용하여 설치 프로그램 타르볼 (.tar.gz) 파일을 확인할 수 있습니다. 이 작업을 수행하여 소프트웨어 게시자의 자격 증명을 확인하고 애플리케이션이 게시된 이후 변경되거나 손상되지 않았는지 확인하는 것이 좋습니다.

타르볼을 확인하려면

SHA256체크섬에 sha256sum 유틸리티를 사용하고 타르볼 파일 이름을 지정합니다. 타르볼 파일을 저장한 디렉터리에서 명령을 실행해야 합니다.

- SHA256

```
$ sha256sum tarball_filename.tar.gz
```

이 명령은 다음 형식의 체크섬 값을 반환해야 합니다.

```
checksum_value tarball_filename.tar.gz
```

명령에서 반환된 체크섬 값을 다음 표에 제공된 체크섬 값과 비교하십시오. 체크섬이 일치하면 설치 스크립트를 실행해도 안전합니다.

> **Important**
>
> 체크섬이 일치하지 않으면 설치 스크립트를 실행하지 마세요. AWS Support에 문의하십시오.

예를 들어, 다음 명령은 Slurm 23.11.9 SHA256 타르볼의 체크섬을 생성합니다.

```
$ sha256sum aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz
```

출력 예제:

```
1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8 aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz
```

다음 표에는 최신 버전의 설치 프로그램에 대한 체크섬이 나와 있습니다. Replace *us-east-1* 사용하는 AWS 리전 곳과 함께 사용하십시오. AWS PCS

설치 관리자	다운로드 URL	SHA256체크섬
슬럼 23.11.9	`https://aws-pcs-repo- us-east-1 .s3.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz`	`1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8`
AWS PCS에이전트 1.0.0	`https://aws-pcs-repo- us-east-1 .s3.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz`	`d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042144b134ce0`

슬럼 버전 입력 AWS PCS

SchedMD는 새로운 기능, 최적화 및 보안 패치를 통해 Slurm을 지속적으로 개선합니다. SchedMD는 정기적으로 새 메이저 버전을 출시하며 언제든지 최대 3개의 버전을 지원할 계획입니다. AWS PCS처음에는 Slurm 23.11을 지원합니다. 새 버전이 출시된 후 Slurm 메이저 버전을 업그레이드할 수 있습니다. AWS PCSSlurm 컨트롤러를 패치 버전으로 자동 업데이트하도록 설계되었습니다.

SchedMD가 특정 메이저 버전에 대한 지원을 종료하면 해당 메이저 버전에 대한 AWS PCS 지원도 종료됩니다. AWS PCSSlurm 메이저 버전의 수명이 거의 다한 경우 사전 알림을 보내 고객이 클러스터를 지원되는 새 버전으로 업그레이드해야 하는 시기를 알 수 있도록 합니다.

지원되는 최신 Slurm 버전을 사용하여 클러스터를 배포하고 최신 개선 사항 및 개선 사항에 액세스하는 것이 좋습니다.

Slurm 버전에 대한 자주 묻는 질문

Slurm 버전은 얼마 동안 AWS PCS 지원되나요?

AWS PCS메이저 버전의 SchedMD 지원 주기를 따릅니다. AWS PCS한 번에 최대 3개의 메이저 버전을 지원합니다. SchedMD가 새 메이저 버전을 출시한 후 가장 오래된 지원 버전을 AWS PCS 폐기합니다. AWS PCS가능한 한 빨리 Slurm의 새 메이저 버전을 릴리스하지만 SchedMD 릴리스와 출시 사이에 지연이 있을 수 있습니다. AWS PCS

Slurm 버전의 Support 수명 종료 (EOSL) 에 대한 AWS PCS 알림은 언제 받나요?

AWS PCS날짜 이전에 미리 정해진 빈도로 여러 번 알려줍니다. EOSL

Slurm 버전이 다가오면 어떻게 해야 하나요? EOSL

안전하고 지원되는 환경을 EOSL 유지하려면 먼저 Slurm 버전을 업데이트해야 합니다.

Slurm의 새 메이저 버전을 사용하도록 클러스터를 업데이트하려면 어떻게 해야 하나요?

Slurm 버전을 업데이트하려면 새 클러스터를 만들어야 합니다. 또한 의 해당 AWS PCS 소프트웨어로 AMI 업그레이드하고 이를 사용하여 새 클러스터의 컴퓨팅 노드 그룹을 생성해야 합니다.

내 클러스터는 어떻게 새로운 Slurm 패치 버전 릴리스를 받게 되나요?

AWS PCS패치를 자동으로 적용하여 Slurm의 일반적인 취약성 및 노출을 해결하도록 설계되었습니다 (). CVEs AWS PCS내부 서비스 소유 계정에서 실행되는 클러스터 컨트롤러에 패치를 적용합니다. AWS Management Console 또는 AWS PCS API 작업을 사용하여 내 EC2 인스턴스에 패치를 설치해야 합니다. AWS 계정

날짜까지 Slurm을 업데이트하지 않으면 어떻게 되나요? EOSL

AWS PCS지원되지 않는 Slurm 버전이 있는 클러스터를 중지하도록 설계되었습니다. Slurm 메이저 버전의 클러스터 컨트롤러와 컴퓨팅 노드 그룹에 설치된 AWS PCS 소프트웨어를 업데이트해야 합니다.

지원하는 Slurm 버전은 몇 개입니까? AWS PCS

AWS PCS현재 및 2개의 이전 메이저 버전을 포함하여 언제든지 최대 3개의 주요 Slurm 버전을 지원합니다.

어떤 Slurm 버전 업데이트를 적용해야 하나요?

클러스터의 모든 구성 요소에서 동일한 메이저 버전을 사용하고 최신 패치가 출시되는 즉시 설치하는 것이 좋습니다. 컴퓨팅 노드 그룹의 AMIs 경우 클러스터 컨트롤러의 Slurm 버전과 호환되는 Slurm 소프트웨어 버전을 사용해야 합니다. 의 Slurm 메이저 버전은 클러스터 컨트롤러에 있는 Slurm 메이저 버전의 두 버전 AMIs 이내여야 합니다. 클러스터에서 실행 중인 EC2 인스턴스에 설치된 Slurm 버전은 클러스터 컨트롤러의 Slurm 버전보다 최신일 수 없습니다. AMI 클러스터에 대한 지원을 유지하려면 지원되는 소프트웨어 버전을 AMIs 사용해야 합니다. AWS PCS

Slurm 메이저 버전을 업데이트했는데 컴퓨팅 노드 AMI 그룹용 Slurm 소프트웨어를 구형 Slurm 소프트웨어를 사용하면 어떻게 되나요?

새 Slurm AWS PCS 기능을 사용하려면 소프트웨어를 동일한 버전으로 업데이트해야 합니다. 완전한 AWS PCS 지원을 위해서는 모든 Slurm 구성 요소가 지원되는 버전을 사용해야 합니다. 요약하면 다음과 같습니다.

- 클러스터 컨트롤러와 모든 구성 요소 (AWS PCS패키지) 가 모두 지원되는 버전을 사용할 경우 완전한 지원을 제공할 수 있습니다 AWS 계정 .
- AWS PCS컨트롤러의 Slurm 버전에 도달하면 클러스터를 중지하도록 설계되었습니다. EOSL
- Slurm 버전의 구성 요소를 사용할 수 AWS 계정 있는 경우 클러스터는 지원되지 않습니다. EOSL

클러스터의 구성 요소를 어떤 순서로 업데이트해야 하나요?

최신 Slurm 버전을 사용하기 전에 클러스터 컨트롤러의 Slurm 버전을 업데이트해야 합니다. AMI 를 사용하려면 컴퓨팅 노드 그룹을 업데이트합니다. AMI AWS PCSAMI를 사용하여 컴퓨팅 노드 그룹에서 새 EC2 인스턴스를 시작합니다. AWS PCS실행 중인 작업이 있는 기존 EC2 인스턴스를 업데이트하지 않으며, 작업이 완료된 후 해당 인스턴스를 종료하도록 설계되었습니다. AWS PCS

Slurm 버전에 대한 확장 지원을 AWS PCS 제공하나요?

아니요. 추가 비용 및 제공되는 특정 지원 범위를 포함하여 확장 지원 옵션에 대한 자세한 정보를 안내해 드리겠습니다.

AWS 병렬 컴퓨팅 서비스의 보안

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 귀사 간의 공동 책임입니다. AWS 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. AWS 병렬 컴퓨팅 서비스에 적용되는 규정 준수 프로그램에 대해 자세히 알아보려면 규정 준수 프로그램별 범위 내 AWSAWS 서비스 규정 준수 프로그램별 참조하십시오.
- 클라우드에서의 보안 — 사용하는 AWS 서비스에 따라 책임이 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 공동 책임 모델을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 AWS PCS 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 AWS PCS 충족하도록 구성하는 방법을 보여줍니다. 또한 AWS PCS 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

- AWS 병렬 컴퓨팅 서비스의 데이터 보호
- 인터페이스 엔드포인트 ()AWS PrivateLink를 사용하여 AWS 병렬 컴퓨팅 서비스에 액세스
- AWS 병렬 컴퓨팅을 위한 Identity 및 Access Management 서비스
- 병렬 컴퓨팅 서비스에 대한 AWS 규정 준수 검증
- AWS 병렬 컴퓨팅 서비스의 탄력성
- AWS 병렬 컴퓨팅 서비스의 인프라 보안
- 병렬 컴퓨팅 서비스의 취약성 분석 및 관리 AWS
- 교차 서비스 혼동된 대리인 방지
- AWS 병렬 컴퓨팅 서비스의 보안 모범 사례

AWS 병렬 컴퓨팅 서비스의 데이터 보호

AWS 공동 책임 모델 AWS 병렬 컴퓨팅 서비스의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시를 참조하십시오FAQ. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 공동 책임 모델 및AWS 보안 GDPR 블로그의 블로그 게시물을 참조하십시오.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 개별 사용자에게 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM) 를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 다단계 인증 (MFA) 을 사용하십시오.
- SSLTLS/를 사용하여 AWS 리소스와 통신하세요. TLS1.2가 필요하고 TLS 1.3을 권장합니다.
- API를 사용하여 사용자 활동 로깅을 설정합니다 AWS CloudTrail.
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API an을 AWS 통해 액세스할 때 FIPS 140-3개의 검증된 암호화 모듈이 필요한 경우 엔드포인트를 사용하십시오. FIPS 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준 () 140-3을 참조하십시오. FIPS

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS CLI, AWS PCS 또는 다른 사용자와 AWS 서비스 함께 작업하는 경우가 포함됩니다. AWS SDKs 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL a를 제공하는 경우 해당 서버에 대한 요청을 URL 검증하기 위해 자격 증명 정보를 에 포함하지 않는 것이 좋습니다.

저장 중 암호화

AWS Management Console, AWS CLI AWS PCSAPI, 또는 를 사용하여 AWS 병렬 컴퓨팅 서비스 (AWS PCS) 클러스터를 만들면 저장된 데이터에 대해 기본적으로 암호화가 AWS SDKs 활성화됩니다. AWS PCSAWS 소유 KMS 키를 사용하여 저장된 데이터를 암호화합니다. 자세한 내용은 AWS

KMS 개발자 안내서의 고객 AWS 키 및 키를 참조하십시오. 클러스터 암호는 Secrets Manager 관리 KMS 키에 AWS Secrets Manager 저장되고 이를 통해 암호화됩니다. 자세한 내용은 에서 클러스터 시크릿 사용하기 AWS PCS 단원을 참조하십시오.

AWS PCS클러스터에는 다음과 같은 데이터가 저장되어 있습니다.

- 스케줄러 상태 - 클러스터에서 실행 중인 작업 및 프로비저닝된 노드에 대한 데이터를 포함합니다. 이 데이터는 Slurm이 사용자 정의에 따라 유지되는 데이터입니다. `StateSaveLocation` `slurm.conf` 자세한 내용은 Slurm 설명서의 StateSaveLocation설명을 참조하십시오. AWS PCS작업이 완료된 후 작업 데이터를 삭제합니다.
- 스케줄러 인증 암호 - 이를 AWS PCS 사용하여 클러스터의 모든 스케줄러 통신을 인증합니다.

스케줄러 상태 정보의 경우, 데이터 및 메타데이터를 파일 시스템에 쓰기 전에 AWS PCS 자동으로 암호화합니다. 암호화된 파일 시스템은 저장된 데이터에 대해 업계 표준 AES -256 암호화 알고리즘을 사용합니다.

전송 중 암호화

연결 시 서명 버전 4 서명 프로세스를 통한 TLS 암호화를 AWS PCS API 사용하며, 이는 AWS Command Line Interface ()AWS CLI또는 사용 여부와 관계없이 이루어집니다. AWS SDKs 자세한 내용은 AWS Identity and Access Management 사용 설명서의 서명 AWS API 요청을 참조하십시오. AWS 연결에 API 사용하는 보안 자격 증명에 대한 IAM 정책을 통해 액세스 제어를 관리합니다.

AWS PCS다른 AWS 서비스에 TLS 연결하는 데 사용합니다.

Slurm 클러스터 내에서 스케줄러는 모든 스케줄러 통신에 대한 인증을 제공하는 `auth/slurm` 인증 플러그인으로 구성됩니다. Slurm은 통신을 위한 애플리케이션 수준의 암호화를 제공하지 않으므로, 클러스터 인스턴스를 통해 흐르는 모든 데이터는 로컬로 유지되므로 해당 인스턴스가 전송 중 EC2 VPC 암호화를 지원하는 경우 VPC 암호화가 적용됩니다. 자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 전송 중 암호화를 참조하십시오. 컨트롤러 (서비스 계정에서 프로비저닝) 와 계정의 클러스터 노드 간 통신은 암호화됩니다.

키 관리

AWS PCSAWS 소유한 KMS 키를 사용하여 데이터를 암호화합니다. 자세한 내용은 AWS KMS 개발자 안내서의 고객 AWS 키 및 키를 참조하십시오. 클러스터 암호는 Secrets Manager 관리 KMS 키에 AWS Secrets Manager 저장되고 이를 통해 암호화됩니다. 자세한 내용은 에서 클러스터 시크릿 사용하기 AWS PCS 단원을 참조하십시오.

인터네트워크 트래픽 개인 정보 보호

AWS PCS클러스터의 컴퓨팅 리소스는 고객 VPC 계정에서 1 내에 있습니다. 따라서 클러스터 내의 모든 내부 AWS PCS 서비스 트래픽은 AWS 네트워크 내에 머물며 인터넷을 통해 이동하지 않습니다. 사용자와 AWS PCS 노드 간의 통신은 인터넷을 통해 이루어질 수 있으므로 Systems Manager를 사용하여 SSH 노드에 연결하는 것이 좋습니다. 자세한 내용은 AWS Systems Manager무엇입니까를 참조하십시오. AWS Systems Manager 사용 설명서에서.

다음 제품을 사용하여 온프레미스 네트워크를 연결할 AWS수도 있습니다.

- AWS Site-to-Site VPN. 자세한 내용은 AWS Site-to-Site VPN무엇입니까를 참조하십시오. AWS Site-to-Site VPN 사용 설명서에서.
- An AWS Direct Connect. 자세한 내용은 AWS Direct Connect무엇입니까를 참조하십시오. AWS Direct Connect 사용 설명서에서.

에 AWS PCS API 액세스하여 서비스에 대한 관리 작업을 수행할 수 있습니다. 귀하와 사용자는 Slurm 엔드포인트 포트에 액세스하여 스케줄러와 직접 상호 작용합니다.

트래픽 암호화 API

에 AWS PCS API 액세스하려면 클라이언트가 전송 계층 보안 (TLS) 1.2 이상을 지원해야 합니다. TLS1.2가 필요하고 TLS 1.3을 권장합니다. 또한 클라이언트는 Ephemeral Diffie-Hellman () 또는 Emphemeral Diffie-Hellman Ephemeral (PFS) 또는 타원 곡선 디피-헬만 Ephemeral () 과 같은 완벽한 순방향 비밀성을 갖춘 암호 제품군을 지원해야 합니다. DHE ECDHE Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다. 또한 액세스 키 ID 및 보안 주체와 연결된 보안 액세스 키를 사용하여 요청에 서명해야 합니다. IAM AWS Security Token Service (AWS STS) 를 사용하여 요청에 서명하기 위한 임시 보안 자격 증명을 생성할 수도 있습니다.

데이터 트래픽 암호화

전송 중인 데이터는 스케줄러 엔드포인트에 액세스하는 지원되는 EC2 인스턴스에서, 그리고 스케줄러 엔드포인트 내에서 ComputeNodeGroup 인스턴스 간에 암호화할 수 있습니다. AWS 클라우드자세한 내용은 전송 중 암호화 단원을 참조하십시오.

인터페이스 엔드포인트 ()AWS PrivateLink를 사용하여 AWS 병렬 컴퓨팅 서비스에 액세스

를 AWS PrivateLink 사용하여 병렬 컴퓨팅 VPC 서비스와 AWS 병렬 컴퓨팅 서비스 (AWS PCS) 간에 비공개 연결을 만들 수 있습니다. 인터넷 게이트웨이, NAT 장치VPC, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고도 마치 집에 있는 AWS PCS 것처럼 액세스할 수 있습니다. 인스턴스에 액세스하는 데 퍼블릭 IP 주소가 VPC 필요하지 않습니다 AWS PCS.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 대상 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다. AWS PCS

자세한 내용은 가이드의 액세스를 참조하십시오 AWS 서비스 . AWS PrivateLinkAWS PrivateLink

에 대한 고려 사항 AWS PCS

에 대한 AWS PCS 인터페이스 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 인터페이스 VPC 엔드포인트를 사용한 AWS 서비스 액세스를 검토하세요.

AWS PCS인터페이스 엔드포인트를 통해 모든 API 작업에 대한 호출을 지원합니다.

인터넷에 직접 액세스할 수 VPC 없는 경우 컴퓨팅 노드 그룹 인스턴스가 AWS PCS `RegisterComputeNodeGroupInstance`API작업을 호출할 수 있도록 VPC 엔드포인트를 구성해야 합니다.

에 대한 인터페이스 엔드포인트를 생성하십시오. AWS PCS

Amazon VPC 콘솔 또는 AWS Command Line Interface (AWS CLI) AWS PCS 를 사용하기 위한 인터페이스 엔드포인트를 생성할 수 있습니다. 자세한 내용은 AWS PrivateLink 설명서의 인터페이스 엔드포인트 생성을 참조하십시오.

다음 서비스 이름을 AWS PCS 사용하기 위한 인터페이스 엔드포인트를 생성하십시오.

```
com.amazonaws.region.pcs
```

Replace *region* 의 AWS 리전 ID를 사용하여 엔드포인트를 생성하십시오 (예:`us-east-1`.

인터페이스 엔드포인트에서 DNS 프라이빗을 활성화하면 기본 지역 DNS 이름을 AWS PCS 사용하도록 API 요청할 수 있습니다. 예: `pcs.us-east-1.amazonaws.com`.

엔드포인트의 엔드포인트 정책 생성

엔드포인트 정책은 인터페이스 엔드포인트에 연결할 수 있는 IAM 리소스입니다. 기본 엔드포인트 정책은 인터페이스 엔드포인트를 AWS PCS 통한 전체 액세스를 허용합니다. AWS PCS에서 허용된 액세스를 제어하려면 사용자 지정 엔드포인트 정책을 인터페이스 엔드포인트에 연결하십시오. VPC

엔드포인트 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 주체 (AWS 계정, IAM 사용자, IAM 역할)
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

예: 작업에 대한 VPC 엔드포인트 정책 AWS PCS

다음은 사용자 지정 엔드포인트 정책의 예입니다. 이 정책을 인터페이스 엔드포인트에 연결하면 클러스터의 모든 보안 주체에 대해 나열된 AWS PCS 작업에 대한 액세스 권한이 지정된 클러스터에 부여됩니다.*cluster-id*. 교체 *region* AWS 리전 클러스터의 ID (예:) 로 입력합니다`us-east-1`. Replace *account-id* 클러스터 AWS 계정 번호와 함께

```
{
    "Statement": [
            {
                "Action": [
                "pcs:CreateCluster",
                "pcs:ListClusters",
                "pcs:DeleteCluster",
                "pcs:GetCluster",
                ],
                "Effect": "Allow",
                "Principal": "*",
                "Resource": [
                    "arn:aws:pcs:region:account-id:cluster/cluster-id*"
                ]
            }
        ]
}
```

AWS 병렬 컴퓨팅을 위한 Identity 및 Access Management 서비스

AWS Identity and Access Management (IAM) 는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와줍니다. IAM관리자는 AWS PCS 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. IAM추가 비용 없이 사용할 AWS 서비스 수 있습니다.

주제

고객

AWS Identity and Access Management (IAM) 를 사용하는 방법은 수행하는 작업에 따라 다릅니다. AWS PCS

서비스 사용자 - AWS PCS 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS PCS 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. 에서 AWS PCS 기능에 액세스할 수 없는 경우 을 참조하십시오AWS 병렬 컴퓨팅 서비스 ID 및 액세스 문제 해결.

서비스 관리자 — 회사에서 AWS PCS 리소스를 담당하고 있다면 전체 액세스 권한이 있을 것입니다 AWS PCS. 서비스 사용자가 액세스해야 하는 AWS PCS 기능과 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음 IAM 관리자에게 서비스 사용자의 권한을 변경해 달라는 요청을 제출해야 합니다. 이

페이지의 정보를 검토하여 의 기본 개념을 IAM 이해하십시오. 회사에서 사용하는 방법에 대한 자세한 내용은 IAM AWS PCS 을 참조하십시오AWS 병렬 컴퓨팅 서비스의 작동 방식 IAM.

IAM관리자 — IAM 관리자인 경우 액세스 관리를 위한 정책을 작성하는 방법에 대해 자세히 알아보는 것이 좋습니다 AWS PCS. 에서 IAM 사용할 수 있는 AWS PCS ID 기반 정책의 예를 보려면 을 참조하십시오. 병렬 컴퓨팅 서비스의 ID 기반 정책 예제 AWS

ID를 통한 인증

인증은 ID 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM사용자로서 또는 역할을 위임하여 인증 (로그인 AWS) 을 받아야 합니다. AWS 계정 루트 사용자 IAM

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAMID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인하는 경우 관리자는 이전에 역할을 사용하여 ID 페더레이션을 설정했습니다. IAM 페더레이션을 AWS 사용하여 액세스하는 경우 간접적으로 역할을 수임하는 것입니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호로 서명할 수 있는 소프트웨어 개발 키트 (SDKCLI) 와 명령줄 인터페이스 () 가 AWS 제공됩니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 사용 IAM설명서의 AWS API요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, 계정 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 AWS 것을 권장합니다. 자세한 내용은 사용 설명서의 다단계 인증 및 사용 AWS IAM Identity Center 설명서의 다단계 인증 사용 (MFA) 을 IAM 참조하십시오. AWS

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 작업의 전체 목록은 사용 설명서의 루트 사용자 자격 증명이 필요한 작업을 참조하십시오. IAM

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS Directory Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAMIdentity Center에서 사용자 및 그룹을 만들거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 AWS 계정 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. ID 센터에 대한 자세한 내용은 IAM ID 센터란 IAM 무엇입니까? 를 참조하십시오. AWS IAM Identity Center 사용 설명서에서.

IAM 사용자 및 그룹

IAM사용자란 단일 사용자 또는 애플리케이션에 대한 특정 권한을 가진 사용자 내의 ID입니다. AWS 계정 가능하면 암호 및 액세스 키와 같은 장기 자격 증명을 가진 IAM 사용자를 만드는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 특정 사용 사례에서 IAM 사용자의 장기 자격 증명이 필요한 경우에는 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 사용 설명서의 장기 자격 증명이 필요한 사용 사례에 대한 정기적인 액세스 키 IAM 교체를 참조하십시오.

IAM그룹은 IAM 사용자 컬렉션을 지정하는 ID입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 이름을 지정한 IAMAdmins그룹을 만들고 해당 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세히 알아보려면 사용 설명서의 역할 대신 IAM 사용자를 만드는 시기를 참조하십시오. IAM

IAM역할

IAM역할은 특정 권한을 AWS 계정 가진 사용자 내의 ID입니다. IAM사용자와 비슷하지만 특정인과 관련이 있는 것은 아닙니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI or AWS API 작업을 호출하거나 사용자 지정을 사용하여 역할을 수임할 수 URL 있습니다. 역할 사용 방법에 대한 자세한 내용은 사용 IAM설명서의 IAM역할 사용을 참조하십시오.

IAM임시 자격 증명이 있는 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션을 위한 역할에 대한 자세한 내용은 IAM사용 설명서의 타사 ID 제공자를 위한 역할 생성을 참조하십시오. IAMIdentity Center를 사용하는 경우 권한 집합을 구성합니다. ID가 인증된 후 액세스할 수 있는 대상을 제어하기 위해 IAM Identity Center는 권한 집합을 역할의 상관 관계와 연결합니다. IAM 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할이 역할을 맡아 특정 작업에 대해 일시적으로 다른 권한을 부여받을 수 있습니다. IAM
- 계정 간 액세스 - IAM 역할을 사용하여 다른 계정의 사용자 (신뢰할 수 있는 사용자) 가 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 하지만 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책 간의 차이점을 알아보려면 사용 설명서의 교차 계정 리소스 액세스를 참조하십시오. IAM IAM
- 서비스 간 액세스 — 일부는 다른 기능을 AWS 서비스 사용합니다. AWS 서비스예를 들어, 서비스를 호출하면 해당 서비스가 Amazon에서 애플리케이션을 EC2 실행하거나 Amazon S3에 객체를 저장하는 것이 일반적입니다. 서비스는직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS요청 시 적용되는 정책 세부 정보는 전달 액세스 세션을 참조하십시오.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 간주하는 IAM 역할입니다. IAM관리자는 내부에서 IAM 서비스 역할을 만들고, 수정하고, 삭제할 수 있습니다. 자세한 내용은 사용 설명서의 역할 만들기를 참조하여 권한을 위임하십시오IAM. AWS 서비스
  - 서비스 연결 역할 - 서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.

- Amazon에서 실행 중인 애플리케이션 EC2 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS API 요청을 보내는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS CLI EC2인스턴스 내에 액세스 키를 저장하는 것보다 이 방법이 더 좋습니다. EC2인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 만들어야 합니다. 인스턴스 프로필에는 역할이 포함되며, 이를 통해 EC2 인스턴스에서 실행 중인 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 내용은 사용 설명서의 IAM역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 IAM 참조하십시오.

IAM역할을 사용할지 IAM 사용자를 사용할지 알아보려면 사용 설명서의 IAM 역할 생성 시기 (사용자 대신) 를 IAM참조하십시오.

정책을 사용한 액세스 관리

정책을 만들고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON정책 문서의 구조 및 내용에 대한 자세한 내용은 IAM사용 설명서의 JSON 정책 개요를 참조하십시오.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. IAM관리자는 IAM 정책을 생성하여 필요한 리소스에서 작업을 수행할 수 있는 권한을 사용자에게 부여할 수 있습니다. 그러면 관리자가 역할에 IAM 정책을 추가할 수 있으며, 사용자는 역할을 수임할 수 있습니다.

IAM정책은 작업을 수행하는 데 사용하는 방법에 관계없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 에서 역할 정보를 가져올 수 AWS API 있습니다.

보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 만드는 방법을 알아보려면 사용 설명서의 IAM정책 생성을 참조하십시오. IAM

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할

수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책과 인라인 정책 중에서 선택하는 방법을 알아보려면 IAM사용 설명서의 관리형 정책과 인라인 정책 중 선택을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 IAM 정책에서는 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록 () ACLs

액세스 제어 목록 (ACLs) 은 리소스에 액세스할 수 있는 권한을 가진 주체 (계정 구성원, 사용자 또는 역할) 를 제어합니다. ACLs정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 JSON 비슷합니다.

지원하는 서비스의 VPC 예로는 Amazon S3와 Amazon이 ACLs 있습니다. AWS WAF자세한 내용은 Amazon 심플 스토리지 서비스 개발자 안내서의 액세스 제어 목록 (ACL) 개요를 참조하십시오. ACLs

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 ID 기반 정책이 IAM 엔티티 (IAM사용자 또는 역할) 에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 객체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 사용 IAM설명서의 IAM 엔티티의 권한 경계를 참조하십시오.
- 서비스 제어 정책 (SCPs) - SCPs 조직 또는 OU (조직 구성 단위) 에 대한 최대 권한을 지정하는 JSON AWS Organizations정책입니다. AWS Organizations 기업이 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직의 모든 기능을 사용하도록 설정하면 일부 또는 모든 계정에 서비스 제어 정책 (SCPs) 을 적용할 수 있습니다. 각 항목을 포함하여 구성원 계정의 엔티

티에 대한 권한을 SCP AWS 계정 루트 사용자제한합니다. Organizations 및 SCPs 에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하십시오.

- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM사용 설명서의 세션 정책을 참조하십시오.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련된 경우 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM사용 설명서의 정책 평가 로직을 참조하십시오.

AWS 병렬 컴퓨팅 서비스의 작동 방식 IAM

액세스를 관리하는 IAM 데 사용하기 전에 사용할 수 있는 IAM 기능에 대해 알아보십시오 AWS PCS. AWS PCS

IAM AWS 병렬 컴퓨팅 서비스와 함께 사용할 수 있는 기능

IAM기능	AWS PCS지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACLs	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예

IAM기능	AWS PCS지원
보안 주체 권한	예
서비스 역할	아니요
서비스 링크 역할	예

대부분의 IAM 기능과 함께 작동하는 방식 AWS PCS 및 기타 AWS 서비스를 개괄적으로 보려면 IAM 사용 IAM 설명서에서 함께 작동하는AWS 서비스를 참조하십시오.

ID 기반 정책은 다음과 같습니다. AWS PCS

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 만드는 방법을 알아보려면 사용 설명서의 IAM정책 생성을 참조하십시오. IAM

IAMID 기반 정책을 사용하면 허용 또는 거부된 작업 및 리소스는 물론 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 사용 IAM설명서의 IAM JSON 정책 요소 참조를 참조하십시오.

아이덴티티 기반 정책 예시: AWS PCS

AWS PCSID 기반 정책의 예를 보려면 을 참조하십시오. 병렬 컴퓨팅 서비스의 ID 기반 정책 예제 AWS

내 리소스 기반 정책 AWS PCS

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정

의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

계정 간 액세스를 활성화하려면 다른 계정의 전체 계정 또는 IAM 엔티티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 AWS 계정경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM사용 설명서의 계정 간 리소스 액세스를 참조하십시오. IAM

에 대한 정책 조치 AWS PCS

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

정책 `Action` 요소는 JSON 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 작업이 없는 권한 전용 작업과 같은 몇 가지 예외가 있습니다. API 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

AWS PCS작업 목록을 보려면 서비스 권한 부여 참조의 AWS 병렬 컴퓨팅 서비스에 의해 정의된 작업을 참조하십시오.

정책 조치는 조치 앞에 다음 접두사를 AWS PCS 사용합니다.

```
pcs
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "pcs:action1",
      "pcs:action2"
         ]
```

AWS PCSID 기반 정책의 예를 보려면 을 참조하십시오. 병렬 컴퓨팅 서비스의 ID 기반 정책 예제 AWS

에 대한 정책 리소스 AWS PCS

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource`JSON정책 요소는 작업이 적용되는 하나 또는 여러 개의 객체를 지정합니다. 문장에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. Amazon 리소스 이름 (ARN) 을 사용하여 리소스를 지정하는 것이 가장 좋습니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

AWS PCS리소스 유형 및 해당 ARNs 유형의 목록을 보려면 서비스 권한 부여 참조의 AWS 병렬 컴퓨팅 서비스에 의해 정의된 리소스를 참조하십시오. 각 리소스의 어떤 작업을 지정할 수 있는지 알아보려면 AWS 병렬 컴퓨팅 서비스에 의해 정의된 작업을 참조하십시오. ARN

AWS PCSID 기반 정책의 예를 보려면 을 참조하십시오. 병렬 컴퓨팅 서비스의 ID 기반 정책 예제 AWS

에 대한 정책 조건 키 AWS PCS

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리

적 0R 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어 리소스에 IAM 사용자 이름이 태그가 지정된 경우에만 리소스에 대한 액세스 권한을 IAM 사용자에게 부여할 수 있습니다. 자세한 내용은 IAM사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하십시오.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM사용 설명서의AWS 글로벌 조건 컨텍스트 키를 참조하십시오.

AWS PCS조건 키 목록을 보려면 서비스 권한 부여 참조의 AWS 병렬 컴퓨팅 서비스의 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업 및 리소스에 대해 알아보려면 AWS 병렬 컴퓨팅 서비스에서 정의한 작업을 참조하십시오.

AWS PCSID 기반 정책의 예를 보려면 을 참조하십시오. 병렬 컴퓨팅 서비스의 ID 기반 정책 예제 AWS

ACLs에서 AWS PCS

지원ACLs: 아니요

액세스 제어 목록 (ACLs) 은 리소스에 액세스할 수 있는 권한을 가진 주체 (계정 구성원, 사용자 또는 역할) 를 제어합니다. ACLs정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 JSON 비슷합니다.

ABAC... AWS PCS

지원 ABAC (정책의 태그): 예

속성 기반 액세스 제어 (ABAC) 는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM엔티티 (사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. 의 ABAC 첫 번째 단계는 엔티티와 리소스에 태그를 지정하는 것입니다. 그런 다음 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC빠르게 성장하는 환경에서 유용하며 정책 관리가 복잡해지는 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

에 대한 자세한 내용은 What is ABAC? 를 참조하십시오. ABAC IAM사용 설명서에서. 설정 ABAC 단계가 포함된 자습서를 보려면 사용 IAM설명서의 속성 기반 액세스 제어 사용 (ABAC) 을 참조하십시오.

임시 자격 증명 사용 AWS PCS

임시 자격 증명 지원: 예

임시 자격 증명을 사용하여 로그인하면 작동하지 AWS 서비스 않는 것도 있습니다. 임시 자격 증명을 사용하는 AWS 서비스 방법을 비롯한 추가 정보는 IAM사용 설명서의 AWS 서비스 해당 자격 증명을 참조하십시오. IAM

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하는 경우 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 Single Sign-On (SSO) 링크를 AWS 사용하여 액세스하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM사용 설명서의 역할 전환 (콘솔) 을 참조하십시오.

AWS CLI 또는 를 사용하여 임시 자격 증명을 수동으로 생성할 수 AWS API 있습니다. 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 내용은 의 임시 보안 자격 증명을 참조하십시오. IAM

서비스 간 사용자 권한: AWS PCS

순방향 액세스 세션 지원 (FAS): 예

에서 IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 사용자는 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS를 호출하는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. AWS 서비스 FAS요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS요청 시 적용되는 정책 세부 정보는 전달 액세스 세션을 참조하십시오.

AWS PCS의 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 간주하는 IAM역할입니다. IAM관리자는 내부에서 IAM 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 사용 설명서의 역할 만들기를 참조하여 권한을 위임하십시오IAM. AWS 서비스

> **Warning**
>
> 서비스 역할의 권한을 변경하면 AWS PCS 기능이 중단될 수 있습니다. 서비스 역할을 편집하기 위한 지침이 AWS PCS 제공되는 경우에만 서비스 역할을 편집하십시오.

서비스 연결 역할은 다음과 같습니다. AWS PCS

서비스 링크 역할 지원: 예

서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.

서비스 연결 역할을 만들거나 관리하는 방법에 대한 자세한 내용은 함께 작동하는 AWS 서비스를 참조하십시오. IAM 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

병렬 컴퓨팅 서비스의 ID 기반 정책 예제 AWS

기본적으로 사용자와 역할에는 리소스를 만들거나 수정할 AWS PCS 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 를 사용하여 작업을 수행할 수도 없습니다 AWS API. IAM관리자는 IAM 정책을 생성하여 필요한 리소스에서 작업을 수행할 수 있는 권한을 사용자에게 부여할 수 있습니다. 그러면 관리자가 역할에 IAM 정책을 추가할 수 있으며, 사용자는 역할을 수임할 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 만드는 방법을 알아보려면 사용 IAM설명서에서 IAM 정책 생성을 참조하십시오.

각 리소스 유형의 형식을 포함하여 에서 정의한 AWS PCS 작업 및 리소스 유형에 ARNs 대한 자세한 내용은 서비스 권한 부여 참조의 AWS 병렬 컴퓨팅 서비스를 위한 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- AWS PCS콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

ID 기반 정책은 누군가가 사용자 계정에서 AWS PCS 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하여 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 내용은 IAM사용 설명서의 AWS 관리형 정책 또는 작업 기능에 대한AWS 관리형 정책을 참조하십시오.
- 최소 권한 적용 — IAM 정책으로 권한을 설정하는 경우 작업 수행에 필요한 권한만 부여하십시오. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. 를 사용하여 권한을 IAM 적용하는 방법에 대한 자세한 내용은 사용 설명서의 정책 및 권한을 참조하십시오. IAM IAM
- IAM정책의 조건을 사용하여 액세스를 추가로 제한 - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, 를 사용하여 모든 요청을 전송하도록 지정하는 정책 조건을 작성할 수 SSL 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 내용은 IAM사용 설명서의 IAMJSON정책 요소: 조건을 참조하십시오.
- IAMAccess Analyzer를 사용하여 IAM 정책을 검증하여 안전하고 기능적인 권한을 보장합니다. IAM Access Analyzer는 새 정책과 기존 정책을 검증하여 정책이 IAM 정책 언어 (JSON) 및 IAM 모범 사례를 준수하는지 확인합니다. IAMAccess Analyzer는 안전하고 기능적인 정책을 작성하는 데 도움이 되는 100개 이상의 정책 검사와 실행 가능한 권장 사항을 제공합니다. 자세한 내용은 사용 설명서의 IAMAccess Analyzer 정책 검증을 참조하십시오. IAM
- 다단계 인증 필요 (MFA) - 사용자 또는 루트 IAM 사용자가 필요한 시나리오가 있는 경우 보안을 강화하려면 이 기능을 MFA 켜십시오. AWS 계정 API작업 호출 MFA 시기를 요구하려면 정책에 MFA 조건을 추가하세요. 자세한 내용은 IAM사용 설명서의 MFA -보호된 API 액세스 구성을 참조하십시오.

의 모범 사례에 IAM 대한 자세한 내용은 IAM사용 설명서의 보안 모범 사례를 참조하십시오. IAM

AWS PCS콘솔 사용

AWS 병렬 컴퓨팅 서비스 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 내 AWS PCS 리소스의 세부 정보를 나열하고 볼 수 있어야 AWS 계정합니다. 최소 필수 권한보다 더

제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 에만 전화를 거는 사용자에게 최소 콘솔 권한을 허용할 필요는 AWS API 없습니다. 대신 수행하려는 작업과 일치하는 API 작업에만 액세스를 허용하세요.

AWS PCS콘솔 사용에 필요한 최소 권한에 대한 자세한 내용은 을 참조하십시오최소 권한: AWS PCS.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제에서는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 만드는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 OR를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 AWS CLI 권한이 포함됩니다. AWS API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

AWSAWS 병렬 컴퓨팅 서비스에 대한 관리형 정책

AWS 관리형 정책은 에서 생성하고 관리하는 독립형 정책입니다. AWS AWS 관리형 정책은 많은 일반 사용 사례에 대한 권한을 제공하도록 설계되었으므로 사용자, 그룹 및 역할에 권한을 할당하기 시작할 수 있습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한 권한을 부여하지 않을 수도 있다는 점에 유의하세요. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

관리형 정책에 정의된 권한은 변경할 수 없습니다. AWS AWS 관리형 정책에 정의된 권한을 업데이트하는 경우 AWS 해당 업데이트는 정책이 연결된 모든 주체 ID (사용자, 그룹, 역할) 에 영향을 미칩니다. AWS 새 정책이 시작되거나 기존 서비스에 새 AWS 서비스 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 가장 높습니다.

자세한 내용은 IAM사용 설명서의 AWS 관리형 정책을 참조하십시오.

AWS 관리형 정책: AWSPCSServiceRolePolicy

AWSPCSServiceRolePolicy IAM엔티티에 연결할 수 없습니다. 이 정책은 사용자를 AWS PCS 대신하여 작업을 수행할 수 있는 서비스 연결 역할에 연결됩니다. 자세한 내용은 AWS PCS에 대한 서비스 연결 역할 단원을 참조하십시오.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `ec2`— Amazon AWS PCS EC2 리소스를 생성하고 관리할 수 있습니다.
- `iam`— Amazon AWS PCS EC2 플릿에 대한 서비스 연결 역할을 생성하고 Amazon에 역할을 전달할 수 있습니다. EC2
- `cloudwatch`— AWS PCS Amazon에 서비스 지표를 게시할 수 CloudWatch 있습니다.
- `secretsmanager`— AWS PCS 클러스터 리소스의 비밀을 관리할 수 AWS PCS 있습니다.

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

```
    "Statement": [
        {
            "Sid": "PermissionsToCreatePCSNetworkInterfaces",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSPCSManaged": "false"
                }
            }
        },
        {
            "Sid": "PermissionsToCreatePCSNetworkInterfacesInSubnet",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Sid": "PermissionsToManagePCSNetworkInterfaces",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "Null": {
                        "aws:ResourceTag/AWSPCSManaged": "false"
                }
            }
        },
        {
            "Sid": "PermissionsToDescribePCSResources",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeVpcs",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PermissionsToCreatePCSLaunchTemplates",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplate"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSPCSManaged": "false"
                }
            }
        },
        {
            "Sid": "PermissionsToManagePCSLaunchTemplates",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteLaunchTemplateVersions",
                "ec2:CreateLaunchTemplateVersion"
            ],
            "Resource": "arn:aws:ec2:*:*:launch-template/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSPCSManaged": "false"
                }
            }
        },
        {
```

```
            "Sid": "PermissionsToTerminatePCSManagedInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:TerminateInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSPCSManaged": "false"
                }
            }
        },
        {
            "Sid": "PermissionsToPassRoleToEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/*/AWSPCS*",
                "arn:aws:iam::*:role/AWSPCS*",
                "arn:aws:iam::*:role/aws-pcs/*",
                "arn:aws:iam::*:role/*/aws-pcs/*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "PermissionsToControlClusterInstanceAttributes",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:CreateFleet"
            ],
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
```

```
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:launch-template/*",
                "arn:aws:ec2:*:*:placement-group/*",
                "arn:aws:ec2:*:*:capacity-reservation/*",
                "arn:aws:resource-groups:*:*:group/*",
                "arn:aws:ec2:*:*:fleet/*"
            ]
        },
        {
            "Sid": "PermissionsToProvisionClusterInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:CreateFleet"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSPCSManaged": "false"
                }
            }
        },
        {
            "Sid": "PermissionsToTagPCSResources",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "RunInstances",
                        "CreateLaunchTemplate",
                        "CreateFleet",
                        "CreateNetworkInterface"
                    ]
                }
            }
        },
```

```
        {
            "Sid": "PermissionsToPublishMetrics",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/PCS"
                }
            }
        },
        {
            "Sid": "PermissionsToManageSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue",
                "secretsmanager:UpdateSecretVersionStage",
                "secretsmanager:DeleteSecret"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:pcs!*",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
 "pcs",
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
           }
        }
    ]
}
```

AWS PCS AWS 관리형 정책 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 AWS PCS 이후의 AWS 관리형 정책 업데이트에 대한 세부 정보를 볼 수 있습니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS PCS 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AWS PCS에서 변경 사항 추적 시작	AWS PCS AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2024년 8월 28일

AWS PCS에 대한 서비스 연결 역할

AWS 병렬 컴퓨팅 서비스는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 직접 연결된 고유한 IAM 역할 유형입니다. AWS PCS 서비스 연결 역할은 미리 정의되며 서비스가 사용자를 AWS PCS 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

서비스에 연결된 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로 설정이 AWS PCS 더 쉬워집니다. AWS PCS서비스 연결 역할의 권한을 정의하며, 달리 정의하지 않는 한 역할만 맡을 AWS PCS 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔티티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제해야만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 AWS PCS 리소스 액세스 권한을 실수로 제거할 수 없으므로 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 함께 작동하는AWS 서비스를 IAM 참조하고 서비스 연결 역할 열에서 '예'로 표시된 서비스를 찾아보세요. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

서비스 연결 역할 권한에 대한 AWS PCS

AWS PCSAWSServiceRoleForPCS— Allow라는 서비스 연결 역할을 AWS PCS 사용하여 Amazon EC2 리소스를 관리합니다.

AWSServiceRoleForPCS 서비스 연결 역할은 역할을 맡을 다음 서비스를 신뢰합니다.

- `pcs.amazonaws.com`

지정된 역할 권한 정책을 AWSPCSServiceRolePolicy사용하면 특정 리소스에 대한 작업을 AWS PCS 완료할 수 있습니다.

사용자, 그룹 또는 역할이 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 사용 권한을 구성해야 합니다. 자세한 내용은 사용 IAM설명서의 서비스 연결 역할 권한을 참조하십시오.

에 대한 서비스 연결 역할 만들기 AWS PCS

서비스 연결 역할을 수동으로 만들 필요는 없습니다. AWS PCS클러스터를 생성할 때 서비스 연결 역할을 자동으로 생성합니다.

에 대한 서비스 연결 역할 편집 AWS PCS

AWS PCS AWSServiceRoleForPCS 서비스 연결 역할을 편집할 수 없습니다. 서비스 링크 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 를 사용하여 역할에 대한 설명을 편집할 수 있습니다. IAM 자세한 내용은 사용 IAM설명서의 서비스 연결 역할 편집을 참조하십시오.

에 대한 서비스 연결 역할 삭제 AWS PCS

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다. 단, 서비스 링크 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

> **Note**
>
> AWS PCS서비스가 역할을 사용하고 있을 때 리소스를 삭제하려고 하면 삭제가 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

에서 사용하는 AWS PCS 리소스를 제거하려면 AWSServiceRoleForPCS

AWSServiceRoleForPCS 서비스 연결 역할을 삭제하려면 클러스터를 모두 삭제해야 합니다. 자세한 내용은 클러스터 삭제를 참조하십시오.

를 사용하여 서비스 연결 역할을 수동으로 삭제하려면 IAM

IAM콘솔 AWS CLI, 또는 를 AWS API 사용하여 AWSServiceRoleForPCS 서비스 연결 역할을 삭제합니다. 자세한 내용은 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오. IAM

AWS PCS 서비스 링크 역할이 지원되는 리전

AWS PCS서비스를 사용할 수 있는 모든 지역에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 섹션을 참조하십시오.

아마존 EC2 스팟 역할 AWS PCS

스팟을 구매 옵션으로 사용하는 AWS PCS 컴퓨팅 노드 그룹을 생성하려면 AWSServiceRoleForEC2Spot서비스 연결 역할도 있어야 합니다. AWS 계정다음 AWS CLI 명령을 사용하여 역할을 생성할 수 있습니다. 자세한 내용은 사용 설명서의 서비스 연결 역할 만들기 및 서비스에 권한을 위임할 역할 만들기를 AWS 참조하십시오.AWS Identity and Access Management

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

> **Note**
>
> AWS 계정 이미 역할이 있는 경우 다음과 같은 오류 메시지가 나타납니다. `AWSServiceRoleForEC2Spot` IAM
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
> operation: Service role name AWSServiceRoleForEC2Spot has been taken in this
> account, please try a different suffix.
> ```

최소 권한: AWS PCS

이 섹션에서는 IAM ID (사용자, 그룹 또는 역할) 가 서비스를 사용하는 데 필요한 최소 IAM 권한을 설명합니다.

목차

API작업을 사용하기 위한 최소 권한

API액션	최소 권한	콘솔에 대한 추가 권한
CreateCluster	`ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:GetSecurityGroupsForVpc, iam:CreateServiceLinkedRole, secretsmanager:CreateSecret, secretsmanager:TagResource, pcs:CreateCluster`	
ListClusters	`pcs:ListClusters`	
GetCluster	`pcs:GetCluster`	`ec2:DescribeSubnets`
DeleteCluster	`pcs:DeleteCluster`	
CreateComputeNodeGroup	`ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags,`	`iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster`

API액션	최소 권한	콘솔에 대한 추가 권한
	`iam:PassRole`, `iam:GetInstanceProfile`, `pcs:CreateComputeNodeGroup`	
ListComputerNodeGroups	`pcs:ListComputeNodeGroups`	`pcs:GetCluster`
GetComputeNodeGroup	`pcs:GetComputeNodeGroup`	`ec2:DescribeSubnets`
UpdateComputeNodeGroup	`ec2:DescribeVpcs`, `ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups`, `ec2:DescribeLaunchTemplates`, `ec2:DescribeLaunchTemplateVersions`, `ec2:DescribeInstanceTypes`, `ec2:RunInstances`, `ec2:CreateFleet`, `ec2:CreateTags`, `iam:PassRole`, `iam:GetInstanceProfile`, `pcs:UpdateComputeNodeGroup`	`pcs:GetComputeNodeGroup`, `iam:ListInstanceProfiles`, `ec2:DescribeImages`, `pcs:GetCluster`
DeleteComputeNodeGroup	`pcs:DeleteComputeNodeGroup`	

API액션	최소 권한	콘솔에 대한 추가 권한
CreateQueue	`pcs:CreateQueue`	`pcs:ListComputeNodeGroups, pcs:GetCluster`
ListQueues	`pcs:ListQueues`	`pcs:GetCluster`
GetQueue	`pcs:GetQueue`	
UpdateQueue	`pcs:UpdateQueue`	`pcs:ListComputeNodeGroups, pcs:GetQueue`
DeleteQueue	`pcs:DeleteQueue`	

태그 사용에 필요한 최소 권한

리소스가 포함된 태그를 사용하려면 다음 권한이 필요합니다 AWS PCS.

```
pcs:ListTagsForResource
pcs:TagResource
pcs:UntagResource
```

로그를 지원하는 데 필요한 최소 권한

AWS PCSAmazon CloudWatch Logs (CloudWatch 로그) 로 로그 데이터를 전송합니다. ID에 CloudWatch 로그를 사용할 수 있는 최소 권한이 있는지 확인해야 합니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs 리소스에 대한 액세스 권한 관리 개요를 참조하십시오.

서비스가 로그를 Logs로 전송하는 데 필요한 권한에 대한 자세한 내용은 Amazon CloudWatch CloudWatch Logs 사용 설명서의 AWS 서비스에서 로깅 활성화를 참조하십시오.

서비스 관리자의 최소 권한

다음 IAM 정책은 IAM ID (사용자, 그룹 또는 역할) 가 AWS PCS 서비스를 구성하고 관리하는 데 필요한 최소 권한을 지정합니다.

> **Note**
>
> 서비스를 구성 및 관리하지 않는 사용자에게는 이러한 권한이 필요하지 않습니다. 작업만 실행하는 사용자는 보안 셸 (SSH) 을 사용하여 클러스터에 연결합니다. AWS Identity and Access Management (IAM) 는 에 대한 인증 또는 권한 부여를 처리하지 않습니다SSH.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:GetSecurityGroupsForVpc",
                "firehose:*",
                "iam:GetInstanceProfile",
                "iam:ListInstanceProfiles",
                "iam:PassRole",
                "kms:*",
                "logs:*",
                "pcs:*",
                "s3:*"
            ],
            "Resource": "*"
        }
    ]
}
```

정책에서 다음 권한을 제외하고 대신 해당 관리형 정책을 다음에서 사용할 수 있습니다IAM.

- `"firehose:*"`

  ```
  AmazonKinesisFirehoseFullAccess
  ```

- `"kms:*"`

  ```
  AWSKeyManagementServicePowerUser
  ```

- `"logs:*"`

  ```
  CloudWatchLogsFullAccess
  ```

- `"s3:*"`

  ```
  AmazonS3FullAccess
  ```

IAM AWS 병렬 컴퓨팅 서비스의 인스턴스 프로필

EC2인스턴스에서 실행되는 애플리케이션은 모든 AWS API 요청에 AWS 자격 증명을 포함해야 합니다. IAM역할을 사용하여 EC2 인스턴스의 임시 자격 증명을 관리하는 것이 좋습니다. 이를 위한 인스턴스 프로필을 정의하고 이를 인스턴스에 연결할 수 있습니다. 자세한 내용은 Amazon EC2 Elastic Compute 클라우드 사용 설명서에서 Amazon의 IAM 역할을 참조하십시오.

> **Note**
>
> 를 사용하여 EC2 Amazon의 AWS Management Console IAM 역할을 생성하면 콘솔이 자동으로 인스턴스 프로필을 생성하고 IAM 역할과 동일한 이름을 지정합니다. AWS CLI, AWS API 작업 또는 AWS SDK an을 사용하여 IAM 역할을 생성하는 경우 별도의 작업으로 인스턴스 프로필이 생성됩니다. 자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 인스턴스 프로필을 참조하십시오.

컴퓨팅 노드 그룹을 생성할 때 인스턴스 프로필을 지정해야 합니다. ARN 일부 또는 모든 컴퓨팅 노드 그룹에 대해 서로 다른 인스턴스 프로필을 선택할 수 있습니다.

인스턴스 프로필 요구 사항

인스턴스 프로필 이름

IAM인스턴스 프로필은 해당 경로에 로 AWSPCS 시작하거나 해당 경로에 ARN /aws-pcs/ 포함되어야 합니다.

Example

- arn:aws:iam::*:instance-profile/AWSPCS-example-role-1 및
- arn:aws:iam::*:instance-profile/aws-pcs/example-role-2.

권한

의 인스턴스 프로필에는 최소한 다음 정책이 AWS PCS 포함되어야 합니다. 이를 통해 컴퓨팅 노드가 작동하게 되면 이를 AWS PCS 서비스에 알릴 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "pcs:RegisterComputeNodeGroupInstance"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

추가 정책

인스턴스 프로필에 관리형 정책을 추가하는 것을 고려해 볼 수 있습니다. 예:

- ReadOnlyAccessAmazonS3는 모든 S3 버킷에 대한 읽기 전용 액세스를 제공합니다.
- A는 Amazon Management 콘솔에서 직접 원격 액세스하는 것과 같은 AWS Systems Manager 서비스의 핵심 기능을 mazonSSMManaged InstanceCore 지원합니다.
- CloudWatchAgentServerPolicy AmazonCloudWatchAgent 서버에서 사용하는 데 필요한 권한이 포함되어 있습니다.

특정 사용 사례를 지원하는 자체 IAM 정책을 포함할 수도 있습니다.

인스턴스 프로파일 생성

Amazon EC2 콘솔에서 직접 인스턴스 프로필을 생성할 수 있습니다. 자세한 내용은 사용 AWS Identity and Access Management 설명서의 인스턴스 프로필 사용을 참조하십시오.

AWS 병렬 컴퓨팅 서비스 ID 및 액세스 문제 해결

다음 정보를 사용하면 및 을 (를) 사용할 때 발생할 수 있는 일반적인 문제를 AWS PCS 진단하고 해결하는 데 도움이 IAM 됩니다.

주제

- 다음과 같은 작업을 수행할 권한이 없습니다. AWS PCS
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 외부 사용자가 내 AWS PCS 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

다음과 같은 작업을 수행할 권한이 없습니다. AWS PCS

작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 권한이 없는 경우 발생합니다. `pcs:`*`GetWidget`*

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 pcs:GetWidget on resource: my-example-widget
```

이 경우 `pcs:`*`GetWidget`*작업을 사용하여 *`my-example-widget`*리소스에 액세스할 수 있도록 `mateojackson`사용자 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

작업을 수행할 권한이 없다는 오류가 발생하는 경우 역할을 넘길 수 있도록 정책을 업데이트해야 합니다. `iam:PassRole` AWS PCS

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 라는 IAM 사용자가 콘솔을 사용하여 에서 작업을 marymajor 수행하려고 할 때 발생합니다. AWS PCS 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 AWS PCS 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록 (ACLs) 을 지원하는 서비스의 경우 이러한 정책을 사용하여 사용자에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 이러한 기능의 AWS PCS 지원 여부를 알아보려면 을 참조하십시오. AWS 병렬 컴퓨팅 서비스의 작동 방식 IAM
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 사용 설명서에서 AWS 계정 자신이 소유한 다른 IAM 사용자의 액세스 권한 제공을 IAM 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM사용 설명서의 제3자가 AWS 계정 소유한 리소스에 대한 액세스 제공을 참조하십시오. AWS 계정
- ID 페더레이션을 통해 액세스를 제공하는 방법을 알아보려면 사용 설명서의 외부 인증된 사용자에게 액세스 제공 (ID 페더레이션) 을 IAM 참조하십시오.
- 계정 간 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 사용 설명서의 계정 간 리소스 액세스를 참조하십시오. IAM IAM

병렬 컴퓨팅 서비스에 대한 AWS 규정 준수 검증

특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면AWS 서비스 규정 준수 프로그램의AWS 서비스 범위별, 규정 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 AWS 규정 준수 프로그램AWS 보증 프로그램 규정AWS 참조하십시오.

를 사용하여 AWS Artifact타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 퀵 스타트 가이드 - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 — 이 백서에서는 기업이 적합한 애플리케이션을 만드는 AWS HIPAA 데 사용할 수 있는 방법을 설명합니다.

> **Note**
>
> 모든 AWS 서비스 사람이 자격이 있는 것은 아닙니다. HIPAA 자세한 내용은 HIPAA적격 서비스 참조를 참조하십시오.

- AWS 규정AWS 준수 리소스 — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (국립 표준 기술 연구소 (NIST), 결제 카드 산업 보안 표준 위원회 (), 국제 표준화 기구 ()) 를 포함한PCI) 전반의 보안 제어에 대한 지침을 매핑합니다. ISO
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub— 이를 AWS 서비스 통해 내부 AWS보안 상태를 포괄적으로 파악할 수 있습니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty — 환경에 의심스럽고 악의적인 활동이 있는지 AWS 계정모니터링하여 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하는 PCI DSS 등 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 수 있습니다.
- AWS Audit Manager— 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

AWS 병렬 컴퓨팅 서비스의 탄력성

AWS 글로벌 인프라는 가용 영역을 중심으로 구축됩니다 AWS 리전 . AWS 리전 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다. 이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

가용 영역에 대한 AWS 리전 자세한 내용은 글로벌 인프라를 참조하십시오AWS .

AWS 병렬 컴퓨팅 서비스의 인프라 보안

AWS 병렬 컴퓨팅 서비스는 관리형 서비스로서 AWS 글로벌 네트워크 보안의 보호를 받습니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 AWS PCS 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안 (TLS). TLS1.2가 필요하고 TLS 1.3을 권장합니다.
- (임시 디피-헬만) 또는 (타원 곡선 임시 디피-헬만PFS) 와 같이 완벽한 순방향 기밀성 DHE () 을 갖춘 암호 제품군. ECDHE Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 액세스 키 ID와 보안 주체와 연결된 비밀 액세스 키를 사용하여 요청에 서명해야 합니다. IAM 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

클러스터를 AWS PCS 생성하면 서비스는 계정의 컴퓨팅 노드와는 별도로 서비스 소유 계정에서 Slurm 컨트롤러를 시작합니다. 컨트롤러와 컴퓨팅 노드 간의 통신을 연결하려면 사용자 내에 계정 간 Elastic Network Interface () 를 AWS PCS 생성하십시오. ENI VPC Slurm 컨트롤러는 를 사용하여 여러 AWS 계정컴퓨팅 노드를 관리하고 통신하여 리소스의 보안과 격리를 유지하면서 효율적인 HPC AI/ML 작업을 촉진합니다. ENI

병렬 컴퓨팅 서비스의 취약성 분석 및 관리 AWS

구성 및 IT 제어는 사용자와 사용자 간의 AWS 공동 책임입니다. 자세한 내용은 AWS 공동 책임 모델을 참조하십시오. AWS 컨트롤러 인스턴스의 운영 체제 패치, 방화벽 구성, 인프라 재해 복구와 같은 서비

스 계정의 기본 AWS 인프라에 대한 기본 보안 작업을 처리합니다. 적합한 제3자가 이 절차를 검토하고 인증하였습니다. 자세한 내용은 보안, 자격 증명 및 규정 준수를 위한 모범 사례를 참조하세요.

다음과 같은 기본 인프라의 보안에 대한 책임은 귀하에게 있습니다. AWS 계정

- 업데이트 및 보안 패치를 포함한 코드를 유지 관리하세요.
- 노드 그룹 인스턴스의 운영 체제를 패치하고 업데이트하십시오.
- 스케줄러를 업데이트하여 지원되는 버전 내에서 유지하세요.
- 사용자 클라이언트와 해당 클라이언트가 연결하는 노드 간의 통신을 인증하고 암호화합니다.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 크로스 서비스 사칭은 AWS대리인 문제로 혼란스러운 결과를 초래할 수 있습니다. 교차 서비스 가장은 한 서비스(호출하는 서비스)가 다른 서비스(호출되는 서비스)를 직접적으로 호출할 때 발생할 수 있습니다. 직접적으로 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS 에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

AWS Parallel Computing Service (AWS PCS) 가 리소스에 다른 서비스에 부여하는 권한을 제한하려면 리소스 정책에 `aws:SourceArn`및 `aws:SourceAccount`글로벌 조건 컨텍스트 키를 사용하는 것이 좋습니다. 하나의 리소스만 교차 서비스 액세스와 연결되도록 허용하려는 경우 `aws:SourceArn`를 사용하십시오. 해당 계정의 모든 리소스가 교차 서비스 사용과 연결되도록 허용하려는 경우 `aws:SourceAccount`을(를) 사용합니다.

혼란스러운 대리자 문제를 방지하는 가장 효과적인 방법은 전체 ARN 리소스와 함께 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용하는 것입니다. 전체 ARN 리소스를 모르거나 여러 리소스를 지정하는 경우에는 `aws:SourceArn` 글로벌 컨텍스트 조건 키를 와일드카드 문자 (*) 와 함께 사용하여 의 알 수 없는 부분을 지정하십시오. ARN 예: `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

`aws:SourceArn`값에 계정 ID (예: Amazon S3 버킷ARN) 가 포함되어 있지 않은 경우 두 글로벌 조건 컨텍스트 키를 모두 사용하여 권한을 제한해야 합니다.

의 값은 `aws:SourceArn` 클러스터여야 합니다ARN.

다음 예는 에서 `aws:SourceArn` 및 `aws:SourceAccount` 글로벌 조건 컨텍스트 키를 사용하여 혼동되는 부정 문제를 방지하는 AWS PCS 방법을 보여줍니다.

```
{
"Version": "2012-10-17",
  "Statement": {
"Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "pcs.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": [
          "arn:aws:pcs:us-east-1:123456789012:cluster/*"
        ]
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

IAM컴퓨팅 노드 그룹의 일부로 프로비저닝된 Amazon EC2 인스턴스의 역할

AWS PCS클러스터에 구성된 각 컴퓨팅 노드 그룹의 Amazon EC2 용량을 자동으로 조정합니다. 컴퓨팅 노드 그룹을 생성할 때 사용자는 필드를 통해 IAM 인스턴스 프로필을 제공해야 합니다. `iamInstanceProfileArn` 인스턴스 프로필은 EC2 프로비저닝된 인스턴스와 관련된 권한을 지정합니다. AWS PCS역할 이름 접두사가 있거나 `/aws-pcs/` 역할 경로의 일부인 모든 AWSPCS 역할을 수락합니다. 컴퓨팅 노드 그룹을 생성하거나 업데이트하는 IAM ID (사용자 또는 역할) 에 대한 `iam:PassRole` 권한이 필요합니다. 사용자가 `CreateComputeNodeGroup` 또는 `UpdateComputeNodeGroup` API 작업을 호출하면 사용자가 작업을 수행할 수 있는지 AWS PCS 확인합니다. `iam:PassRole`

다음 예제 정책은 이름이 로 시작하는 IAM 역할만 전달할 수 있는 권한을 AWSPCS 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::123456789012:role/AWSPCS*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

AWS 병렬 컴퓨팅 서비스의 보안 모범 사례

이 섹션에서는 AWS 병렬 컴퓨팅 서비스 (AWS PCS) 와 관련된 보안 모범 사례를 설명합니다. 의 보안 모범 사례에 대해 자세히 알아보려면 보안 AWS, ID 및 규정 준수의 모범 사례를 참조하십시오.

AMI관련 보안

- 프로덕션 AMIs 워크로드에는 AWS PCS 샘플을 사용하지 마세요. AMIs샘플은 지원되지 않으며 테스트용으로만 사용됩니다.
- AWS PCS인스턴스의 운영 체제와 소프트웨어를 정기적으로 업데이트하여 취약성을 완화하십시오.
- 패치를 자동화하고 보안 정책 준수를 유지하는 AWS Systems Manager 데 사용합니다.
- 공식 소스에서 다운로드한 인증된 공식 AWS PCS 패키지만 사용하십시오. AWS
- 컴퓨팅 노드의 AWS PCS 패키지를 정기적으로 업데이트하여 보안 패치 및 개선 사항을 받으십시오. 취약성을 최소화하려면 이 프로세스를 자동화하는 것을 고려해 보십시오.

Slurm 워크로드 매니저 보안

- 액세스 제어 및 네트워크 제한을 구현하여 Slurm 제어 및 컴퓨팅 노드를 보호하십시오. 신뢰할 수 있는 사용자와 시스템만 작업을 제출하고 Slurm 관리 명령에 액세스할 수 있도록 허용하십시오.
- Slurm 인증과 같은 Slurm에 내장된 보안 기능을 사용하여 작업 제출 및 통신이 인증되었는지 확인하십시오.
- Slurm 버전을 업데이트하여 원활한 운영과 클러스터 지원을 유지하십시오.

> **Important**
>
> 지원 수명이 끝난 Slurm 버전 (EOSL) 을 사용하는 모든 클러스터는 즉시 중지됩니다. 사용자 가이드 페이지 상단의 링크를 사용하여 AWS PCS 설명서 RSS 피드를 구독하면 Slurm 버전이 출시될 때 알림을 받을 수 있습니다. EOSL

모니터링 및 로깅

- Amazon CloudWatch AWS CloudTrail Logs를 사용하여 클러스터의 작업을 모니터링하고 기록하십시오 AWS 계정. 데이터를 문제 해결 및 감사에 사용하십시오.

네트워크 보안

- AWS PCS클러스터를 별도로 VPC 배포하여 HPC 환경을 다른 네트워크 트래픽으로부터 격리하십시오.
- 보안 그룹과 네트워크 액세스 제어 목록 (ACLs) 을 사용하여 AWS PCS 인스턴스와 서브넷에 대한 인바운드 및 아웃바운드 트래픽을 제어할 수 있습니다.
- AWS PrivateLink 또는 VPC 엔드포인트를 사용하여 클러스터와 네트워크 내부의 다른 AWS 서비스 간의 네트워크 트래픽을 유지합니다. AWS

에 대한 로깅 및 모니터링 AWS PCS

모니터링은 기타 AWS 리소스의 안정성, 가용성, 성능을 유지하는 데 AWS PCS 있어 중요한 부분입니다. AWS문제 발생 시 이를 확인하고 보고하고 적절한 AWS PCS 경우 자동 조치를 취할 수 있는 다음과 같은 모니터링 도구를 제공합니다.

- Amazon은 실행 중인 AWS 리소스와 애플리케이션을 AWS 실시간으로 CloudWatch 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요할 때 새 인스턴스를 자동으로 시작할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.
- Amazon CloudWatch Logs를 사용하면 Amazon EC2 인스턴스 및 기타 소스에서 로그 파일을 모니터링 CloudTrail, 저장 및 액세스할 수 있습니다. CloudWatch 로그는 로그 파일의 정보를 모니터링하고 특정 임계값이 충족되면 알려줄 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서를 참조하십시오.
- AWS CloudTrail사용자 계정에서 또는 AWS 계정을 대신하여 이루어진 API 호출 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 어떤 사용자와 계정이 전화를 걸었는지 AWS, 어떤 소스 IP 주소에서 호출이 이루어졌는지, 언제 호출이 발생했는지 식별할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

AWS PCS스케줄러 로그

클러스터 스케줄러에서 Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3) 및 Amazon Data Firehose로 상세한 로깅 데이터를 AWS PCS 전송하도록 구성할 수 있습니다. 이는 모니터링 및 문제 해결에 도움이 될 수 있습니다. AWS PCS콘솔을 사용하거나 또는 를 사용하여 프로그래밍 방식으로 AWS PCS 스케줄러 로그를 설정할 수 있습니다. AWS CLI SDK

목차

  - 클러스터 전송 소스를 전송 대상에 연결
- 스케줄러 로그 스트림 경로 및 이름
- 예제 AWS PCS 스케줄러 로그 레코드

사전 조건

AWSPCS클러스터를 관리하는 데 사용되는 IAM 보안 주체는 허용해야 합니다. `pcs:AllowVendedLogDeliveryForResource` 다음은 이를 활성화하는 샘플 AWS IAM 정책입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PcsAllowVendedLogsDelivery",
            "Effect": "Allow",
            "Action": ["pcs:AllowVendedLogDeliveryForResource"],
            "Resource": [
                "arn:aws:pcs:::cluster/*"
            ]
        }
    ]
}
```

콘솔을 AWS PCS 사용하여 스케줄러 로그 설정

콘솔에서 AWS PCS 스케줄러 로그를 설정하려면 다음 단계를 따르십시오.

1. AWS PCS콘솔을 엽니다.
2. 클러스터를 선택하고 로깅을 활성화할 AWS PCS 클러스터의 세부 정보 페이지로 이동합니다.
3. [Logs]를 선택합니다.
4. 언더 로그 전송 — 스케줄러 로그 — 선택 사항
   a. 로그 전송 대상을 최대 3개까지 추가할 수 있습니다. CloudWatch 로그, Amazon S3 또는 Firehose를 선택할 수 있습니다.
   b. [로그 전송 업데이트] 를 선택합니다.

이 페이지를 다시 방문하여 로그 전달을 재구성, 추가 또는 제거할 수 있습니다.

를 사용하여 스케줄러 로그를 설정합니다. AWS CLI

이 작업을 수행하려면 적어도 하나의 전송 대상, 하나의 전송 원본 (PCS클러스터) 및 하나의 배달 (원본과 대상을 연결하는 관계) 이 필요합니다.

배송 목적지를 생성하세요.

AWSPCS클러스터에서 스케줄러 로그를 수신하려면 전송 대상이 하나 이상 필요합니다. CloudWatch API사용 설명서의 PutDeliveryDestination 섹션에서 이 주제에 대해 자세히 알아볼 수 있습니다.

를 사용하여 배송지를 생성하려면 AWS CLI

- 다음 명령을 사용하여 목적지를 생성합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.
  - Replace *region-code* 목적지를 생성할 AWS 리전 위치를 사용합니다. 이 지역은 일반적으로 AWS PCS 클러스터가 배포되는 지역과 동일합니다.
  - Replace *pcs-logs-destination* 원하는 이름을 사용하세요. 계정 내 모든 배송 목적지에 대해 고유해야 합니다.
  - Replace *resource-arn* ARNLogs의 기존 로그 그룹, S3 버킷 또는 Firehose의 전송 스트림을 사용합니다. CloudWatch 그러한 예는 다음과 같습니다.
    - CloudWatch 로그 그룹

      ```
      arn:aws:logs:region-code:account-id:log-group:/log-group-name:*
      ```

    - S3 버킷

      ```
      arn:aws:s3:::bucket-name
      ```

    - Firehose 딜리버리 스트림

      ```
      arn:aws:firehose:region-code:account-id:deliverystream/stream-name
      ```

```
aws logs put-delivery-destination --region region-code \
    --name pcs-logs-destination \
    --delivery-destination-configuration destinationResourceArn=resource-arn
```

전송을 구성하는 데 필요하므로 새 전송 목적지의 경우 이 정보를 기록해 두십시오. ARN

AWS PCS클러스터를 전송 소스로 활성화합니다.

에서 AWS PCS 스케줄러 로그를 수집하려면 클러스터를 전송 소스로 구성하십시오. 자세한 내용은 Amazon CloudWatch 로그 API 참조를 참조하십시오 PutDeliverySource.

를 사용하여 클러스터를 전송 소스로 구성하려면 AWS CLI

- 다음 명령을 사용하여 클러스터에서 로그 전송을 활성화합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.
  - Replace *region-code* 클러스터가 배포된 AWS 리전 위치와 함께
  - Replace *cluster-logs-source-name* 이 소스의 이름을 포함합니다. 내 모든 배송 소스에서 고유해야 합니다 AWS 계정. AWS PCS클러스터의 이름 또는 ID를 통합하는 것을 고려해 보십시오.
  - Replace *cluster-arn* 사용 중인 클러스터와 함께 ARN 사용하십시오. AWS PCS

```
aws logs put-delivery-source \
    --region region-code \
    --name cluster-logs-source-name \
    --resource-arn cluster-arn \
    --log-type PCS_SCHEDULER_LOGS
```

클러스터 전송 소스를 전송 대상에 연결

클러스터에서 목적지로 스케줄러 로그 데이터가 흐르도록 하려면 스케줄러 로그 데이터를 연결하는 전송을 구성해야 합니다. 자세한 내용은 Amazon CloudWatch 로그 API 참조를 참조하십시오 CreateDelivery.

를 사용하여 배송을 생성하려면 AWS CLI

- 다음 명령을 사용하여 배달을 생성합니다. 명령을 실행하기 전에 다음과 같은 바꾸기를 합니다.
  - Replace *region-code* 소스 및 목적지가 AWS 리전 있는 위치를 사용합니다.
  - Replace *cluster-logs-source-name* 위에 적힌 배송 출처 이름과 함께
  - Replace *destination-arn* 로그를 전송하고자 하는 배송지에서 ARN 가져온 것과 함께 사용하세요.

```
aws logs create-delivery \
    --region region-code \
```

```
    --delivery-source-name cluster-logs-source \
    --delivery-destination-arn destination-arn
```

스케줄러 로그 스트림 경로 및 이름

AWSPCS스케줄러 로그의 경로와 이름은 대상 유형에 따라 달라집니다.

- CloudWatch 로그
  - A CloudWatch 로그 스트림은 이 명명 규칙을 따릅니다.

    ```
    AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
    ```

    Example

    ```
    AWSLogs/PCS/abcdef0123/slurmctld_24.05.log
    ```
- S3 버킷
  - S3 버킷 출력 경로는 다음과 같은 명명 규칙을 따릅니다.

    ```
    AWSLogs/${account-id}/PCS/${region}/${cluster_id}/${log_name}/
    ${scheduler_major_version}/yyyy/MM/dd/HH/
    ```

    Example

    ```
    AWSLogs/111111111111/PCS/us-east-2/abcdef0123/slurmctld/24.05/2024/09/01/00.
    ```
  - S3 객체 이름은 다음 규칙을 따릅니다.

    ```
    PCS_${log_name}_${scheduler_major_version}_#{expr date 'event_timestamp', format:
     "yyyy-MM-dd-HH"}_${cluster_id}_${hash}.log
    ```

    Example

    ```
    PCS_slurmctld_24.05_2024-09-01-00_abcdef0123_0123abcdef.log
    ```

예제 AWS PCS 스케줄러 로그 레코드

AWSPCS스케줄러 로그는 구조화되어 있습니다. 여기에는 Slurm 컨트롤러 프로세스에서 내보내는 로그 메시지 외에도 클러스터 식별자, 스케줄러 유형, 메이저 및 패치 버전과 같은 필드가 포함됩니다. 다음 예를 참고하세요


```
{
    "resource_id": "s3431v9rx2",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1721230979,
    "log_level": "info",
    "log_name": "slurmctld",
    "scheduler_type": "slurm",
    "scheduler_major_version": "23.11",
    "scheduler_patch_version": "8",
    "node_type": "controller_primary",
    "message": "[2024-07-17T15:42:58.614+00:00] Running as primary controller\n"
}
```


Amazon을 통한 AWS 병렬 컴퓨팅 서비스 모니터링 CloudWatch

CloudWatch Amazon은 클러스터에서 일정 간격으로 지표를 수집하여 AWS 병렬 컴퓨팅 서비스(AWS PCS) 클러스터 상태 및 성능을 모니터링합니다. 이러한 지표는 보존되므로 과거 데이터에 액세스하고 시간 경과에 따른 클러스터 성능에 대한 통찰력을 얻을 수 있습니다.

CloudWatch 또한 에서 시작한 EC2 인스턴스를 모니터링하여 규모 조정 요구 사항을 AWS PCS 충족할 수 있습니다. 실행 중인 인스턴스의 로그를 검사할 수 있지만 CloudWatch 지표와 로깅 데이터는 일반적으로 인스턴스가 종료되면 삭제됩니다. 하지만 EC2 시작 템플릿을 사용하여 인스턴스가 종료된 후에도 지표와 로그를 유지하도록 인스턴스에서 CloudWatch 에이전트를 구성하여 장기간 모니터링 및 분석이 가능하도록 할 수 있습니다.

모니터링을 AWS PCS 사용하는 CloudWatch 방법에 대해 자세히 알아보려면 이 섹션의 주제를 살펴보세요.

주제

- AWS PCS지표 모니터링: 사용 CloudWatch
- Amazon을 사용하여 AWS PCS 인스턴스 모니터링 CloudWatch

AWS PCS지표 모니터링: 사용 CloudWatch

AWS PCS클러스터에서 데이터를 수집하여 실시간에 가까운 지표로 변환하는 CloudWatch Amazon을 사용하여 클러스터 상태를 모니터링할 수 있습니다. 이러한 통계는 15개월 동안 보관되므로 기록 정보에 액세스하고 클러스터 성능을 더 잘 파악할 수 있습니다. 클러스터 지표는 1분 CloudWatch 간격으로 전송됩니다. 에 대한 CloudWatch 자세한 내용은 Amazon이란 무엇입니까 CloudWatch? 를 참조하십시오. Amazon CloudWatch 사용 설명서에서 확인할 수 있습니다.

AWS PCS의 AWS/PCS네임스페이스에 다음 지표를 게시합니다. CloudWatch 측정기준은 1개입니다. `ClusterId`

명칭	설명	단위
ActualCapacity	IdleCapacity + UtilizedCapacity	개수
CapacityUtilization	UtilizedCapacity / ActualCapacity	개수
DesiredCapacity	ActualCapacity + PendingCapacity	개수
IdleCapacity	실행 중이지만 작업에 할당되지 않은 인스턴스 수	개수
UtilizedCapacity	실행 중이고 작업에 할당된 인스턴스 수	개수

Amazon을 사용하여 AWS PCS 인스턴스 모니터링 CloudWatch

AWSPCSPCS컴퓨팅 노드 그룹에 정의된 조정 요구 사항을 충족하기 위해 필요에 따라 Amazon EC2 인스턴스를 시작합니다. Amazon을 사용하여 실행 중인 인스턴스를 모니터링할 수 CloudWatch 있습니다. 인스턴스에 로그인하고 대화형 명령줄 도구를 사용하여 실행 중인 인스턴스의 로그를 검사할 수 있습니다. 하지만 기본적으로 CloudWatch 메트릭 데이터는 인스턴스가 종료된 후 제한된 기간 동안만 보존되며, 인스턴스 로그는 일반적으로 인스턴스를 지원하는 EBS 볼륨과 함께 삭제됩니다. 시작 템플릿을 사용하여 시작된 인스턴스의 메트릭이나 로깅 데이터를 보존하려면 시작 템플릿을 사용하여 인스턴스에 CloudWatch EC2 에이전트를 구성할 수 있습니다. PCS 이 항목에서는 실행 중인 인스턴스 모니터링에 대한 개요를 제공하고 영구 인스턴스 측정치 및 로그를 구성하는 방법의 예를 제공합니다.

실행 중인 인스턴스 모니터링

AWSPCS인스턴스 찾기

에서 시작된 인스턴스를 모니터링하려면 클러스터 또는 컴퓨팅 노드 그룹과 연결된 실행 중인 인스턴스를 찾으십시오. PCS 그런 다음 EC2 콘솔에서 해당 인스턴스의 상태 및 경보와 모니터링 섹션을 살펴보세요. 해당 인스턴스에 대한 로그인 액세스가 구성된 경우 인스턴스에 연결하여 인스턴스의 다양한 로그 파일을 검사할 수 있습니다. 어떤 인스턴스가 관리되는지 식별하는 방법에 대한 자세한 내용은을 PCS 참조하십시오에서 컴퓨팅 노드 그룹 인스턴스 찾기 AWS PCS.

세부 지표 활성화

기본적으로 인스턴스 지표는 5분 간격으로 수집됩니다. 1분 간격으로 지표를 수집하려면 컴퓨팅 노드 그룹 시작 템플릿에서 세부 CloudWatch 모니터링을 활성화하십시오. 자세한 내용은 세부 CloudWatch 모니터링 켜기 단원을 참조하십시오.

영구 인스턴스 지표 및 로그 구성

Amazon CloudWatch 에이전트를 설치하고 구성하여 인스턴스의 지표와 로그를 유지할 수 있습니다. 이는 세 가지 주요 단계로 구성됩니다.

1. CloudWatch 에이전트 구성을 생성합니다.
2. PCS인스턴스에서 검색할 수 있는 위치에 구성을 저장합니다.
3. CloudWatch 에이전트 소프트웨어를 설치하고, 구성을 가져오고, 구성을 사용하여 CloudWatch 에이전트를 시작하는 EC2 시작 템플릿을 작성하십시오.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트를 통한 지표, 로그 및 추적 수집을 참조하십시오Amazon EC2 시작 템플릿을 다음과 같이 사용하기 AWS PCS.

CloudWatch 에이전트 구성 생성

CloudWatch 에이전트를 인스턴스에 배포하기 전에 수집할 지표, 로그 및 추적을 지정하는 JSON 구성 파일을 생성해야 합니다. 구성 파일은 마법사를 사용하거나 텍스트 편집기를 사용하여 수동으로 생성할 수 있습니다. 이 데모에서는 구성 파일을 수동으로 생성합니다.

AWSCLI설치한 컴퓨터에 다음 내용이 포함된 config.json이라는 CloudWatch 구성 파일을 생성합니다. 다음을 URL 사용하여 파일 사본을 다운로드할 수도 있습니다.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

참고

- 샘플 파일의 로그 경로는 Amazon Linux 2용입니다. 인스턴스에서 다른 기본 운영 체제를 사용할 경우 경로를 적절하게 변경하십시오.
- 다른 로그를 캡처하려면 아래에 항목을 더 추가하십시오`collect_list`.
- 의 `{brackets}` 값은 템플릿 변수입니다. 지원되는 변수의 전체 목록은 Amazon CloudWatch User Guide의 CloudWatch 에이전트 구성 파일 수동 생성 또는 편집을 참조하십시오.
- 이러한 정보 유형을 `logs` 생략하거나 수집하지 않으려는 `metrics` 경우 선택할 수 있습니다.

```
{
    "agent": {
        "metrics_collection_interval": 60
    },
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/var/log/cloud-init.log",
                        "log_group_class": "STANDARD",
                        "log_group_name": "/PCSLogs/instances",
                        "log_stream_name": "{instance_id}.cloud-init.log",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/cloud-init-output.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.cloud-init-output.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/amazon/pcs/bootstrap.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.bootstrap.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/slurmd.log",
                        "log_group_class": "STANDARD",
```

```
                    "log_stream_name": "{instance_id}.slurmd.log",
                    "log_group_name": "/PCSLogs/instances",
                    "retention_in_days": 30
                },
                {
                    "file_path": "/var/log/messages",
                    "log_group_class": "STANDARD",
                    "log_stream_name": "{instance_id}.messages",
                    "log_group_name": "/PCSLogs/instances",
                    "retention_in_days": 30
                },
                {
                    "file_path": "/var/log/secure",
                    "log_group_class": "STANDARD",
                    "log_stream_name": "{instance_id}.secure",
                    "log_group_name": "/PCSLogs/instances",
                    "retention_in_days": 30
                }
            ]
        }
    }
},
"metrics": {
    "aggregation_dimensions": [
        [
            "InstanceId"
        ]
    ],
    "append_dimensions": {
        "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
        "ImageId": "${aws:ImageId}",
        "InstanceId": "${aws:InstanceId}",
        "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
        "cpu": {
            "measurement": [
                "cpu_usage_idle",
                "cpu_usage_iowait",
                "cpu_usage_user",
                "cpu_usage_system"
            ],
            "metrics_collection_interval": 60,
            "resources": [
```

```
                    "*"
                ],
                "totalcpu": false
            },
            "disk": {
                "measurement": [
                    "used_percent",
                    "inodes_free"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "diskio": {
                "measurement": [
                    "io_time"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "mem": {
                "measurement": [
                    "mem_used_percent"
                ],
                "metrics_collection_interval": 60
            },
            "swap": {
                "measurement": [
                    "swap_used_percent"
                ],
                "metrics_collection_interval": 60
            }
        }
    }
}
```

이 파일은 CloudWatch 에이전트가 인스턴스 부트스트래핑, 인증 및 로그인, 기타 문제 해결 도메인의 오류를 진단하는 데 도움이 될 수 있는 여러 파일을 모니터링하도록 지시합니다. 다음이 포함됩니다.

- `/var/log/cloud-init.log`— 인스턴스 구성 초기 단계의 출력

- `/var/log/cloud-init-output.log`— 인스턴스 구성 중에 실행되는 명령의 출력
- `/var/log/amazon/pcs/bootstrap.log`— 인스턴스 PCS 구성 중에 실행되는 특정 작업의 출력
- `/var/log/slurmd.log`— Slurm 워크로드 매니저의 데몬 slurmd의 출력
- `/var/log/messages`— 커널, 시스템 서비스 및 애플리케이션의 시스템 메시지
- `/var/log/secure`— sudo 및 기타 보안 이벤트와 같은 SSH 인증 시도와 관련된 로그

이라는 `/PCSLogs/instances` 로그 그룹에 CloudWatch 로그 파일이 전송됩니다. 로그 스트림은 인스턴스 ID와 로그 파일의 기본 이름을 조합한 것입니다. 로그 그룹의 보존 기간은 30일입니다.

또한 파일은 CloudWatch 에이전트가 몇 가지 일반적인 지표를 수집하여 인스턴스 ID별로 집계하도록 지시합니다.

구성을 저장합니다.

CloudWatch 에이전트 구성 파일은 PCS 컴퓨팅 노드 인스턴스가 액세스할 수 있는 위치에 저장해야 합니다. 이 작업을 수행하는 일반적인 두 가지 방법이 있습니다. 컴퓨팅 노드 그룹 인스턴스가 인스턴스 프로필을 통해 액세스할 수 있는 Amazon S3 버킷에 업로드할 수 있습니다. 또는 Amazon Systems Manager 파라미터 스토어에 SSM 파라미터로 저장할 수도 있습니다.

S3 버킷에 업로드

S3에 파일을 저장하려면 다음 AWS CLI 명령을 사용하십시오. 명령을 실행하기 전에 다음과 같이 대체하십시오.

- Replace *DOC-EXAMPLE-BUCKET* 고유한 S3 버킷 이름으로

먼저 (기존 버킷이 있는 경우 선택 사항) 구성 파일을 보관할 버킷을 생성합니다.

```
aws s3 mb s3://DOC-EXAMPLE-BUCKET
```

다음으로 파일을 버킷에 업로드합니다.

```
aws s3 cp ./config.json s3://DOC-EXAMPLE-BUCKET/
```

SSM매개변수로 저장

파일을 SSM 매개변수로 저장하려면 다음 명령을 사용합니다. 명령을 실행하기 전에 다음과 같이 바꾸십시오.

- Replace *region-code* 작업 중인 AWS 지역과 함께 AWSPCS.
- (선택 사항) 교체 *AmazonCloudWatch-PCS* 파라미터에 사용자 이름을 입력하세요. 참고로 이름의 접두사를 변경하는 경우 노드 그룹 인스턴스 프로필의 SSM 파라미터에 대한 읽기 권한을 구체적으로 추가해야 합니다. AmazonCloudWatch-

```
aws ssm put-parameter \
   --region region-code \
   --name "AmazonCloudWatch-PCS" \
   --type String \
   --value file://config.json
```

EC2시작 템플릿을 작성하세요.

시작 템플릿의 구체적인 세부 정보는 구성 파일이 S3에 저장되어 있는지 또는 S3에 저장되어 있는지에 따라 달라집니다SSM.

S3에 저장된 구성을 사용하십시오.

이 스크립트는 CloudWatch 에이전트를 설치하고, S3 버킷에서 구성 파일을 가져온 다음, CloudWatch 에이전트를 시작합니다. 이 스크립트의 다음 값을 사용자 세부 정보로 바꾸십시오.

- *DOC-EXAMPLE-BUCKET* — 계정에서 읽을 수 있는 S3 버킷의 이름
- */config.json* — 구성이 저장된 S3 버킷 루트를 기준으로 한 경로

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-cloudwatch-agent

runcmd:
- aws s3 cp s3://DOC-EXAMPLE-BUCKET/config.json /etc/s3-cw-config.json
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
 ec2 -s -c file://etc/s3-cw-config.json

--==MYBOUNDARY==--
```

노드 그룹의 IAM 인스턴스 프로필에 버킷에 대한 액세스 권한이 있어야 합니다. 다음은 위의 사용자 데이터 스크립트에 있는 버킷 IAM 정책 예시입니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

또한 인스턴스는 S3 및 CloudWatch 엔드포인트로 향하는 아웃바운드 트래픽을 허용해야 합니다. 클러스터 아키텍처에 따라 보안 그룹 또는 VPC 엔드포인트를 사용하여 이 작업을 수행할 수 있습니다.

에 저장된 구성을 사용하십시오. SSM

이 스크립트는 CloudWatch 에이전트를 설치하고, SSM 매개변수에서 구성 파일을 가져온 다음, 구성 파일을 사용하여 CloudWatch 에이전트를 시작합니다. 이 스크립트의 다음 값을 사용자 세부 정보로 바꾸십시오.

- (선택 사항) 바꾸기 *AmazonCloudWatch-PCS* 파라미터에 사용자 이름을 입력하세요.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-cloudwatch-agent

runcmd:
```

```
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
 ec2 -s -c ssm:AmazonCloudWatch-PCS

--==MYBOUNDARY==--
```

노드 그룹의 IAM 인스턴스 정책에는 해당 정책이 CloudWatchAgentServerPolicy연결되어 있어야 합니다.

파라미터 이름이 로 AmazonCloudWatch- 시작하지 않는 경우 노드 그룹 인스턴스 프로필의 SSM 파라미터에 대한 읽기 권한을 구체적으로 추가해야 합니다. 다음은 접두사에 대해 이를 설명하는 예제 IAM 정책입니다.*DOC-EXAMPLE-PREFIX*.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "CustomCwSsmMParamReadOnly",
      "Effect" : "Allow",
      "Action" : [
        "ssm:GetParameter"
      ],
      "Resource" : "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

또한 인스턴스는 SSM 및 CloudWatch 엔드포인트로 향하는 아웃바운드 트래픽을 허용해야 합니다. 클러스터 아키텍처에 따라 보안 그룹이나 VPC 엔드포인트를 사용하여 이 작업을 수행할 수 있습니다.

를 사용하여 AWS 병렬 컴퓨팅 서비스 API 호출 로깅 AWS CloudTrail

AWS PCS에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스와 통합됩니다 AWS PCS. CloudTrail AWS PCSas에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 AWS PCS 콘솔에서의 호출과 AWS PCS API 작업에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 에 대한 이벤트를 포함하여 Amazon S3 버킷으로 CloudTrail 이벤트를 지속적으로 전송할 수 AWS PCS 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 요청을 받은 사람 AWS PCS, 요청한 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

자세한 CloudTrail 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

AWS PCS자세한 내용은 CloudTrail

CloudTrail 계정을 만들 AWS 계정 때 활성화됩니다. 에서 AWS PCS 활동이 발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. 내 페이지에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 이벤트 기록으로 CloudTrail 이벤트 보기를 참조하십시오.

에 대한 이벤트를 포함하여 내 이벤트의 진행 중인 기록을 보려면 AWS PCS 트레일을 생성하십시오 AWS 계정. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 다음에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신

모든 AWS PCS 작업은 AWS 병렬 컴퓨팅 서비스 API 참조에 의해 CloudTrail 기록되고 문서화됩니다. 예를 들어 `CreateComputeNodeGroupUpdateQueue`, 및 `DeleteCluster` 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail userIdentity요소를 참조하십시오.

의 CloudTrail 로그 파일 항목 이해 AWS PCS

트레일은 지정한 S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 `CreateQueue` 작업에 대한 CloudTrail 로그 항목을 보여줍니다.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
        "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
        "accountId": "012345678910",
        "accessKeyId": "ASIAY36PTPIEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAY36PTPIEEXAMPLE",
                "arn": "arn:aws:iam::012345678910:role/Admin",
                "accountId": "012345678910",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2024-07-16T17:05:51Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-07-16T17:13:09Z",
    "eventSource": "pcs.amazonaws.com",
    "eventName": "CreateQueue",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36",
    "requestParameters": {
        "clientToken": "c13b7baf-2894-42e8-acec-example",
        "clusterIdentifier": "abcdef0123",
        "computeNodeGroupConfigurations": [
            {
```

```
                "computeNodeGroupId": "abcdef0123"
            }
        ],
        "queueName": "all"
    },
    "responseElements": {
        "queue": {
            "arn": "arn:aws:pcs:us-east-1:609783872011:cluster/abcdef0123/queue/
abcdef0123",
            "clusterId": "abcdef0123",
            "computeNodeGroupConfigurations": [
                {
                    "computeNodeGroupId": "abcdef0123"
                }
            ],
            "createdAt": "2024-07-16T17:13:09.276069393Z",
            "id": "abcdef0123",
            "modifiedAt": "2024-07-16T17:13:09.276069393Z",
            "name": "all",
            "status": "CREATING"
        }
    },
    "requestID": "a9df46d7-3f6d-43a0-9e3f-example",
    "eventID": "7ab18f88-0040-47f5-8388-example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "012345678910",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "pcs.us-east-1.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

엔드포인트 및 서비스 할당량 AWS PCS

다음 섹션에서는 병렬 컴퓨팅 서비스 () 의 엔드포인트 및 서비스 할당량에 대해 AWS 설명합니다.AWS PCS 이전에 한도라고 불렸던 서비스 할당량은 서비스 리소스 또는 운영의 최대 수입니다. AWS 계정

AWS 계정 Y에는 각 서비스에 대한 기본 할당량이 있습니다. AWS 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다. 일부 할당량에 대한 증가를 요청할 수 있으며 다른 할당량은 늘릴 수 없습니다.

자세한 내용을 알아보려면 AWS 일반 참조의 AWS 서비스 할당량을 참조하세요.

목차

Service endpoints

지역명	지역	엔드포인트	프로토콜
미국 동부(버지니아 북부)	us-east-1	pcs.us-east-1.amazonaws.com	HTTPS
미국 동부(오하이오)	us-east-2	pcs.us-east-2.amazonaws.com	HTTPS
미국 서부(오레곤)	us-west-2	pcs.us-west-2.amazonaws.com	HTTPS
아시아 태평양(싱가포르)	ap-southeast-1	pcs.ap-southeast-1.amazonaws.com	HTTPS
아시아 태평양(시드니)	ap-southeast-2	pcs.ap-southeast-2.amazonaws.com	HTTPS

지역명	지역	엔드포인트	프로토콜
아시아 태평양(도쿄)	ap-northeast-1	pcs.ap-northeast-1.amazonaws.com	HTTPS
유럽(프랑크푸르트)	eu-central-1	pcs.eu-central-1.amazonaws.com	HTTPS
유럽(아일랜드)	eu-west-1	pcs.eu-west-1.amazonaws.com	HTTPS
유럽(스톡홀름)	eu-north-1	pcs.eu-north-1.amazonaws.com	HTTPS

Service quotas

이름	기본값	조정 가능	설명
클러스터	5	예	1개당 최대 클러스터 AWS 리전수

> **Note**
>
> 기본값은 에서 설정한 초기 할당량입니다. AWS이러한 기본값은 실제 적용된 할당량 값 및 가능한 최대 서비스 할당량과 별개입니다. 자세한 내용은 Service Quotas 사용 설명서의 Service Quotas 용어를 참조하십시오.

이러한 서비스 할당량은 의 AWS 병렬 컴퓨팅 서비스 () PCS 에 나열되어 있습니다. AWS Management Console 조정 가능한 것으로 표시된 값에 대해 할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하십시오.

> **Important**
>
> 에서 현재 AWS 리전 설정을 확인하는 것을 잊지 마세요. AWS Management Console

내부 할당량

다음 할당량은 내부 할당량이며 조정할 수 없습니다.

이름	기본값	조정 가능	설명
동시 클러스터 생성	1	아니요	`Creating`주별 AWS 리전최대 클러스터 수입니다.

다른 서비스의 관련 할당량 AWS

AWS PCS다른 AWS 서비스를 사용합니다. 해당 서비스의 서비스 할당량은 서비스 사용에 영향을 미칩니다. AWS PCS

영향을 EC2 미치는 Amazon 서비스 할당량 AWS PCS

- 스팟 인스턴스 요청
- 온디맨드 인스턴스 실행
- 시작 템플릿
- 시작 템플릿 버전
- 아마존 EC2 API 요청

자세한 내용은 Amazon Elastic Compute 클라우드 사용 설명서의 Amazon EC2 서비스 할당량을 참조하십시오.

AWS PCS샘플용 릴리스 노트 AMIs

AWS PCSAMIs샘플에는 보안 패치의 야간 릴리스 주기가 있습니다. 이러한 증분 보안 패치는 공식 릴리스 노트에는 포함되어 있지 않습니다.

> **Important**
>
> AMIs샘플은 데모용이며 프로덕션 워크로드에는 권장되지 않습니다.

목차

AWS PCS슬럼 AMI 23.11용 샘플 x86_64 (아마존 리눅스 2)

이 문서에서는 AWS PCS 샘플 x86_64 AMI (Amazon Linux 2) 의 최신 변경 사항, 추가 사항, 알려진 문제 및 수정 사항에 대해 설명합니다.

- 생성 날짜: 2024년 7월 15일
- 출시일: 2024년 8월 22일
- 최근 업데이트: 2024년 8월 22일

AMI이름

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`

지원되는 EC2 인스턴스

- 64비트 x86 프로세서가 장착된 모든 인스턴스. 호환되는 인스턴스를 찾으려면 Amazon EC2 콘솔로 이동하십시오. 인스턴스 유형을 선택한 다음 `Architectures=x86_64` 검색하십시오.

AMI목차

- 지원되는 AWS 서비스: AWS PCS

- 운영 체제: 아마존 리눅스 2
- 컴퓨팅 아키텍처: x86_64
- 리눅스 커널: 5.10.220-209.867.amzn2.x86_64
- EBS볼륨 유형: gp2
- AWS PCS슬럼 23.11 인스톨러: 23.11.9-1
- AWS PCS소프트웨어 설치 프로그램: 1.0.0-1
- EFA인스톨러: 1.33.0
- GDRCopy: 2.4
- NVIDIA드라이버: 535.154.05
- NVIDIACUDA: 12.2.2_535.104.05

고지 사항

- None

출시일: 2024-08-22

Updated

- 없음. 첫 출시.

추가됨

- 없음. 첫 출시.

제거됨

- 없음. 첫 출시.

AWS PCS슬럼 AMI 23.11용 샘플 Arm64 (아마존 리눅스 2)

이 문서에서는 AWS PCS 샘플 Arm64 (AMIAmazon Linux 2) 의 최신 변경 사항, 추가 사항, 알려진 문제 및 수정 사항에 대해 설명합니다.

- 생성 날짜: 2024년 7월 15일

- 출시일: 2024년 8월 22일
- 최근 업데이트: 2024년 8월 22일

AMI이름

- `aws-pcs-sample_ami-amzn2-arm64-slurm-23.11`

지원되는 EC2 인스턴스

- 64비트 Arm 프로세서가 설치된 모든 인스턴스 호환되는 인스턴스를 찾으려면 Amazon EC2 콘솔로 이동하십시오. 인스턴스 유형을 선택한 다음 `Architectures=arm64` 검색하십시오.

AMI목차

- 지원되는 AWS 서비스: AWS PCS
- 운영 체제: 아마존 리눅스 2
- 컴퓨팅 아키텍처: arm64
- 리눅스 커널: 5.10.220-209.867.amzn2.aarch64
- EBS볼륨 유형: gp2
- AWS PCS슬럼 23.11 인스톨러: 23.11.9-1
- AWS PCS소프트웨어 설치 프로그램: 1.0.0-1
- EFA인스톨러: 1.33.0
- GDRCopy: 2.4
- NVIDIA드라이버: 535.154.05
- NVIDIACUDA: 12.2.2_535.104.05

고지 사항

- None

출시일: 2024-08-22

Updated

- 없음. 첫 출시.

추가됨

- 없음. 첫 출시.

제거됨

- 없음. 첫 출시.

AWS PCS 사용 설명서 기록

다음 표에는 의 설명서 릴리스가 설명되어 AWS PCS 있습니다.

날짜	변경 사항	설명서 업데이트	API버전 업데이트
2024년 8월 28일	관리형 정책 페이지 추가됨	자세한 내용은 AWSAWS 병렬 컴퓨팅 서비스에 대한 관리형 정책 단원을 참조하십시오.	N/A
2024년 8월 28일	AWS PCS릴리즈	AWS PCS사용 설명서의 최초 릴리스.	AWS SDK: 2024-08-28

AWS 용어집

최신 AWS 용어는 참조의 AWS 용어집을 참조하십시오.AWS 용어집

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.