aws

Panduan Pengguna

AWS PCS

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

AWS PCS: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin atau mungkin tidak berafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Layanan Komputasi AWS Paralel?

AWS Parallel Computing Service (AWS PCS) adalah layanan terkelola yang membuatnya lebih mudah untuk menjalankan dan menskalakan beban kerja komputasi (HPC) kinerja tinggi, dan membangun model ilmiah dan teknik AWS menggunakan Slurm. Gunakan AWS PCS untuk membangun cluster komputasi yang paling terintegrasi dalam AWS komputasi kelas, penyimpanan, jaringan, dan visualisasi. Jalankan simulasi atau bangun model ilmiah dan teknik. Sederhanakan dan sederhanakan operasi klaster Anda menggunakan kemampuan manajemen dan observabilitas bawaan. Berdayakan pengguna Anda untuk fokus pada penelitian dan inovasi dengan memungkinkan mereka menjalankan aplikasi dan pekerjaan mereka di lingkungan yang akrab.

Konsep utama

Sebuah cluster AWS PCS memiliki 1 atau lebih antrian, terkait dengan setidaknya 1 grup node komputasi. Pekerjaan dikirimkan ke antrian dan dijalankan pada EC2 instance yang ditentukan oleh grup node komputasi. Anda dapat menggunakan fondasi ini untuk menerapkan HPC arsitektur canggih.

Klaster

Cluster adalah sumber daya untuk mengelola sumber daya dan menjalankan beban kerja. Cluster adalah AWS PCS sumber daya yang mendefinisikan perakitan konfigurasi komputasi, jaringan, penyimpanan, identitas, dan penjadwal pekerjaan. Anda membuat klaster dengan menentukan penjadwal pekerjaan mana yang ingin Anda gunakan (Slurm saat ini), konfigurasi penjadwal apa yang Anda inginkan, pengontrol layanan apa yang ingin Anda kelola cluster, dan di mana VPC Anda ingin sumber daya cluster diluncurkan. Penjadwal menerima dan menjadwalkan pekerjaan, dan juga meluncurkan node komputasi (EC2instance) yang memproses pekerjaan tersebut.

Hitung grup node

Grup node komputasi adalah kumpulan node komputasi yang AWS PCS digunakan untuk menjalankan pekerjaan atau menyediakan akses interaktif ke cluster. Saat menentukan grup node komputasi, Anda menentukan ciri umum seperti jenis EC2 instans Amazon, jumlah instans minimum dan maksimum, VPC subnet target, Amazon Machine Image (AMI), opsi pembelian, dan konfigurasi peluncuran kustom. AWS PCSmenggunakan pengaturan ini untuk secara efisien meluncurkan, mengelola, dan menghentikan node komputasi dalam grup node komputasi.

Antrean

Ketika Anda ingin menjalankan pekerjaan pada cluster tertentu, Anda mengirimkannya ke antrian tertentu (juga kadang-kadang disebut partisi). Pekerjaan tetap dalam antrian sampai AWS PCS menjadwalkannya untuk berjalan pada grup node komputasi. Anda mengaitkan satu atau beberapa grup node komputasi dengan setiap antrian. Antrian diperlukan untuk menjadwalkan dan mengeksekusi pekerjaan pada sumber daya grup node komputasi yang mendasarinya menggunakan berbagai kebijakan penjadwalan yang ditawarkan oleh penjadwal pekerjaan. Pengguna tidak mengirimkan pekerjaan langsung ke node komputasi atau grup node komputasi.

Administrator sistem

Administrator sistem menyebarkan, memelihara, dan mengoperasikan cluster. Mereka dapat mengakses AWS PCS melalui AWS Management Console, AWS PCSAPI, dan AWS SDK. Mereka memiliki akses ke cluster tertentu melalui SSH atau AWS Systems Manager, di mana mereka dapat menjalankan tugas administratif, menjalankan pekerjaan, mengelola data, dan melakukan aktivitas berbasis shell lainnya. Untuk informasi selengkapnya, lihat Dokumentasi AWS Systems Manager.

Pengguna akhir

Pengguna akhir tidak memiliki day-to-day tanggung jawab untuk menyebarkan atau mengoperasikan klaster. Mereka menggunakan antarmuka terminal (sepertiSSH) untuk mengakses sumber daya cluster, menjalankan pekerjaan, mengelola data, dan melakukan aktivitas berbasis shell lainnya.

Menyiapkan Layanan Komputasi AWS Paralel

Selesaikan tugas-tugas berikut untuk menyiapkan AWS Parallel Computing Service (AWS PCS).

Topik

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Aktifkan otentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan MFA perangkat virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan IAM Pengguna.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat IAM Identitas.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat IAM Identitas, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat IAM Identitas, gunakan login URL yang dikirim ke alamat email saat Anda membuat pengguna Pusat IAM Identitas.

  Untuk bantuan masuk menggunakan pengguna Pusat IAM Identitas, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat IAM Identitas, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Instal AWS CLI

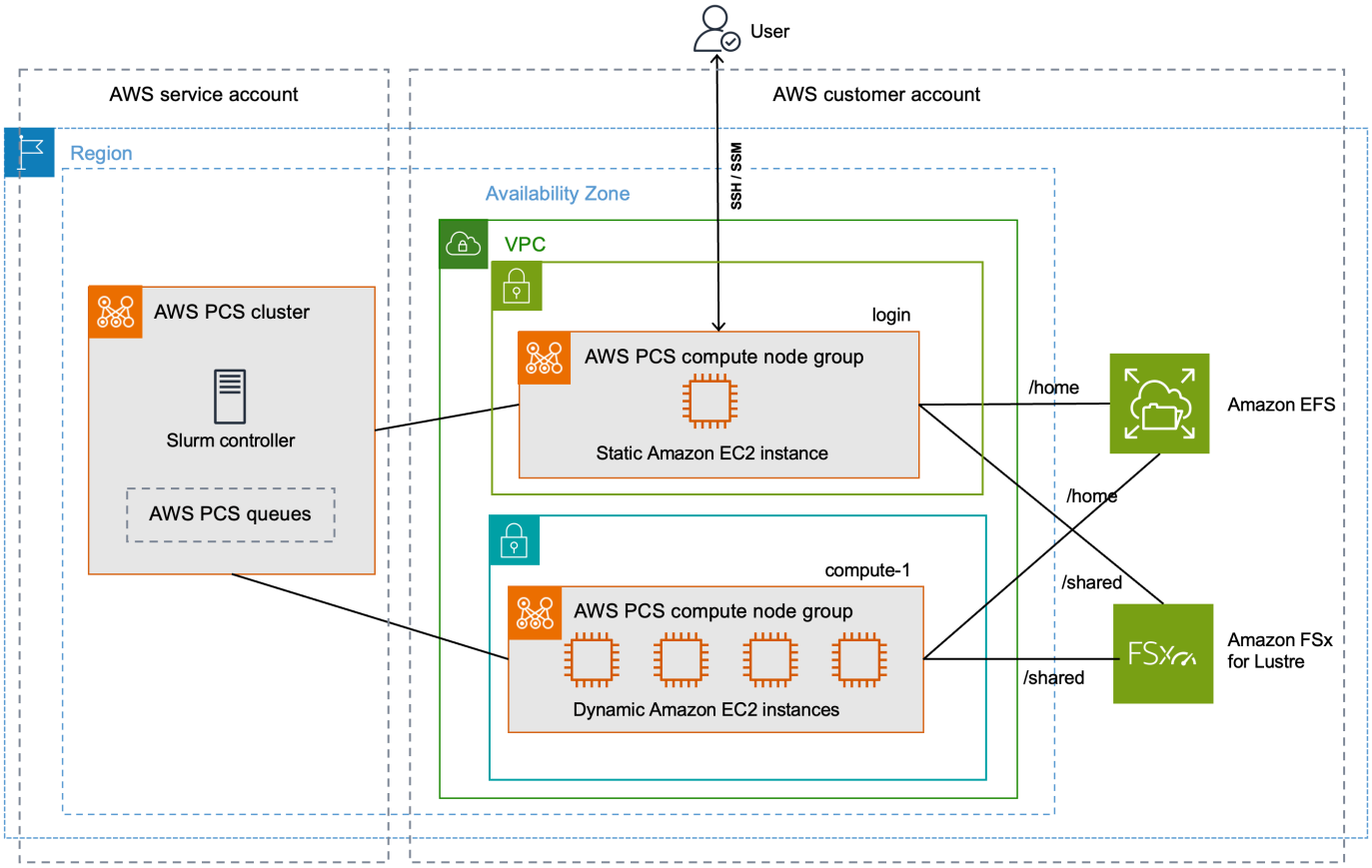
Anda harus menggunakan versi terbaru dari AWS CLI. Untuk selengkapnya, lihat Menginstal atau memperbarui ke versi terbaru dari Panduan AWS Command Line Interface Pengguna untuk Versi 2. AWS CLI

Masukkan perintah berikut pada prompt perintah untuk memeriksa Anda AWS CLI; itu harus menampilkan informasi bantuan.

```
aws pcs help
```

Memulai dengan AWS PCS

Ini adalah tutorial untuk membuat cluster sederhana yang dapat Anda gunakan untuk mencoba AWS PCS. Gambar berikut menunjukkan desain cluster.



Desain cluster tutorial memiliki komponen kunci berikut:

- A VPC dan subnet yang memenuhi persyaratan AWS PCS jaringan.
- Sistem EFS file Amazon, yang akan digunakan sebagai direktori home bersama.
- Sistem file Amazon FSx untuk Lustre, yang menyediakan direktori kinerja tinggi bersama.
- Sebuah AWS PCS cluster, yang menyediakan controller Slurm.
- 2 menghitung grup node.
  - Grup `login` node, yang menyediakan akses interaktif berbasis shell ke sistem.
  - Grup `compute-1` node menyediakan instance penskalaan elastis untuk menjalankan pekerjaan.
- 1 antrian yang mengirimkan pekerjaan ke EC2 instance di grup `compute-1` node.

Cluster memerlukan AWS sumber daya tambahan, seperti grup keamanan, IAM peran, dan templat EC2 peluncuran, yang tidak ditampilkan dalam diagram.

Topik

Prasyarat untuk memulai AWS PCS

Sebelum Anda memulai tutorial ini, instal dan konfigurasikan alat dan sumber daya berikut yang Anda butuhkan untuk membuat dan mengelola AWS PCS cluster.

- AWS CLI— Alat baris perintah untuk bekerja dengan AWS layanan, termasuk AWS PCS. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui ke versi terbaru dari Panduan AWS Command Line Interface Pengguna untuk Versi 2. AWS CLI Setelah menginstal AWS CLI, kami sarankan Anda juga mengkonfigurasinya. Untuk informasi selengkapnya, lihat Mengkonfigurasi AWS CLI dalam Panduan AWS Command Line Interface Pengguna untuk Versi 2.
- IAMIzin yang diperlukan — Prinsip IAM keamanan yang Anda gunakan harus memiliki izin untuk bekerja dengan AWS PCS IAM peran, peran terkait layanan, AWS CloudFormation VPC, dan sumber daya terkait. Untuk informasi selengkapnyaIdentity and Access Management untuk Layanan Komputasi AWS Paralel, lihat, dan Membuat peran terkait layanan di AWS Identity and Access Management Panduan Pengguna. Anda harus menyelesaikan semua langkah dalam

panduan ini sebagai pengguna yang sama. Untuk memeriksa pengguna saat ini, jalankan perintah berikut:

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

- Kami menyarankan Anda menyelesaikan langkah-langkah baris perintah dalam topik ini di shell Bash. Jika Anda tidak menggunakan shell Bash, beberapa perintah skrip seperti karakter kelanjutan baris dan cara variabel diatur dan digunakan memerlukan penyesuaian untuk shell Anda. Selain itu, aturan mengutip dan melarikan diri untuk shell Anda mungkin berbeda. Untuk informasi selengkapnya, lihat Tanda kutip dan literal dengan string AWS CLI di Panduan AWS Command Line Interface Pengguna untuk Versi 2.

Buat VPC dan subnet untuk AWS PCS

Anda dapat membuat VPC dan subnet dengan CloudFormation template. Gunakan yang berikut ini URL untuk mengunduh CloudFormation templat, lalu unggah templat di AWS CloudFormation konsol untuk membuat CloudFormation tumpukan baru. Untuk informasi selengkapnya, lihat Menggunakan AWS CloudFormation konsol di Panduan AWS CloudFormation Pengguna.

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

Dengan templat terbuka di AWS CloudFormation konsol, masukkan opsi berikut. Anda dapat menggunakan nilai default yang disediakan dalam template.

- Di bawah Berikan nama tumpukan:
  - Di bawah nama Stack, masukkan:

    ```
    hpc-networking
    ```
- Di bawah Parameter:
  - Di bawah VPC:
    - Di bawah CidrBlock, masukkan:

      ```
      10.3.0.0/16
      ```
  - Di bawah Subnet A:
    - Di bawah CidrPublicSubnetA, masukkan:

```
10.3.0.0/20
```

- Di bawah CidrPrivateSubnetA, masukkan:

```
10.3.128.0/20
```

- Di bawah Subnet B:
  - Di bawah CidrPublicSubnetB, masukkan:

```
10.3.16.0/20
```

  - Di bawah CidrPrivateSubnetB, masukkan:

```
10.3.144.0/20
```

- Di bawah Subnet C:
  - Untuk ProvisionSubnetsC, pilih True
  - Di bawah CidrPublicSubnetC, masukkan:

```
10.3.32.0/20
```

  - Di bawah CidrPrivateSubnetC, masukkan:

```
10.3.160.0/20
```

- Di bawah Kemampuan:
  - Centang kotak untuk saya akui yang AWS CloudFormation mungkin membuat IAM sumber daya.

Pantau status CloudFormation tumpukan. Saat mencapai`CREATE_COMPLETE`, temukan ID untuk grup keamanan default di yang baruVPC. Anda menggunakan ID nanti dalam tutorial.

Temukan grup keamanan default untuk klaster VPC

Untuk menemukan ID untuk grup keamanan default di yang baruVPC, ikuti prosedur ini:

- Arahkan ke VPCkonsol Amazon.
- Di bawah VPCDasbor, pilih Filter menurut VPC.
  - Pilih di VPC mana nama dimulai`hpc-networking`.

    - Di bawah Keamanan, pilih Grup keamanan.
- Temukan ID grup Keamanan untuk grup bernama`default`. Ini memiliki deskripsi`default VPC security group`. Anda menggunakan ID nanti untuk mengonfigurasi templat EC2 peluncuran.

Buat grup keamanan untuk AWS PCS

AWS PCSbergantung pada grup keamanan untuk mengelola lalu lintas jaringan masuk dan keluar dari cluster dan grup node komputasinya. Untuk informasi rinci tentang topik ini, lihatPersyaratan dan pertimbangan kelompok keamanan.

Pada langkah ini, Anda akan menggunakan CloudFormation templat untuk dua grup keamanan.

- Grup keamanan cluster, yang memungkinkan komunikasi antara AWS PCS controller, node komputasi, dan node login.
- Grup SSH keamanan masuk, yang dapat Anda tambahkan secara opsional ke node login Anda untuk mendukung akses SSH

Buat grup keamanan untuk AWS PCS

Anda dapat membuat VPC dan subnet dengan CloudFormation template ini. Gunakan yang berikut ini URL untuk mengunduh CloudFormation templat, lalu unggah templat di AWS CloudFormation konsol untuk membuat CloudFormation tumpukan baru. Untuk informasi selengkapnya, lihat Menggunakan AWS CloudFormation konsol di Panduan AWS CloudFormation Pengguna.

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

Dengan templat terbuka di AWS CloudFormation konsol, masukkan opsi berikut. Perhatikan bahwa beberapa opsi akan diisi sebelumnya di template - Anda cukup membiarkannya sebagai nilai default.

- Di bawah Berikan nama tumpukan
    - Di bawah nama Stack, masukkan:

      ```
      getstarted-sg
      ```
- Di bawah Parameter

- Di bawah VpcId, pilih VPC tempat nama dimulaihpc-networking.
- (Opsional) Di bawah ClientIpCidr, masukkan rentang IP yang lebih ketat untuk grup SSH keamanan masuk. Kami menyarankan Anda membatasi ini dengan IP/subnet Anda sendiri (x.x.x.x/32 untuk ip Anda sendiri atau x.x.x.x/24 untuk jangkauan. Ganti x.x.x.x dengan IP Anda sendiri. PUBLIC Anda bisa mendapatkan IP publik Anda menggunakan alat seperti https://ifconfig.co/)

Pantau status CloudFormation tumpukan. Ketika mencapai CREATE_COMPLETE sumber daya kelompok keamanan siap.

Ada dua grup keamanan yang dibuat, dengan nama:

- cluster-getstarted-sg— ini adalah kelompok keamanan cluster
- inbound-ssh-getstarted-sg— ini adalah grup keamanan untuk memungkinkan akses masuk SSH

Buat cluster di AWS PCS

Di AWS PCS, cluster adalah sumber daya persisten untuk mengelola sumber daya dan menjalankan beban kerja. Anda membuat klaster untuk penjadwal tertentu (AWS PCSsaat ini mendukung Slurm) di subnet baru atau yang sudah ada. VPC Cluster menerima dan menjadwalkan pekerjaan, dan juga meluncurkan node komputasi (EC2instance) yang memproses pekerjaan tersebut.

Untuk membuat klaster Anda

1. Buka AWS PCSkonsol dan pilih Buat cluster.
2. Di bagian Pengaturan cluster, masukkan bidang berikut:
   - Nama cluster — Enter get-started
   - Ukuran pengontrol - Pilih Kecil
3. Di bagian Jaringan, pilih nilai untuk bidang berikut:
   - VPC— Pilih yang VPC bernama hpc-networking:Large-Scale-HPC
   - Subnet — Pilih subnet tempat nama dimulai hpc-networking:PrivateSubnetA
   - Grup keamanan — Pilih grup keamanan klaster bernama cluster-getstarted-sg
4. Pilih Buat klaster.

> **Note**
>
> Bidang Status menunjukkan Membuat saat klaster sedang disediakan. Pembuatan cluster dapat memakan waktu beberapa menit.

Buat penyimpanan bersama AWS PCS di Amazon Elastic File System

Amazon Elastic File System (AmazonEFS) adalah AWS layanan yang menyediakan penyimpanan file tanpa server dan sepenuhnya elastis sehingga Anda dapat berbagi data file tanpa menyediakan atau mengelola kapasitas dan kinerja penyimpanan. Untuk informasi selengkapnya, lihat Apa itu Amazon Elastic File System? di Panduan Pengguna Amazon Elastic File System.

Cluster AWS PCS demonstrasi menggunakan sistem EFS file untuk menyediakan direktori home bersama antara node cluster. Buat sistem EFS file yang VPC sama dengan cluster Anda.

Untuk membuat sistem EFS file Amazon Anda

1. Pergi ke EFSkonsol Amazon.
2. Pastikan itu diatur ke Wilayah AWS tempat yang sama di mana Anda akan mencoba AWS PCS.
3. Pilih Buat sistem file.
4. Pada halaman Create file system, atur parameter berikut:
   - Untuk Nama, masukkan `getstarted-efs`
   - Di bawah Virtual Private Cloud (VPC), pilih VPC nama `hpc-networking:Large-Scale-HPC`
   - Pilih Buat. Ini mengembalikan Anda ke halaman sistem File.
5. Catat ID sistem File untuk sistem `getstarted-efs` file. Anda menggunakan informasi ini nanti.

Buat penyimpanan bersama untuk AWS PCS di Amazon FSx untuk Lustre

Amazon FSx for Lustre memudahkan dan hemat biaya untuk meluncurkan dan menjalankan sistem file Lustre yang populer dan berkinerja tinggi. Anda menggunakan Lustre untuk beban kerja di mana

kecepatan penting, seperti pembelajaran mesin, komputasi kinerja tinggi (HPC), pemrosesan video, dan pemodelan keuangan. Untuk informasi lebih lanjut, lihat Apa itu Amazon FSx untuk Kilau? di Amazon FSx untuk Panduan Pengguna Lustre.

Cluster AWS PCS demonstrasi dapat menggunakan sistem file FSx for Lustre untuk menyediakan direktori bersama berkinerja tinggi antara node cluster. Buat sistem file FSx for Lustre VPC sama dengan cluster Anda.

Untuk membuat sistem FSx file Lustre Anda

1. Buka FSxkonsol Amazon.
2. Pastikan konsol diatur untuk menggunakan yang Wilayah AWS sama dengan cluster Anda.
3. Pilih Buat sistem file.
   - Untuk Pilih jenis sistem file, pilih Amazon FSx untuk Lustre, lalu pilih Berikutnya.
4. Pada halaman Tentukan detail sistem file, atur parameter berikut:
   - Di bawah rincian sistem File
     - Untuk Nama, masukkan `getstarted-fsx`
     - Untuk jenis Deployment dan storage, pilih Persistent, SSD
     - Untuk Throughput per unit penyimpanan, pilih 125 MB/s/Tib
     - Untuk kapasitas Penyimpanan, masukkan 1,2 TiB
     - Untuk Konfigurasi Metadata, pilih Otomatis
     - Untuk tipe kompresi data, pilih LZ4
   - Di bawah Jaringan & keamanan
     - Untuk Virtual Private Cloud (VPC), pilih VPC nama `hpc-networking:Large-Scale-HPC`
     - Untuk Grup VPC Keamanan, tinggalkan grup keamanan bernama `default`
     - Untuk Subnet, pilih subnet tempat nama dimulai `hpc-networking:PrivateSubnetA`
   - Biarkan opsi lain diatur ke nilai defaultnya.
   - Pilih Berikutnya.
5. Pada halaman Tinjau dan buat, pilih Buat sistem file. Ini mengembalikan Anda ke halaman sistem File.
6. Arahkan ke halaman detail untuk sistem file FSx untuk Lustre yang Anda buat.
7. Catat ID sistem File dan nama Mount. Anda menggunakan informasi ini nanti.

> **Note**
>
> Bidang Status menunjukkan Membuat saat sistem file sedang disediakan. Pembuatan sistem file dapat memakan waktu beberapa menit. Tunggu sampai selesai sebelum melanjutkan dengan sisa tutorial.

Buat grup node komputasi di AWS PCS

Grup node komputasi adalah kumpulan virtual node komputasi (EC2instance) yang AWS PCS diluncurkan dan dikelola. Saat Anda menentukan grup node komputasi, Anda menentukan ciri umum seperti tipe EC2 instance, jumlah instans minimum dan maksimum, VPC subnet target, opsi pembelian pilihan, dan konfigurasi peluncuran kustom. AWS PCSsecara otomatis meluncurkan, mengelola, dan mengakhiri node komputasi dalam grup node komputasi, sesuai dengan pengaturan ini. Cluster demonstrasi menggunakan grup node komputasi untuk menyediakan node login untuk akses pengguna, dan grup node komputasi terpisah untuk memproses pekerjaan. Topik berikut menjelaskan prosedur untuk menyiapkan grup node komputasi ini di cluster Anda.

Topik

Buat profil instance untuk AWS PCS

Grup node komputasi memerlukan profil instance saat dibuat. Jika Anda menggunakan AWS Management Console untuk membuat peran untuk AmazonEC2, konsol secara otomatis membuat profil instance dan memberinya nama yang sama dengan peran tersebut. Untuk informasi selengkapnya, lihat Menggunakan profil instans di Panduan AWS Identity and Access Management Pengguna.

Dalam prosedur berikut, Anda menggunakan AWS Management Console untuk membuat peran untuk AmazonEC2, yang juga membuat profil instance untuk grup node komputasi Anda.

Untuk membuat profil peran dan contoh

- Navigasikan ke konsol IAM tersebut.
- Di bagian Manajemen akses, pilih Kebijakan.
  - Pilih Buat kebijakan.
  - Di bawah Tentukan izin, untuk editor Kebijakan, pilih JSON.
  - Ganti isi editor teks dengan yang berikut ini:

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

  - Pilih Berikutnya.
  - Di bawah Tinjau dan buat, untuk nama Kebijakan, masukkan`AWSPCS-getstarted-policy`.
  - Pilih Buat kebijakan.
- Di bawah Manajemen akses, pilih Peran.
- Pilih Buat peran.
- Di bawah Pilih entitas tepercaya:
  - Untuk jenis entitas Tepercaya, pilih AWS layanan
  - Di bawah Kasus penggunaan, pilih EC2.
    - Kemudian, di bawah Pilih kasus penggunaan untuk layanan yang ditentukan, pilih EC2.
  - Pilih Berikutnya.
- Di bawah Tambahkan izin:
  - Di kebijakan Izin, cari AWSPCS-getstarted-policy.
  - Centang kotak di samping AWSPCS-getstarted-policy untuk menambahkannya ke peran.
  - Di kebijakan Izin, cari A. mazonSSMManaged InstanceCore

- Centang kotak di samping A mazonSSMManaged InstanceCore untuk menambahkannya ke peran.
- Pilih Berikutnya.

- Di bawah Nama, tinjau, dan buat:
  - Di bawah Rincian Peran:
    - Untuk Nama peran, masukkan `AWSPCS-getstarted-role`.
  - Pilih Buat peran.

Buat template peluncuran untuk AWS PCS

Saat membuat grup node komputasi, Anda menyediakan template EC2 peluncuran yang AWS PCS digunakan untuk mengonfigurasi EC2 instance yang diluncurkan. Ini termasuk pengaturan seperti grup keamanan dan skrip yang berjalan saat instance diluncurkan.

Pada langkah ini, satu CloudFormation template akan digunakan untuk membuat dua template EC2 peluncuran. Satu template akan digunakan untuk membuat node login, dan yang lainnya akan digunakan untuk membuat node komputasi. Perbedaan utama di antara mereka adalah bahwa node login dapat dikonfigurasi untuk memungkinkan SSH akses masuk.

Akses CloudFormation template

Gunakan yang berikut ini URL untuk mengunduh CloudFormation templat, lalu unggah templat di AWS CloudFormation konsol untuk membuat CloudFormation tumpukan baru. Untuk informasi selengkapnya, lihat Menggunakan AWS CloudFormation konsol di Panduan AWS CloudFormation Pengguna.

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

Gunakan CloudFormation template untuk membuat template EC2 peluncuran

Gunakan prosedur berikut untuk menyelesaikan CloudFormation template di AWS CloudFormation konsol

- Di bawah Berikan nama tumpukan:
  - Di bawah nama Stack, masukkan`getstarted-lt`.
- Di bawah Parameter:

- Di bawah Keamanan
  - Untuk VpcSecurityGroupId, pilih grup keamanan yang disebutkan `default` di klaster AndaVPC.
  - Untuk ClusterSecurityGroupId, pilih grup bernama `cluster-getstarted-sg`
  - Untuk SshSecurityGroupId, pilih grup bernama `inbound-ssh-getstarted-sg`
  - Untuk SshKeyName, pilih SSH key pair pilihan Anda.
- Di bawah sistem File
  - Untuk EfsFilesystemId, masukkan ID sistem file dari sistem EFS file yang Anda buat sebelumnya dalam tutorial.
  - Untuk FSxLustreFilesystemId, masukkan ID sistem file dari sistem file FSx for Lustre yang Anda buat sebelumnya dalam tutorial.
  - Untuk FSxLustreFilesystemMountName, masukkan nama mount untuk yang sama FSx untuk sistem file Lustre.
- Pilih Berikutnya, lalu pilih Berikutnya lagi.
- Pilih Kirim.

Pantau status CloudFormation tumpukan. Ketika mencapai `CREATE_COMPLETE` template peluncuran siap untuk digunakan.

> **Note**
>
> Untuk melihat semua sumber daya yang dibuat CloudFormation template, buka AWS CloudFormation konsol. Pilih `getstarted-lt` tumpukan dan kemudian pilih tab Sumber Daya.

Buat grup node komputasi untuk node login di AWS PCS

Grup node komputasi adalah kumpulan virtual node komputasi (EC2instance) yang AWS PCS diluncurkan dan dikelola. Saat menentukan grup node komputasi, Anda menentukan ciri umum seperti tipe EC2 instance, jumlah instans minimum dan maksimum, VPC subnet target, opsi pembelian pilihan, dan konfigurasi peluncuran khusus. AWS PCSsecara otomatis meluncurkan, mengelola, dan mengakhiri node komputasi dalam grup node komputasi, sesuai dengan pengaturan ini.

Pada langkah ini, Anda akan meluncurkan grup node komputasi statis yang menyediakan akses interaktif ke cluster. Anda dapat menggunakan SSH atau Amazon EC2 Systems Manager (SSM) untuk masuk ke sana, lalu menjalankan perintah shell dan mengelola pekerjaan Slurm.

Untuk membuat grup node komputasi

- Buka AWS PCSkonsol dan arahkan ke Cluster.
- Pilih cluster bernama `get-started`
- Arahkan ke Compute node groups dan pilih Create.
- Di bagian pengaturan grup simpul komputasi, berikan yang berikut ini:
  - Hitung nama grup node — Enter`login`.
- Di bawah konfigurasi Komputasi, masukkan atau pilih nilai-nilai ini:
  - EC2template peluncuran - Pilih template peluncuran di mana namanya berada `login-getstarted-lt`
  - IAMcontoh profil - Pilih contoh profil bernama `AWSPCS-getstarted-role`
  - Subnet — Pilih subnet tempat nama dimulai. `hpc-networking:PublicSubnetA`
  - Contoh - Pilih`c6i.xlarge`.
  - Konfigurasi penskalaan - Untuk Min. jumlah instans, masukkan. `1` Untuk Max. jumlah contoh, masukkan`1`.
- Di bawah Pengaturan tambahan, tentukan yang berikut ini:
  - AMIID - Pilih AMI tempat nama dimulai `aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`
- Pilih Buat grup node komputasi.

Bidang Status menunjukkan Membuat saat grup node komputasi sedang disediakan. Anda dapat melanjutkan ke langkah berikutnya dalam tutorial saat sedang berlangsung.

Buat grup node komputasi untuk menjalankan pekerjaan komputasi di AWS PCS

Pada langkah ini, Anda akan meluncurkan grup node komputasi yang menskalakan secara elastis untuk menjalankan pekerjaan yang dikirimkan ke cluster.

Untuk membuat grup node komputasi

- Buka AWS PCSkonsol dan arahkan ke Cluster.
- Pilih cluster bernama `get-started`
- Arahkan ke Compute node groups dan pilih Create.
- Di bagian pengaturan grup simpul komputasi, berikan yang berikut ini:
  - Hitung nama grup node — Enter`compute-1`.
- Di bawah konfigurasi Komputasi, masukkan atau pilih nilai-nilai ini:
  - EC2template peluncuran - Pilih template peluncuran di mana namanya berada `compute-getstarted-lt`
  - IAMcontoh profil - Pilih contoh profil bernama `AWSPCS-getstarted-role`
  - Subnet — Pilih subnet tempat nama dimulai. `hpc-networking:PrivateSubnetA`
  - Contoh - Pilih`c6i.xlarge`.
  - Konfigurasi penskalaan - Untuk Min. jumlah instans, masukkan. `0` Untuk Max. jumlah contoh, masukkan`4`.
- Di bawah Pengaturan tambahan, tentukan yang berikut ini:
  - AMIID — Pilih AMI tempat nama dimulai`aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`.
- Pilih Buat grup node komputasi.

Bidang Status menunjukkan Membuat saat grup node komputasi sedang disediakan.

> **Important**
>
> Tunggu kolom Status untuk menunjukkan Aktif sebelum melanjutkan ke langkah berikutnya dalam tutorial ini.

Buat antrian untuk mengelola pekerjaan di AWS PCS

Anda mengirimkan pekerjaan ke antrian untuk menjalankannya. Pekerjaan tetap dalam antrian sampai AWS PCS menjadwalkannya untuk berjalan pada grup node komputasi. Setiap antrian dikaitkan dengan satu atau lebih grup node komputasi, yang menyediakan EC2 contoh yang diperlukan untuk melakukan pemrosesan.

Pada langkah ini, Anda akan membuat antrian yang menggunakan grup node komputasi untuk memproses pekerjaan.

Untuk membuat antrean

- Buka AWS PCSkonsol.
- Pilih cluster bernama`get-started`.
- Arahkan ke Compute node groups dan pastikan status `compute-1` grup adalah Active.

> **Important**
>
> Status `compute-1` grup harus Aktif sebelum Anda melanjutkan ke langkah berikutnya.

- Arahkan ke Antrian dan pilih Buat antrian.
  - Di bagian konfigurasi Antrian, berikan nilai berikut:
    - Nama antrian - Masukkan yang berikut ini: `demo`
    - Compute node groups - Pilih grup node komputasi bernama. `compute-1`
- Pilih Buat antrean.

Bidang Status menunjukkan Membuat saat antrian sedang dibuat.

> **Important**
>
> Tunggu kolom Status untuk menunjukkan Aktif sebelum melanjutkan ke langkah berikutnya dalam tutorial ini.

Connect ke AWS PCS klaster

Setelah status grup node `login` komputasi menjadi Aktif, Anda dapat terhubung ke EC2 instance yang dibuatnya.

Untuk terhubung ke node login

- Buka AWS PCSkonsol dan arahkan ke Cluster.
- Pilih cluster bernama`get-started`.

- Pilih Compute Node Groups.
- Arahkan ke grup node komputasi bernama`login`.
- Temukan ID grup node Compute.
- Di jendela atau tab browser lain, buka EC2konsol Amazon.
  - Pilih Instans.
  - Cari EC2 contoh dengan tag berikut. Ganti *node-group-id* dengan nilai ID grup node Compute dari langkah sebelumnya. Harus ada 1 contoh.

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

  - Connect ke EC2 instance. Anda dapat menggunakan Session Manager atauSSH.

    Session Manager

    - Pilih instans.
    - Pilih Hubungkan.
    - Di bawah Connect to instance, pilih Session Manager.
    - Pilih Hubungkan.
    - Pilih Hubungkan. Terminal interaktif diluncurkan di browser Anda.

    SSH

    - Pilih instans.
    - Pilih Hubungkan.
    - Di bawah Connect to instance, pilih SSHklien.
    - Ikuti instruksi yang diberikan oleh konsol.

    > **Note**
    >
    > Nama pengguna untuk instance **`ec2-user`**tidak`root`.

Jelajahi lingkungan cluster di AWS PCS

Setelah Anda masuk ke cluster, Anda dapat menjalankan perintah shell. Misalnya, Anda dapat mengubah pengguna, bekerja dengan data pada sistem file bersama, dan berinteraksi dengan Slurm.

Ubah pengguna

Jika Anda telah masuk ke cluster menggunakan Session Manager, Anda mungkin terhubung sebagai`ssm-user`. Ini adalah pengguna khusus yang dibuat untuk Session Manager. Beralih ke pengguna default di Amazon Linux 2 menggunakan perintah berikut. Anda tidak perlu melakukan ini jika Anda terhubung menggunakanSSH.

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

Bekerja dengan sistem file bersama

Anda dapat mengonfirmasi bahwa EFS sistem file dan FSx untuk sistem file Lustre tersedia dengan perintah. `df -h` Output pada cluster Anda harus menyerupai berikut ini:

```
[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
```

Sistem `/home` file dipasang 127.0.0.1 dan memiliki kapasitas yang sangat besar. Ini adalah sistem EFS file yang Anda buat sebelumnya dalam tutorial. Setiap file yang ditulis di sini akan tersedia di bawah `/home` pada semua node di cluster.

Sistem `/shared` file memasang IP pribadi dan memiliki kapasitas 1,2 TB. Ini adalah sistem file FSx untuk Lustre yang Anda buat sebelumnya dalam tutorial. Setiap file yang ditulis di sini akan tersedia di bawah `/shared` pada semua node di cluster.

Berinteraksi dengan Slurm

Topik

- Daftar antrian dan node

- Tampilkan lowongan kerja

Daftar antrian dan node

Anda dapat membuat daftar antrian dan node yang terkait dengannya. `sinfo` Output dari cluster Anda harus menyerupai berikut ini:

```
[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 ~]$
```

Perhatikan partisi bernama`demo`. Statusnya adalah up dan memiliki maksimal 4 node. Hal ini terkait dengan node dalam kelompok `compute-1` node. Jika Anda mengedit grup node komputasi dan meningkatkan jumlah maksimum instance menjadi 8, jumlah node akan dibaca 8 dan daftar node akan terbaca. `compute-1-[1-8]` Jika Anda membuat grup node komputasi kedua bernama `test` dengan 4 node, dan menambahkannya ke `demo` antrian, node tersebut akan muncul dalam daftar node juga.

Tampilkan lowongan kerja

Anda dapat membuat daftar semua pekerjaan, di negara bagian mana pun, pada sistem dengan`squeue`. Output dari cluster Anda harus menyerupai berikut ini:

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

Coba jalankan `squeue` lagi nanti, ketika Anda memiliki pekerjaan Slurm yang tertunda atau berjalan.

Jalankan pekerjaan node tunggal di AWS PCS

Untuk menjalankan pekerjaan menggunakan Slurm, Anda menyiapkan skrip pengiriman yang menentukan persyaratan pekerjaan dan mengirimkannya ke antrian dengan perintah. `sbatch` Biasanya, ini dilakukan dari direktori bersama sehingga node login dan komputasi memiliki ruang umum untuk mengakses file.

Connect ke node login cluster Anda dan jalankan perintah berikut pada prompt shell nya.

- Menjadi pengguna default. Ubah ke direktori bersama.

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

- Gunakan perintah berikut untuk membuat contoh skrip pekerjaan:

```
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
```

- Kirim skrip pekerjaan ke penjadwal Slurm:

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

- Ketika pekerjaan diserahkan, itu akan mengembalikan ID pekerjaan sebagai nomor. Gunakan ID itu untuk memeriksa status pekerjaan. Ganti *job-id* dalam perintah berikut dengan nomor dikembalikan dari`sbatch`.

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

  Example

```
squeue --job 1
```

  `squeue`Perintah mengembalikan output yang mirip dengan berikut ini:

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

- Lanjutkan untuk memeriksa status pekerjaan hingga mencapai status `R` (berjalan). Pekerjaan dilakukan ketika `squeue` tidak mengembalikan apa pun.
- Periksa isi `/shared` direktori.

```
ls -alth /shared
```

Output perintah mirip dengan yang berikut:

```
-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
```

File bernama `single.1.out` dan `single.1.err` ditulis oleh salah satu node komputasi cluster Anda. Karena pekerjaan dijalankan di direktori bersama (`/shared`), mereka juga tersedia di node login Anda. Inilah sebabnya mengapa Anda mengonfigurasi sistem file FSx for Lustre untuk cluster ini.

- Periksa isi `single.1.out` file.

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

Output Anda akan serupa dengan yang berikut ini.

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

Jalankan MPI pekerjaan multi-node dengan Slurm in AWS PCS

Instruksi ini menunjukkan menggunakan Slurm untuk menjalankan tugas message passing interface (MPI) di. AWS PCS

Jalankan perintah berikut pada prompt shell dari node login Anda.

- Menjadi pengguna default. Ubah ke direktori home nya.

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

- Buat kode sumber dalam bahasa pemrograman C.

```
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
```

- Muat MPI modul Buka.

```
module load openmpi
```

- Kompilasi program C.

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

- Tulis skrip pengiriman pekerjaan Slurm.

```
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
```

- Ubah ke direktori bersama.

```
cd /shared
```

- Kirimkan skrip pekerjaan.

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

- Gunakan `squeue` untuk memantau pekerjaan sampai selesai.
- Periksa isi`multi.out`:

```
cat multi.out
```

Output Anda serupa dengan yang berikut ini. Perhatikan bahwa setiap peringkat memiliki alamat IP sendiri karena berjalan pada node yang berbeda.

```
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
```

Hapus AWS sumber daya Anda untuk AWS PCS

Setelah Anda selesai dengan kelompok cluster dan node yang Anda buat untuk tutorial ini, Anda harus menghapus sumber daya yang Anda buat.

> **Important**
>
> Anda mendapatkan biaya penagihan untuk semua sumber daya yang berjalan di Akun AWS

Untuk menghapus AWS PCS sumber daya yang Anda buat untuk tutorial ini

- Buka AWS PCSkonsol.
- Arahkan ke cluster bernama get-started.
- Arahkan ke bagian Antrian.
- Pilih antrian bernama demo.
- Pilih Hapus.

> **Important**
>
> Tunggu hingga antrian telah dihapus sebelum melanjutkan.

- Arahkan ke bagian Compute node groups.
- Pilih grup node komputasi bernama compute-1.
- Pilih Hapus.

- Pilih grup node komputasi bernama login.
- Pilih Hapus.

> **Important**
>
> Tunggu hingga kedua grup node komputasi telah dihapus sebelum melanjutkan.

- Di halaman detail cluster untuk memulai, pilih Hapus.

> **Important**
>
> Tunggu sampai cluster telah dihapus sebelum melanjutkan dengan langkah-langkah selanjutnya.

Untuk menghapus AWS sumber daya lain yang Anda buat untuk tutorial ini

- Buka IAMkonsol.
  - Pilih Peran.
  - Pilih peran bernama AWSPCS-getstarted-role lalu pilih Delete.
  - Setelah peran dihapus, pilih Kebijakan.
  - Pilih kebijakan bernama AWSPCS-getstarted-policy lalu pilih Hapus.
- Buka konsol AWS CloudFormation.
  - Pilih tumpukan bernama getstarted-lt.
  - Pilih Hapus.

> **Important**
>
> Tunggu tumpukan dihapus sebelum melanjutkan.

- Buka EFSkonsol Amazon.
  - Pilih Sistem file.
  - Pilih sistem file bernama getstarted-efs.
  - Pilih Hapus.

> **Important**
>
> Tunggu hingga sistem file dihapus sebelum melanjutkan.

- Buka FSxkonsol Amazon.
  - Pilih Sistem file.
  - Pilih sistem file bernama getstarted-fsx.
  - Pilih Hapus.

> **Important**
>
> Tunggu hingga sistem file dihapus sebelum melanjutkan.

- Buka konsol AWS CloudFormation.
  - Pilih tumpukan bernama getstarted-sg.
  - Pilih Hapus.
- Buka konsol AWS CloudFormation.
  - Pilih tumpukan bernama hpc-networking.
  - Pilih Hapus.

Bekerja dengan AWS PCS

Bab ini memberikan informasi dan panduan untuk membantu Anda menggunakannya AWS PCS.

Topik

AWS PCSkluster

AWS PCSCluster terdiri dari komponen-komponen berikut:

- Instans terkelola dari perangkat lunak penjadwal HPC sistem, seperti daemon kontrol Slurm (). `slurmctld`
- Komponen yang terintegrasi dengan penjadwal HPC sistem untuk menyediakan dan mengelola EC2 instans Amazon.
- Komponen yang terintegrasi dengan penjadwal HPC sistem untuk mengirimkan log dan metrik ke Amazon. CloudWatch

Komponen ini berjalan di akun yang dikelola oleh AWS. Mereka bekerja sama untuk mengelola EC2 instans Amazon di akun pelanggan Anda. AWS PCSmenyediakan antarmuka jaringan elastis di VPC subnet Amazon Anda untuk menyediakan konektivitas dari perangkat lunak penjadwal ke EC2 instans Amazon (misalnya, untuk mendukung penjadwalan pekerjaan batch pada mereka dan memungkinkan pengguna menjalankan perintah penjadwal untuk membuat daftar dan mengelola pekerjaan tersebut).

Topik

Membuat cluster di Layanan Komputasi AWS Paralel

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda membuat klaster di AWS Parallel Computing Service (AWS PCS). Jika ini adalah pertama kalinya Anda membuat AWS PCS cluster, kami sarankan Anda mengikutiMemulai dengan AWS PCS. Tutorial ini dapat membantu Anda membuat HPC sistem kerja tanpa memperluas ke semua opsi yang tersedia dan arsitektur sistem yang mungkin.

Prasyarat

- Subnet yang ada VPC dan yang memenuhi AWS PCSJaringan persyaratan. Sebelum Anda menerapkan cluster untuk penggunaan produksi, kami sarankan Anda memiliki pemahaman menyeluruh tentang persyaratan VPC dan subnet. Untuk membuat VPC dan subnet, lihatMembuat VPC untuk AWS PCS cluster Anda.
- IAMPrinsipal dengan izin untuk membuat dan mengelola AWS PCS sumber daya. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Layanan Komputasi AWS Paralel.

Buat AWS PCS cluster

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk membuat cluster.

AWS Management Console

Untuk membuat klaster DB

1. Buka AWS PCS konsol di https://console.aws.amazon.com/pcs/rumah #/cluster dan pilih Buat cluster.
2. Di bagian Pengaturan cluster, masukkan bidang berikut:
   - Nama cluster — Nama untuk cluster Anda. Nama hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Itu harus dimulai dengan karakter alfabet dan

tidak boleh lebih dari 40 karakter. Nama harus unik di dalam Wilayah AWS dan Akun AWS tempat Anda membuat cluster.

- Scheduler — Pilih penjadwal dan versi. AWS PCSsaat ini mendukung Slurm 23.11. Untuk informasi selengkapnya, lihat Versi slurm di AWS PCS.
- Ukuran pengontrol - Pilih ukuran untuk pengontrol Anda. Ini menentukan berapa banyak pekerjaan bersamaan dan node komputasi yang dapat dikelola oleh cluster. AWS PCS Anda hanya dapat mengatur ukuran pengontrol saat cluster dibuat. Untuk informasi lebih lanjut tentang ukuran, lihatMemilih ukuran AWS PCS cluster.

3. Di bagian Jaringan, pilih nilai untuk bidang berikut:

   - VPC— Pilih yang sudah ada VPC yang memenuhi AWS PCS persyaratan. Untuk informasi selengkapnya, lihat AWS PCSVPCdan persyaratan dan pertimbangan subnet. Setelah Anda membuat cluster, Anda tidak dapat mengubahnyaVPC. Jika tidak VPCs terdaftar, Anda harus membuatnya terlebih dahulu.
   - Subnet - Semua subnet yang tersedia dalam yang dipilih VPC terdaftar. Pilih dua di Availability Zone yang berbeda. Setiap subnet harus memenuhi persyaratan AWS PCS subnet. Untuk informasi selengkapnya, lihat AWS PCSVPCdan persyaratan dan pertimbangan subnet. Kami menyarankan Anda memilih subnet pribadi untuk menghindari mengekspos endpoint scheduler Anda ke internet publik.
   - Grup keamanan — Tentukan grup keamanan yang AWS PCS ingin Anda kaitkan dengan antarmuka jaringan yang dibuatnya untuk klaster Anda. Anda harus memilih setidaknya satu grup keamanan yang memungkinkan komunikasi antara cluster Anda dan node komputasinya. Untuk informasi selengkapnya, lihat Persyaratan dan pertimbangan kelompok keamanan.

4. (Opsional) Di bawah Enkripsi, Anda dapat menentukan kunci khusus untuk mengenkripsi data pengontrol Anda dengan mengatur bidang ini:

   - KMSID kunci — Tinggalkan `aws/pcs` untuk menggunakan KMS kunci yang PCS membuat. Pilih alias KMS kunci yang ada untuk menggunakan KMS kunci kustom. Perhatikan bahwa akun yang digunakan untuk membuat cluster harus memiliki `kms:Decrypt` hak istimewa pada KMS kunci kustom.

5. (Opsional) Di bagian konfigurasi Slurm, Anda dapat menentukan opsi konfigurasi Slurm yang mengganti default yang ditetapkan oleh: AWS PCS

   - Turunkan waktu idle — Ini mengontrol berapa lama node komputasi yang disediakan secara dinamis tetap aktif setelah pekerjaan yang ditempatkan pada mereka selesai

atau dihentikan. Menyetel ini ke nilai yang lebih panjang dapat membuatnya lebih mungkin bahwa pekerjaan berikutnya dapat berjalan di node, tetapi dapat menyebabkan peningkatan biaya. Nilai yang lebih pendek akan mengurangi biaya, tetapi dapat meningkatkan proporsi waktu yang dihabiskan HPC sistem Anda untuk menyediakan node dibandingkan dengan menjalankan pekerjaan pada mereka.

- Prolog — Ini adalah jalur yang sepenuhnya memenuhi syarat ke direktori skrip prolog pada instance grup node komputasi Anda. Ini sesuai dengan pengaturan Prolog di Slurm. Perhatikan bahwa ini harus berupa direktori, bukan jalur ke executable tertentu.
- Epilog — Ini adalah jalur yang sepenuhnya memenuhi syarat ke direktori skrip epilog pada instance grup node komputasi Anda. Ini sesuai dengan pengaturan Epilog di Slurm. Perhatikan bahwa ini harus berupa direktori, bukan jalur ke executable tertentu.
- Pilih parameter tipe — Ini membantu mengontrol algoritma pemilihan sumber daya yang digunakan oleh Slurm. Menyetel nilai ini `CR_CPU_Memory` akan mengaktifkan penjadwalan sadar memori, sementara menyetelnya ke `CR_CPU` akan mengaktifkan CPU penjadwalan -only. Parameter ini sesuai dengan SelectTypeParameterspengaturan di Slurm di mana `SelectType` diatur ke `select/cons_tres` oleh. AWS PCS

6. (Opsional) Di bawah Tag, tambahkan tag apa pun ke AWS PCS cluster Anda.
7. Pilih Buat klaster. Bidang Status ditampilkan `Creating` saat AWS PCS membuat cluster. Proses ini dapat memakan waktu beberapa menit.

> **Important**
>
> Hanya ada 1 cluster dalam satu `Creating` keadaan per Wilayah AWS per Akun AWS. AWS PCSmengembalikan kesalahan jika sudah ada cluster dalam `Creating` keadaan ketika Anda mencoba membuat cluster.

AWS CLI

Untuk membuat klaster DB

1. Buat cluster Anda dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:

   - Ganti *`region`* dengan ID tempat Wilayah AWS Anda ingin membuat cluster Anda, sepertius-east-1.

- Ganti *my-cluster* dengan nama untuk cluster Anda. Nama hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Itu harus dimulai dengan karakter alfabet dan tidak boleh lebih dari 40 karakter. Nama harus unik di dalam Wilayah AWS dan Akun AWS di mana Anda membuat cluster.
- Ganti *23.11* dengan versi Slurm yang didukung.

  > **Note**
  >
  > AWS PCSsaat ini mendukung Slurm 23.11.

- Ganti *SMALL* dengan ukuran cluster yang didukung. Ini menentukan berapa banyak pekerjaan bersamaan dan node komputasi yang dapat dikelola oleh cluster. AWS PCS Itu hanya dapat diatur ketika cluster dibuat. Untuk informasi lebih lanjut tentang ukuran, lihatMemilih ukuran AWS PCS cluster.
- Ganti nilainya `subnetIds` dengan milik Anda sendiri. Kami menyarankan Anda memilih subnet pribadi untuk menghindari mengekspos endpoint scheduler Anda ke internet publik.
- Tentukan `securityGroupIds` yang AWS PCS ingin Anda kaitkan dengan antarmuka jaringan yang dibuatnya untuk cluster Anda. Kelompok keamanan harus VPC sama dengan cluster. Anda harus memilih setidaknya satu grup keamanan yang memungkinkan komunikasi antara cluster Anda dan node komputasinya. Untuk informasi selengkapnya, lihat Persyaratan dan pertimbangan kelompok keamanan.
- Secara opsional, Anda dapat menyempurnakan perilaku Slurm dengan menambahkan opsi. `--slurm-configration` Misalnya, Anda dapat mengatur waktu idle scale-down menjadi 60 menit (3600 detik) dengan. `--slurm configuration scaleDownIdeTime=3600`
- Secara opsional, Anda dapat memberikan KMS kunci khusus untuk mengenkripsi data pengontrol Anda menggunakan. `--kms-key-id` *kms-key* Ganti *kms-key* dengan ID kunci KMSARN, atau alias yang sudah ada. Perhatikan bahwa akun yang digunakan untuk membuat cluster harus memiliki `kms:Decrypt` hak istimewa pada KMS kunci kustom.

```
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. Diperlukan beberapa menit untuk menyediakan cluster. Anda dapat melakukan kueri status klaster Anda dengan perintah berikut. Jangan melanjutkan untuk membuat antrian atau menghitung grup node sampai bidang status klaster berada. `ACTIVE`

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

> **Important**
>
> Hanya ada 1 cluster dalam satu `Creating` keadaan per Wilayah AWS per Akun AWS. AWS PCSmengembalikan kesalahan jika sudah ada cluster dalam `Creating` keadaan ketika Anda mencoba membuat cluster.

Langkah selanjutnya yang disarankan untuk klaster Anda

- Tambahkan grup node komputasi.
- Tambahkan antrian.
- Aktifkan logging.

Menghapus cluster di AWS PCS

Topik ini memberikan gambaran umum tentang cara menghapus AWS PCS klaster.

Pertimbangan saat menghapus cluster AWS PCS

- Semua antrian yang terkait dengan cluster harus dihapus sebelum cluster dapat dihapus. Untuk informasi selengkapnya, lihat Menghapus antrian di AWS PCS.
- Semua grup node komputasi yang terkait dengan cluster harus dihapus sebelum cluster dapat dihapus. Untuk informasi selengkapnya, lihat Menghapus grup node komputasi di AWS PCS.

Hapus cluster

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menghapus cluster.

AWS Management Console

Untuk menghapus klaster

1. Buka AWS PCSkonsol.
2. Pilih cluster yang akan dihapus.
3. Pilih Hapus.
4. Bidang Status cluster menunjukkan`Deleting`. Ini bisa memakan waktu beberapa menit untuk menyelesaikannya.

AWS CLI

Untuk menghapus klaster

1. Gunakan perintah berikut untuk menghapus cluster, dengan penggantian ini:
   - Ganti *region-code* dengan cluster Wilayah AWS Anda ada di.
   - Ganti *my-cluster* dengan nama atau ID cluster Anda.

   ```
   aws pcs delete-cluster --region region-code --cluster-identifier my-cluster
   ```
2. Diperlukan beberapa menit untuk menghapus cluster. Anda dapat memeriksa status cluster Anda dengan perintah berikut.

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

Memilih ukuran AWS PCS cluster

AWS PCSmenyediakan cluster yang sangat tersedia dan aman, sambil mengotomatiskan tugas-tugas utama seperti patching, penyediaan node, dan pembaruan.

Saat Anda membuat cluster, Anda memilih ukuran untuk itu berdasarkan dua faktor:

- Jumlah node komputasi yang akan dikelola
- Jumlah pekerjaan aktif dan antrian yang Anda harapkan untuk dijalankan di cluster

Ukuran cluster slurm	Jumlah instans yang dikelola	Jumlah pekerjaan aktif dan antrian
Kecil	Hingga 32	Hingga 256
Sedang	Hingga 512	Hingga 8192
Besar	Hingga 2048	Hingga 16384

Contoh

- Jika klaster Anda memiliki hingga 24 instans terkelola dan menjalankan hingga 100 pekerjaan, pilih Small.
- Jika klaster Anda memiliki hingga 24 instans terkelola dan menjalankan hingga 1000 pekerjaan, pilih Medium.
- Jika klaster Anda memiliki hingga 1000 instans terkelola dan menjalankan hingga 100 pekerjaan, pilih Large.
- Jika klaster Anda memiliki hingga 1000 instans terkelola dan menjalankan hingga 10.000 pekerjaan, pilih Large.

Bekerja dengan rahasia cluster di AWS PCS

Sebagai bagian dari pembuatan cluster, AWS PCS buat rahasia cluster yang diperlukan untuk terhubung ke penjadwal pekerjaan di cluster. Anda juga membuat grup node AWS PCS komputasi, yang menentukan kumpulan instance yang akan diluncurkan sebagai respons terhadap peristiwa penskalaan. AWS PCSmengonfigurasi instance yang diluncurkan oleh grup node komputasi tersebut dengan rahasia cluster sehingga mereka dapat terhubung ke penjadwal pekerjaan. Ada kasus di mana Anda mungkin ingin mengkonfigurasi klien Slurm secara manual. Contohnya termasuk membangun node login persisten atau menyiapkan manajer alur kerja dengan kemampuan manajemen pekerjaan.

AWS PCSmenyimpan rahasia cluster sebagai rahasia terkelola dengan awalan `pcs!` di AWS Secrets Manager. Biaya rahasia sudah termasuk dalam biaya untuk menggunakan AWS PCS.

> **Warning**
>
> Jangan memodifikasi rahasia cluster Anda. AWS PCStidak akan dapat berkomunikasi dengan cluster Anda jika Anda memodifikasi rahasia cluster Anda. AWS PCStidak mendukung rotasi rahasia cluster. Anda harus membuat cluster baru jika Anda perlu memodifikasi rahasia cluster Anda.

Daftar Isi

Temukan rahasia cluster Slurm

Anda dapat menemukan rahasia yang AWS PCS dikelola menggunakan AWS Secrets Manager konsol atauAPI, langsung dari AWS PCS, atau menggunakan tag.

Gunakan AWS Secrets Manager untuk menemukan rahasia cluster

AWS Management Console

1. Arahkan ke konsol Secrets Manager.
2. Pilih Rahasia, lalu cari `pcs!` awalan.

> **Note**
>
> Sebuah rahasia AWS PCS cluster memiliki nama dalam bentuk di `pcs!slurm-secret-`*`cluster-id`* mana *`cluster-id`* adalah ID AWS PCS cluster.

AWS CLI

Setiap rahasia AWS PCS cluster juga ditandai dengan`aws:pcs:`*`cluster-id`*. Anda bisa mendapatkan ID rahasia untuk cluster dengan perintah yang mengikuti. Buat substitusi ini sebelum menjalankan perintah:

- Ganti *region* dengan Wilayah AWS untuk membuat cluster Anda, sepertius-east-1.
- Ganti *cluster-id* dengan ID AWS PCS cluster untuk menemukan rahasia cluster untuk.

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

Gunakan AWS PCS untuk menemukan rahasia cluster

Anda dapat menggunakan AWS CLI untuk menemukan rahasia AWS PCS cluster. ARN Masukkan perintah berikut, buat substitusi berikut:

- Ganti *region* dengan Wilayah AWS untuk membuat cluster Anda, sepertius-east-1.
- Ganti *my-cluster* dengan nama atau pengenal untuk cluster Anda.

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

Contoh output berikut adalah dari `get-cluster` perintah. Anda dapat menggunakan `secretArn` dan `secretVersion` bersama-sama untuk mendapatkan rahasia.

```
{
    "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"
    }
}
```

Dapatkan rahasia cluster Slurm

Anda dapat menggunakan Secrets Manager untuk mendapatkan versi rahasia cluster Slurm yang disandikan base64 saat ini. Contoh berikut menggunakan file. AWS CLI Buat substitusi berikut sebelum menjalankan perintah.

- Ganti *region* dengan Wilayah AWS untuk membuat cluster Anda, seperti`us-east-1`.
- Ganti *secret-arn* dengan `secretArn` dari AWS PCS cluster.

```
aws secretsmanager get-secret-value \
    --region region \
    --secret-id 'secret-arn' \
    --version-stage AWSCURRENT \
    --query 'SecretString' \
    --output text
```

Untuk informasi tentang cara menggunakan rahasia cluster Slurm, lihat. Menggunakan instance mandiri sebagai node login AWS PCS

Izin

Anda menggunakan IAM kepala sekolah untuk mendapatkan rahasia cluster Slurm. IAMKepala sekolah harus memiliki izin untuk membaca rahasianya. Untuk informasi selengkapnya, lihat Istilah dan konsep peran di Panduan AWS Identity and Access Management Pengguna.

IAMKebijakan sampel berikut memungkinkan akses ke contoh rahasia klaster.

```
{
    "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 PCSmenghitung grup simpul

Grup node AWS PCS komputasi adalah kumpulan node yang logis (EC2instance Amazon). Node ini dapat digunakan untuk menjalankan pekerjaan komputasi, serta untuk menyediakan akses interaktif berbasis shell ke suatu HPC sistem. Grup node komputasi terdiri dari aturan untuk membuat node, termasuk jenis EC2 instans Amazon mana yang akan digunakan, berapa banyak instance yang akan dijalankan, apakah akan menggunakan Instans Spot atau Instans Sesuai Permintaan, subnet dan grup keamanan mana yang akan digunakan, dan cara mengonfigurasi setiap instance saat diluncurkan. Saat aturan tersebut diperbarui, AWS PCS memperbarui sumber daya yang terkait dengan grup node komputasi agar sesuai.

Topik

- Membuat grup node komputasi di AWS PCS
- Memperbarui grup node AWS PCS komputasi
- Menghapus grup node komputasi di AWS PCS
- Menemukan instance grup node komputasi di AWS PCS

Membuat grup node komputasi di AWS PCS

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda membuat grup node komputasi di AWS Parallel Computing Service (AWS PCS). Jika ini adalah pertama kalinya Anda membuat grup node komputasi di AWS PCS, kami sarankan Anda mengikuti tutorial diMemulai dengan AWS PCS. Tutorial ini dapat membantu Anda membuat HPC sistem kerja tanpa memperluas ke semua opsi yang tersedia dan arsitektur sistem yang mungkin.

Prasyarat

- Kuota layanan yang memadai untuk meluncurkan jumlah EC2 instans yang diinginkan di Anda. Wilayah AWS Anda dapat menggunakan AWS Management Consoleuntuk memeriksa dan meminta kenaikan kuota layanan Anda.
- Subnet yang sudah ada VPC dan yang memenuhi persyaratan AWS PCS jaringan. Kami menyarankan agar Anda benar-benar memahami persyaratan ini sebelum Anda menerapkan klaster untuk penggunaan produksi. Untuk informasi selengkapnya, lihat AWS PCSVPCdan persyaratan dan pertimbangan subnet. Anda juga dapat menggunakan CloudFormation template untuk membuat VPC dan subnet. AWS menyediakan HPC resep untuk CloudFormation template. Untuk informasi lebih lanjut, lihat aws-hpc-recipesdi GitHub.
- Profil IAM instans dengan izin untuk memanggil AWS PCS `RegisterComputeNodeGroupInstance` API tindakan dan akses ke AWS sumber daya lain yang diperlukan untuk instance grup node Anda. Untuk informasi selengkapnya, lihat IAMprofil instance untuk Layanan Komputasi AWS Paralel.
- Template peluncuran untuk instance grup node Anda. Untuk informasi selengkapnya, lihat Menggunakan template EC2 peluncuran Amazon dengan AWS PCS.
- Untuk membuat grup node komputasi yang menggunakan instans Amazon EC2 Spot, Anda harus memiliki peran yang AWSServiceRoleForEC2Spotditautkan layanan di dalamnya. Akun AWS Untuk informasi selengkapnya, lihat Peran Amazon EC2 Spot untuk AWS PCS.

Buat grup node komputasi di AWS PCS

Anda dapat membuat grup node komputasi menggunakan AWS Management Console atau. AWS CLI

AWS Management Console

Untuk membuat grup node komputasi menggunakan konsol

1. Buka AWS PCSkonsol.
2. Pilih cluster tempat Anda ingin membuat grup node komputasi. Arahkan ke Compute node groups dan pilih Create.
3. Di bagian pengaturan grup node komputasi, berikan nama untuk grup node Anda. Nama hanya dapat berisi karakter alfanumerik peka huruf besar/kecil dan tanda hubung. Itu harus dimulai dengan karakter alfabet dan tidak boleh lebih dari 25 karakter. Nama harus unik di dalam cluster.
4. Di bawah konfigurasi Komputasi, masukkan atau pilih nilai-nilai ini:

    a. EC2template peluncuran - Pilih template peluncuran kustom untuk digunakan untuk grup node ini. Template peluncuran dapat digunakan untuk menyesuaikan pengaturan jaringan seperti subnet, dan grup keamanan, konfigurasi pemantauan, dan penyimpanan tingkat instance. Jika Anda belum menyiapkan template peluncuran, lihat Menggunakan template EC2 peluncuran Amazon dengan AWS PCS untuk mempelajari cara membuatnya.

    > **Important**
    >
    > AWS PCSmembuat template peluncuran terkelola untuk setiap grup node komputasi. Ini dinamai`pcs-`*`identifier`*`-do-not-delete`. Jangan pilih ini saat Anda membuat atau memperbarui grup node komputasi, atau grup node tidak akan berfungsi dengan benar.

    b. EC2luncurkan versi template - Pilih versi template peluncuran kustom Anda. Anda dapat memilih versi tertentu, yang dapat meningkatkan reproduktifitas. Jika Anda mengubah versi nanti, Anda harus memperbarui grup node komputasi untuk mendeteksi perubahan dalam template peluncuran. Untuk informasi selengkapnya, lihat Memperbarui grup node AWS PCS komputasi.

    c. AMIID — jika template peluncuran Anda tidak menyertakan AMI ID, atau jika Anda ingin mengganti nilai dalam template peluncuran, berikan AMI ID di sini. Perhatikan bahwa yang AMI digunakan untuk grup node harus kompatibel dengan AWS PCS. Anda juga dapat memilih sampel yang AMI disediakan oleh AWS. Untuk informasi lebih lanjut tentang topik ini, lihatGambar Mesin Amazon (AMIs) untuk AWS PCS.

d. IAMprofil instance - Pilih profil instance untuk grup simpul. Profil instans memberikan izin instans untuk mengakses AWS sumber daya dan layanan dengan aman. Jika Anda belum menyiapkannya, lihat IAMprofil instance untuk Layanan Komputasi AWS Paralel untuk mempelajari cara membuatnya.

e. Subnet — Pilih satu atau beberapa subnet di VPC tempat AWS PCS klaster Anda digunakan. Jika Anda memilih beberapa subnet, EFA komunikasi tidak akan tersedia di antara node, dan komunikasi antar node dalam subnet yang berbeda mungkin telah meningkatkan latensi. Pastikan subnet yang Anda tentukan di sini cocok dengan apa pun yang Anda tentukan dalam template EC2 peluncuran.

f. Instance — Pilih satu atau beberapa jenis instance untuk memenuhi permintaan penskalaan dalam grup node. Semua tipe instance harus memiliki arsitektur prosesor yang sama (x864_64 atau arm64) dan jumlah. vCPUs Jika instance memilikiGPUs, semua jenis instance harus memiliki jumlah yang sama. GPUs

g. Konfigurasi penskalaan - Tentukan jumlah instance minimum dan maksimum untuk grup node. Anda dapat menentukan konfigurasi statis, di mana ada sejumlah node tetap yang berjalan, atau konfigurasi dinamis, di mana hingga jumlah maksimum node dapat berjalan. Untuk konfigurasi statis, atur minimum dan maksimum ke angka yang sama, lebih besar dari angka nol. Untuk konfigurasi dinamis, atur instance minimum ke nol dan instance maksimum ke angka yang lebih besar dari nol. AWS PCStidak mendukung grup node komputasi dengan campuran instance statis dan dinamis.

5. (Opsional) Di bawah Pengaturan tambahan, tentukan yang berikut ini:

   a. Opsi pembelian — pilih antara instans Spot dan On-Demand.

   b. Strategi alokasi — jika Anda telah memilih opsi pembelian Spot, Anda dapat menentukan bagaimana kumpulan kapasitas Spot dipilih saat meluncurkan instance di grup node. Untuk informasi selengkapnya, lihat Strategi alokasi untuk Instans Spot di Panduan Pengguna Amazon Elastic Compute Cloud. Opsi ini tidak berpengaruh jika Anda telah memilih opsi Pembelian sesuai permintaan.

6. (Opsional) Di bagian pengaturan Slurm khusus, berikan nilai-nilai ini:

   a. Berat - Nilai ini menetapkan prioritas node dalam grup untuk tujuan penjadwalan. Node dengan bobot yang lebih rendah memiliki prioritas yang lebih tinggi, dan unitnya arbitrer. Untuk informasi selengkapnya, lihat Berat dalam Slurm dokumentasi.

   b. Memori nyata — Nilai ini menetapkan ukuran (dalam GB) memori nyata pada node dalam grup node. Ini dimaksudkan untuk digunakan bersama dengan `CR_CPU_Memory`

opsi dalam Slurm konfigurasi Cluster di AWS PCS. Untuk informasi lebih lanjut, lihat RealMemorydi Slurm dokumentasi.

7. (Opsional) Di bawah Tag, tambahkan tag apa pun ke grup node komputasi Anda.
8. Pilih Buat grup node komputasi. Bidang Status menunjukkan `Creating` sementara AWS PCS ketentuan grup node. Ini dapat memakan waktu beberapa menit.

Direkomendasikan langkah selanjutnya

- Tambahkan grup node Anda ke antrian AWS PCS untuk memungkinkannya memproses pekerjaan.

AWS CLI

Untuk membuat grup node komputasi Anda menggunakan AWS CLI

Buat antrian Anda dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:

1. Ganti *`region`*dengan ID Wilayah AWS untuk membuat cluster Anda, sepertius`-east-1`.
2. Ganti *`my-cluster`* dengan nama atau `clusterId` klaster Anda.
3. Ganti *`my-node-group`*dengan nama untuk grup node komputasi Anda. Nama hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Itu harus dimulai dengan karakter alfabet dan tidak boleh lebih dari 25 karakter. Nama harus unik di dalam cluster.
4. Ganti *`subnet-ExampleID1`* dengan satu atau lebih subnet IDs dari cluster VPC Anda.
5. Ganti *`lt-ExampleID1`* dengan ID untuk template peluncuran kustom Anda. Jika Anda belum menyiapkannya, lihat Menggunakan template EC2 peluncuran Amazon dengan AWS PCS untuk mempelajari cara membuatnya.

> **Important**
>
> AWS PCSmembuat template peluncuran terkelola untuk setiap grup node komputasi. Ini dinamaipcs-*`identifier`*`-do-not-delete`. Jangan pilih ini saat Anda membuat atau memperbarui grup node komputasi, atau grup node tidak akan berfungsi dengan benar.

6. Ganti *launch-template-version* dengan versi template peluncuran tertentu jika Anda ingin mengaitkan grup node Anda dengan versi tertentu.
7. Ganti *arn:InstanceProfile*dengan profil IAM instans Anda. ARN Jika Anda belum menyiapkannya, lihat Menggunakan template EC2 peluncuran Amazon dengan AWS PCS bimbingan.
8. Ganti *min-instances* and *max-instances* dengan nilai integer. Anda dapat menentukan konfigurasi statis, di mana ada sejumlah node tetap yang berjalan, atau konfigurasi dinamis, di mana hingga jumlah maksimum node dapat berjalan. Untuk konfigurasi statis, atur minimum dan maksimum ke angka yang sama, lebih besar dari angka nol. Untuk konfigurasi dinamis, atur instance minimum ke nol dan instance maksimum ke angka yang lebih besar dari nol. AWS PCStidak mendukung grup node komputasi dengan campuran instance statis dan dinamis.
9. Ganti *t3.large* dengan tipe instance lain. Anda dapat menambahkan lebih banyak jenis instance dengan menentukan daftar `instanceType` pengaturan. Misalnya, *--instance-configs instanceType=c6i.16xlarge,instanceType=c6a.16xlarge*. Semua tipe instance harus memiliki arsitektur prosesor yang sama (x864_64 atau arm64) dan jumlah. vCPUs Jika instance memilikiGPUs, semua jenis instance harus memiliki jumlah yang sama. 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
```

Ada beberapa pengaturan konfigurasi opsional yang dapat Anda tambahkan ke `create-compute-node-group` perintah.

- Anda dapat menentukan `--amiId` apakah templat peluncuran kustom Anda tidak menyertakan referensi keAMI, atau jika Anda ingin mengganti nilai tersebut. Perhatikan bahwa yang AMI digunakan untuk grup node harus kompatibel dengan AWS PCS. Anda juga dapat memilih sampel yang AMI disediakan oleh AWS. Untuk informasi lebih lanjut tentang topik ini, lihatGambar Mesin Amazon (AMIs) untuk AWS PCS.

- Anda dapat memilih antara instans on-demand (`ONDEMAND`) dan Spot (`SPOT`) menggunakan. `--purchase-option` On-demand adalah default. Jika Anda memilih instans Spot, Anda juga dapat menggunakan `--allocation-strategy` untuk menentukan cara AWS PCS memilih kumpulan kapasitas Spot saat meluncurkan instance di grup node. Untuk informasi selengkapnya, lihat Strategi alokasi untuk Instans Spot di Panduan Pengguna Amazon Elastic Compute Cloud.
- Dimungkinkan untuk memberikan opsi Slurm konfigurasi untuk node dalam kelompok node menggunakan`--slurm-configuration`. Anda dapat mengatur bobot (prioritas penjadwalan) dan memori nyata. Node dengan bobot yang lebih rendah memiliki prioritas yang lebih tinggi, dan unitnya arbitrer. Untuk informasi selengkapnya, lihat Berat dalam Slurm dokumentasi. Memori nyata menetapkan ukuran (dalam GB) memori nyata pada node dalam grup node. Ini dimaksudkan untuk digunakan bersama dengan `CR_CPU_Memory` opsi untuk cluster AWS PCS dalam Slurm konfigurasi Anda. Untuk informasi lebih lanjut, lihat RealMemorydi Slurm dokumentasi.

> **Important**
>
> Diperlukan beberapa menit untuk membuat grup node komputasi.

Anda dapat menanyakan status grup node Anda dengan perintah berikut. Anda tidak akan dapat mengaitkan grup node dengan antrian sampai statusnya tercapai`ACTIVE`.

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

Memperbarui grup node AWS PCS komputasi

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda memperbarui grup node AWS PCS komputasi.

Opsi untuk memperbarui grup node AWS PCS komputasi

Memperbarui grup node AWS PCS komputasi memungkinkan Anda mengubah properti instance yang diluncurkan oleh AWSPCS, serta aturan tentang cara instance tersebut diluncurkan. Misalnya,

Anda dapat mengganti instance grup AMI for node dengan yang lain dengan perangkat lunak berbeda yang diinstal di dalamnya. Atau, Anda dapat memperbarui grup keamanan untuk mengubah konektivitas jaringan masuk atau keluar. Anda juga dapat mengubah konfigurasi penskalaan atau bahkan mengubah opsi pembelian pilihan ke atau dari instans Spot.

Pengaturan grup node berikut tidak dapat diubah setelah pembuatan:

- Nama
- Instans

Pertimbangan saat memperbarui grup node AWS PCS komputasi

Grup node komputasi menentukan EC2 instance yang digunakan untuk memproses pekerjaan, menyediakan akses shell interaktif, dan tugas lainnya. Mereka sering dikaitkan dengan satu atau lebih AWS PCS antrian. Saat Anda memperbarui grup node komputasi untuk mengubah perilakunya (atau perilakunya), pertimbangkan hal berikut:

- Perubahan untuk menghitung properti grup node menjadi efektif ketika status grup node komputasi berubah dari Memperbarui ke Aktif. Instans baru diluncurkan dengan properti yang diperbarui.
- Pembaruan yang tidak memengaruhi konfigurasi node tertentu tidak memengaruhi node yang sedang berjalan. Misalnya, menambahkan subnet dan mengubah strategi alokasi.
- Jika Anda memperbarui template peluncuran untuk grup node komputasi, Anda harus memperbarui grup node komputasi untuk menggunakan versi baru.
- Untuk menambah atau menghapus grup keamanan dari node dalam grup node komputasi, edit template peluncurannya dan perbarui grup node komputasi. Instans baru diluncurkan dengan kumpulan grup keamanan yang diperbarui.
- Jika Anda langsung mengedit grup keamanan yang digunakan oleh grup node komputasi, itu akan segera berlaku pada instance running dan future.
- Jika Anda menambahkan atau menghapus izin dari profil IAM instance yang digunakan oleh grup node komputasi, ini akan segera berlaku pada instance running dan future.
- Untuk mengubah instance yang AMI digunakan oleh grup node komputasi, perbarui grup node komputasi (atau templat peluncurannya) untuk menggunakan yang baru AMI dan tunggu AWS PCS untuk mengganti instance.
- AWS PCSmenggantikan instance yang ada di grup node setelah operasi pembaruan grup node. Jika ada pekerjaan yang berjalan pada node, pekerjaan tersebut diizinkan untuk diselesaikan sebelum AWS PCS menggantikan node. Proses pengguna interaktif (seperti pada instance node

login) dihentikan. Status grup node kembali ke `Active` saat AWS PCS menandai instance untuk penggantian, tetapi penggantian sebenarnya terjadi ketika instance menganggur.

- Jika Anda mengurangi jumlah maksimum instance yang diizinkan dalam grup node komputasi, AWS PCS hapus node dari Slurm untuk memenuhi maksimum baru. AWS PCSmengakhiri instance yang sedang berjalan terkait dengan node Slurm yang dihapus. Pekerjaan yang berjalan pada node yang dihapus gagal dan kembali ke antrian mereka.
- AWS PCSmembuat template peluncuran terkelola untuk setiap grup node komputasi. Mereka diberi nama`pcs-`*`identifier`*`-do-not-delete`. Jangan memilihnya saat Anda membuat atau memperbarui grup node komputasi, atau grup node tidak akan berfungsi dengan benar.
- Jika Anda memperbarui grup node komputasi untuk menggunakan Spot untuk opsi pembeliannya, Anda harus memiliki peran AWSServiceRoleForEC2Spotterkait layanan di akun Anda. Untuk informasi selengkapnya, lihat Peran Amazon EC2 Spot untuk AWS PCS.

Untuk memperbarui grup node AWS PCS komputasi

Anda dapat memperbarui grup node menggunakan AWS Management Console atau AWSCLI.

AWS Management Console

Untuk memperbarui grup node komputasi

1. Buka AWS PCS konsol di `https://console.aws.amazon.com/pcs/home#/clusters`
2. Pilih cluster tempat Anda ingin memperbarui grup node komputasi.
3. Arahkan ke Compute node groups, buka grup node yang ingin Anda perbarui, lalu pilih Edit.
4. Di bagian konfigurasi Komputasi, Pengaturan tambahan, dan pengaturan Slurmpenyesuaian, perbarui nilai apa pun kecuali:
   - Instance - Anda tidak dapat mengubah instance dalam grup node komputasi.
5. Pilih Perbarui. Bidang Status akan menampilkan Memperbarui saat perubahan sedang diterapkan.

> **Important**
>
> Menghitung pembaruan grup node dapat memakan waktu beberapa menit.

AWS CLI

Untuk memperbarui grup node komputasi

1. Perbarui grup node komputasi Anda dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:

   a. Ganti *region-code* dengan AWS Wilayah tempat Anda ingin membuat cluster Anda.

   b. Ganti *my-node-group* dengan nama atau `computeNodeGroupId` untuk grup node komputasi Anda.

   c. Ganti *my-cluster* dengan nama atau `clusterId` klaster Anda.

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

2. Perbarui parameter grup node apa pun kecuali untuk`--instance-configs`. Misalnya, untuk menetapkan AMI ID baru, lewati `--amiId my-custom-ami-id` di mana *my-custom-ami-id* digantikan oleh AMI pilihan Anda.

> **Important**
>
> Diperlukan beberapa menit untuk memperbarui grup node komputasi.

Anda dapat menanyakan status grup node Anda dengan perintah berikut.

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

Menghapus grup node komputasi di AWS PCS

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda menghapus grup node komputasi. AWS PCS

Pertimbangan saat menghapus grup node komputasi

Grup node komputasi menentukan EC2 instance yang digunakan untuk memproses pekerjaan, menyediakan akses shell interaktif, dan tugas lainnya. Mereka sering dikaitkan dengan satu atau lebih AWS PCS antrian. Sebelum Anda menghapus grup node komputasi, pertimbangkan hal berikut:

- Setiap EC2 instance yang diluncurkan oleh grup node komputasi akan dihentikan. Ini akan membatalkan pekerjaan yang berjalan pada instance ini, dan menghentikan proses interaktif yang sedang berjalan.
- Anda harus memisahkan grup node komputasi dari semua antrian sebelum Anda dapat menghapusnya. Untuk informasi selengkapnya, lihat Memperbarui AWS PCS antrian.

Hapus grup node komputasi

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menghapus grup node komputasi.

AWS Management Console

Untuk menghapus grup node komputasi

1. Buka AWS PCSkonsol.
2. Pilih cluster dari grup node komputasi.
3. Arahkan ke Compute node groups dan pilih compute node group yang akan dihapus.
4. Pilih Hapus.
5. Bidang Status menunjukkan`Deleting`. Ini bisa memakan waktu beberapa menit untuk menyelesaikannya.

> **Note**
>
> Anda dapat menggunakan perintah asli penjadwal Anda untuk mengonfirmasi bahwa grup node komputasi dihapus. Misalnya, gunakan `sinfo` atau `squeue` untuk slurm.

AWS CLI

Untuk menghapus grup node komputasi

- Gunakan perintah berikut untuk menghapus grup node komputasi, dengan penggantian ini:
  - Ganti *region-code* dengan cluster Wilayah AWS Anda ada di.
  - Ganti *my-node-group* dengan nama atau ID grup node komputasi Anda.
  - Ganti *my-cluster* dengan nama atau ID cluster Anda.

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

Diperlukan beberapa menit untuk menghapus grup node komputasi.

> **Note**
>
> Anda dapat menggunakan perintah asli penjadwal Anda untuk mengonfirmasi bahwa grup node komputasi dihapus. Misalnya, gunakan `sinfo` atau `squeue` untuk slurm.

Menemukan instance grup node komputasi di AWS PCS

Setiap grup node AWS PCS komputasi dapat meluncurkan EC2 instance dengan konfigurasi bersama. Anda dapat menggunakan EC2 tag untuk menemukan instance dalam grup node komputasi di AWS Management Console atau dengan. AWS CLI

AWS Management Console

Untuk menemukan instance grup node komputasi Anda

1. Buka AWS PCSkonsol.
2. Pilih cluster.
3. Pilih Compute Node Groups.
4. Temukan ID untuk grup node login yang Anda buat.
5. Arahkan ke EC2konsol dan pilih Instans.

6. Cari instance dengan tag berikut. Ganti *node-group-id* dengan ID (bukan nama) grup node komputasi Anda.

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

7. (Opsional) Anda dapat mengubah nilai status Instance di kolom pencarian untuk menemukan instance yang sedang dikonfigurasi atau yang baru saja dihentikan.
8. Temukan ID instans dan alamat IP untuk setiap instance dalam daftar instance yang ditandai.

AWS CLI

Untuk menemukan instance grup node Anda, gunakan perintah yang mengikuti. Sebelum menjalankan perintah, buat penggantian berikut:

- Ganti *region-code* dengan Wilayah AWS cluster Anda. Contoh: `us-east-1`
- Ganti *node-group-id* dengan ID (bukan nama) grup node komputasi Anda.
- Ganti `running` dengan status instance lain seperti `pending` atau `terminated` untuk menemukan EC2 instance di negara bagian lain.

```
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}
```

Perintah ini menghasilkan output serupa dengan berikut: Nilai dari `PublicIP` adalah `null` jika instance berada dalam subnet pribadi.

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

```
]
```

> **Note**
>
> Jika Anda berharap `describe-instances` untuk mengembalikan sejumlah besar instance, Anda harus menggunakan opsi untuk beberapa halaman. Untuk informasi selengkapnya, lihat DescribeInstancesdi Amazon Elastic Compute Cloud API Reference.

Menggunakan template EC2 peluncuran Amazon dengan AWS PCS

Di AmazonEC2, template peluncuran dapat menyimpan serangkaian preferensi sehingga Anda tidak perlu menentukannya satu per satu saat meluncurkan instance. AWS PCSmenggabungkan template peluncuran sebagai cara yang fleksibel untuk mengkonfigurasi grup node komputasi. Saat Anda membuat grup node, Anda menyediakan template peluncuran. AWS PCSmembuat template peluncuran turunan darinya yang mencakup transformasi untuk membantu memastikannya berfungsi dengan layanan.

Memahami apa opsi dan pertimbangannya saat menulis templat peluncuran khusus dapat membantu Anda menuliskannya untuk digunakan. AWS PCS Untuk informasi selengkapnya tentang template peluncuran, lihat Meluncurkan Instance dari Meluncurkan instance dari template peluncuran di Panduan EC2 Pengguna Amazon.

Topik

- Gambaran Umum
- Buat template peluncuran dasar
- Bekerja dengan data EC2 pengguna Amazon
- Reservasi Kapasitas di AWS PCS
- Parameter template peluncuran yang berguna

Gambaran Umum

Ada lebih dari 30 parameter yang tersedia yang dapat Anda sertakan dalam template EC2 peluncuran, mengendalikan banyak aspek bagaimana instance dikonfigurasi. Sebagian besar sepenuhnya kompatibel dengan AWS PCS, tetapi ada beberapa pengecualian.

Parameter template EC2 Launch berikut akan diabaikan oleh AWS PCS karena properti ini harus langsung dikelola oleh layanan:

- Jenis instance/tentukan atribut tipe instance (`InstanceRequirements`) - AWS PCS tidak mendukung pemilihan instance berbasis atribut.
- Jenis instans (`InstanceType`) - Tentukan jenis instance saat Anda membuat grup simpul.
- Detail lanjutan/profil IAM instance (`IamInstanceProfile`) - Anda memberikan ini saat Anda membuat atau memperbarui grup node.
- Detail lanjutan/nonaktifkan API terminasi (`DisableApiTermination`) — AWS PCS harus mengontrol siklus hidup instance grup node yang diluncurkan.
- Detail lanjutan/Nonaktifkan API stop (`DisableApiStop`) - AWS PCS harus mengontrol siklus hidup instance grup node yang diluncurkan.
- Detail lanjutan/Stop — Perilaku hibernate (`HibernationOptions`) — AWS PCS tidak mendukung hibernasi instance.
- Detail Lanjutan/Elastic GPU () `ElasticGpuSpecifications` — Amazon Elastic Graphics mencapai akhir masa pakai pada 8 Januari 2024.
- Detail lanjutan/Inferensi elastis (`ElasticInferenceAccelerators`) - Amazon Elastic Inference tidak lagi tersedia untuk pelanggan baru.
- AAdvancedDetail/tentukan CPU opsi/utas per inti (`ThreadsPerCore`) - AWS PCS mengatur jumlah utas per inti menjadi 1.

Parameter ini memiliki persyaratan khusus yang mendukung kompatibilitas dengan AWS PCS:

- Data pengguna (`UserData`) — Ini harus dikodekan multi-bagian. Lihat Bekerja dengan data EC2 pengguna Amazon.
- Aplikasi dan Gambar OS (`ImageId`) - Anda dapat memasukkan ini. Namun, jika Anda menentukan AMI ID saat Anda membuat atau memperbarui grup node, itu akan mengganti nilai dalam template peluncuran. Yang AMI Anda berikan harus kompatibel dengan AWS PCS. Untuk informasi lebih lanjut, lihat "Gambar Mesin Amazon (AMIs) untuk AWS PCS.
- Pengaturan Jaringan/Firewall (grup keamanan**`SecurityGroups`**) () - Daftar nama grup keamanan tidak dapat diatur dalam templat peluncuran. AWS PCS Anda dapat mengatur daftar grup keamanan IDs (`SecurityGroupIds`), kecuali Anda menentukan antarmuka jaringan dalam template peluncuran. Kemudian, Anda harus menentukan grup keamanan IDs untuk setiap antarmuka. Untuk informasi selengkapnya, lihat Kelompok keamanan di AWS PCS.

- Pengaturan jaringan/Konfigurasi jaringan lanjutan (`NetworkInterfaces`) - Jika Anda menggunakan EC2 instance dengan kartu jaringan tunggal, dan tidak memerlukan konfigurasi jaringan khusus, AWS PCS dapat mengonfigurasi jaringan instance untuk Anda. Untuk mengonfigurasi beberapa kartu jaringan atau mengaktifkan Elastic Fabric Adapter pada instans Anda, gunakan`NetworkInterfaces`. Setiap antarmuka jaringan harus memiliki daftar grup keamanan IDs di bawah`Groups`. Untuk informasi selengkapnya, lihat Beberapa antarmuka jaringan di AWS PCS.
- Detail lanjutan/reservasi kapasitas (`CapacityReservationSpecification`) — Ini dapat diatur, tetapi tidak dapat merujuk spesifik `CapacityReservationId` saat bekerja dengan. AWS PCS Namun, Anda dapat mereferensikan grup reservasi kapasitas, di mana grup tersebut berisi satu atau lebih reservasi kapasitas. Untuk informasi selengkapnya, lihat Reservasi Kapasitas di AWS PCS.

Buat template peluncuran dasar

Anda dapat membuat template peluncuran menggunakan AWS Management Console atau AWS CLI.

AWS Management Console

Untuk membuat templat peluncuran

1. Buka EC2konsol Amazon dan pilih Luncurkan templat.
2. Pilih Buat templat peluncuran.
3. Di bawah Nama dan deskripsi template Luncurkan, masukkan nama unik dan khas untuk nama template Peluncuran
4. Di bawah Key pair (login) pada nama Key pair, pilih SSH key pair yang akan digunakan untuk login ke EC2 instance yang dikelola oleh AWS PCS. Ini memang opsional, tetapi direkomendasikan.
5. Di bawah Pengaturan jaringan, lalu Firewall (grup keamanan), pilih grup keamanan untuk dilampirkan ke antarmuka jaringan. Semua grup keamanan dalam template peluncuran harus dari AWS PCS cluster AndaVPC. Minimal, pilih:
   - Grup keamanan yang memungkinkan komunikasi dengan AWS PCS cluster
   - Grup keamanan yang memungkinkan komunikasi antar EC2 instans yang diluncurkan oleh AWS PCS
   - (Opsional) Grup keamanan yang memungkinkan SSH akses masuk ke instans interaktif

- (Opsional) Grup keamanan yang memungkinkan node komputasi untuk membuat koneksi keluar ke Internet
- (Opsional) Grup keamanan yang memungkinkan akses ke sumber daya jaringan seperti sistem file bersama atau server database.

6. ID template peluncuran baru Anda akan dapat diakses di EC2 konsol Amazon di bawah Peluncuran template. ID template peluncuran akan memiliki formulir`lt-0123456789abcdef01`.

Direkomendasikan langkah selanjutnya

- Gunakan template peluncuran baru untuk membuat atau memperbarui grup node AWS PCS komputasi.

AWS CLI

Untuk membuat templat peluncuran

Buat template peluncuran Anda dengan perintah berikut.

- Sebelum menjalankan perintah, buat penggantian berikut:
  a. Ganti *`region-code`* dengan Wilayah AWS tempat Anda bekerja dengan AWS PCS
  b. Ganti *`my-launch-template-name`* dengan nama untuk template Anda. Itu harus unik untuk Akun AWS dan Wilayah AWS Anda gunakan.
  c. Ganti *`my-ssh-key-name`* dengan nama SSH kunci pilihan Anda.
  d. Ganti *`sg-ExampleID1`* and *`sg-ExampleID2`* dengan grup keamanan IDs yang memungkinkan komunikasi antara EC2 instans Anda dan penjadwal dan komunikasi antar EC2 instance. Jika Anda hanya memiliki satu grup keamanan yang memungkinkan semua lalu lintas ini, Anda dapat menghapus `sg-ExampleID2` dan karakter koma sebelumnya. Anda juga dapat menambahkan lebih banyak grup keamananIDs. Semua grup keamanan yang Anda sertakan dalam template peluncuran harus dari AWS PCS cluster AndaVPC.

```
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"]}'
```

Teks keluaran AWS CLI akan menyerupai berikut ini. ID template peluncuran ditemukan di`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"
    }
}
```

Direkomendasikan langkah selanjutnya

- Gunakan template peluncuran baru untuk membuat atau memperbarui grup node AWS PCS komputasi.

Bekerja dengan data EC2 pengguna Amazon

Anda dapat menyediakan data EC2 pengguna di template peluncuran yang `cloud-init` berjalan saat instans diluncurkan. Blok data pengguna dengan tipe konten `cloud-config` dijalankan sebelum instance mendaftar dengan AWS PCSAPI, sementara blok data pengguna dengan tipe konten `text/x-shellscript` dijalankan setelah pendaftaran selesai, tetapi sebelum daemon Slurm dimulai. Untuk informasi selengkapnya, lihat dokumentasi cloud-init.

data pengguna kami dapat melakukan skenario konfigurasi umum, termasuk namun tidak terbatas pada hal-hal berikut:

- Termasuk pengguna atau grup
- Menginstal paket
- Membuat partisi dan sistem file
- Memasang sistem file jaringan

Data pengguna dalam templat peluncuran harus dalam format arsip MIME multi-bagian. Ini karena data pengguna Anda digabungkan dengan data AWS PCS pengguna lain yang diperlukan untuk mengkonfigurasi node di grup node Anda. Anda dapat menggabungkan beberapa blok data pengguna menjadi satu file MIME multi-bagian.

File MIME multi-bagian terdiri dari komponen-komponen berikut:

- Jenis konten dan deklarasi batas bagian: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- Deklarasi MIME versi: `MIME-Version: 1.0`
- Satu atau beberapa blok data pengguna yang berisi komponen berikut:
  - Batas pembuka yang menandakan awal dari blok data pengguna:. `--==BOUNDARY==` Anda harus menjaga garis sebelum batas ini kosong.
  - Deklarasi tipe konten untuk blok: `Content-Type: text/cloud-config; charset="us-ascii"` atau`Content-Type: text/x-shellscript; charset="us-ascii"`. Anda harus menjaga baris setelah deklarasi tipe konten kosong.
  - Isi data pengguna, seperti daftar perintah atau `cloud-config` arahan shell.
- Batas penutupan yang menandakan akhir file MIME multi-bagian:. `--==BOUNDARY==--` Anda harus menjaga garis sebelum batas penutupan kosong.

> **Note**
>
> Jika Anda menambahkan data pengguna ke template peluncuran di EC2 konsol Amazon, Anda dapat menempelkannya sebagai teks biasa. Atau, Anda dapat mengunggahnya dari file. Jika Anda menggunakan AWS CLI atau AWS SDK, Anda harus terlebih dahulu mengkodekan data pengguna base64 dan mengirimkan string itu sebagai nilai `UserData` parameter saat Anda memanggil CreateLaunchTemplate, seperti yang ditunjukkan dalam file iniJSON.

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

```
}
```

Contoh

- Contoh: Instal perangkat lunak dari repositori paket
- Contoh: Jalankan skrip dari bucket S3
- Contoh: Mengatur variabel lingkungan global
- Menggunakan sistem berkas jaringan dengan AWS PCS
- Contoh: Gunakan sistem EFS file sebagai direktori home bersama

Contoh: Instal perangkat lunak untuk AWS PCS dari repositori paket

Berikan skrip ini sebagai nilai `"userData"` dalam template peluncuran Anda. Untuk informasi selengkapnya, lihat Bekerja dengan data EC2 pengguna Amazon.

Skrip ini menggunakan cloud-config untuk menginstal paket perangkat lunak pada instance grup node saat peluncuran. Untuk informasi selengkapnya, lihat Format data pengguna dalam dokumentasi cloud-init. Contoh ini menginstal `curl` dan`llvm`.

> **Note**
>
> Instance Anda harus dapat terhubung ke repositori paket yang dikonfigurasi.

```
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==--
```

Contoh: Jalankan skrip tambahan untuk AWS PCS dari bucket S3

Berikan skrip ini sebagai nilai `"userData"` dalam template peluncuran Anda. Untuk informasi selengkapnya, lihat Bekerja dengan data EC2 pengguna Amazon.

Skrip ini menggunakan cloud-config untuk mengimpor skrip dari bucket S3 dan menjalankannya pada instance grup node saat diluncurkan. Untuk informasi selengkapnya, lihat Format data pengguna dalam dokumentasi cloud-init.

Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *my-bucket-name* — Nama bucket S3 yang dapat dibaca akun Anda.
- *path* — Jalur relatif terhadap root bucket S3.
- *shell* — Shell Linux yang digunakan untuk menjalankan skrip, sepertibash.

```
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==--
```

Profil IAM instance untuk grup node harus memiliki akses ke bucket. IAMKebijakan berikut adalah contoh untuk bucket dalam skrip data pengguna di atas.

```
{
    "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/*"
            ]
        }
    ]
}
```

Contoh: Tetapkan variabel lingkungan global untuk AWS PCS

Berikan skrip ini sebagai nilai `"userData"` dalam template peluncuran Anda. Untuk informasi selengkapnya, lihat Bekerja dengan data EC2 pengguna Amazon.

Contoh berikut digunakan `/etc/profile.d` untuk mengatur variabel global pada instance grup node.

```
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==--
```

Contoh: Gunakan sistem EFS file sebagai direktori home bersama untuk AWS PCS

Berikan skrip ini sebagai nilai `"userData"` dalam template peluncuran Anda. Untuk informasi selengkapnya, lihat Bekerja dengan data EC2 pengguna Amazon.

Contoh ini memperluas contoh EFS mount in Menggunakan sistem berkas jaringan dengan AWS PCS untuk mengimplementasikan direktori home bersama. Isi /home dicadangkan sebelum sistem EFS file dipasang. Konten kemudian dengan cepat disalin ke tempatnya pada penyimpanan bersama setelah pemasangan selesai.

Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *`/mount-point-directory`* — Jalur pada contoh di mana Anda ingin me-mount sistem EFS file.
- *`filesystem-id`* — ID sistem file untuk sistem EFS file.

```
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==--
```

Mengaktifkan tanpa kata sandi SSH

Anda dapat membangun contoh direktori home bersama untuk mengimplementasikan SSH koneksi antar instance cluster menggunakan SSH kunci. Untuk setiap pengguna yang menggunakan sistem file home bersama, jalankan skrip yang menyerupai berikut ini:

```
#!/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**
>
> Instance harus menggunakan grup keamanan yang memungkinkan SSH koneksi antara node cluster.

Reservasi Kapasitas di AWS PCS

Anda dapat memesan EC2 kapasitas Amazon di Availability Zone tertentu dan untuk durasi tertentu menggunakan Reservasi Kapasitas Sesuai Permintaan atau Blok EC2 Kapasitas untuk memastikan bahwa Anda memiliki kapasitas komputasi yang diperlukan yang tersedia saat Anda membutuhkannya.

> **Note**
>
> AWS PCSmendukung Reservasi Kapasitas Sesuai Permintaan (ODCR) tetapi saat ini tidak mendukung Blok Kapasitas untuk ML.

Menggunakan ODCRs dengan AWS PCS

Anda dapat memilih cara AWS PCS mengkonsumsi instans cadangan Anda. Jika Anda membuat openODCR, setiap instans yang cocok yang diluncurkan oleh AWS PCS atau proses lain di akun Anda dihitung terhadap reservasi. Dengan targetODCR, hanya instans yang diluncurkan dengan ID reservasi tertentu yang dihitung terhadap reservasi. Untuk beban kerja yang sensitif terhadap waktu, ditargetkan ODCRs lebih umum.

Anda dapat mengonfigurasi grup node AWS PCS komputasi untuk menggunakan target ODCR dengan menambahkannya ke template peluncuran. Berikut adalah langkah-langkah untuk melakukannya:

1. Buat Reservasi Kapasitas sesuai permintaan yang ditargetkan (ODCR).
2. Tambahkan ODCR ke grup Reservasi Kapasitas.
3. Kaitkan grup Reservasi Kapasitas dengan templat peluncuran.
4. Buat atau perbarui grup node AWS PCS komputasi untuk menggunakan template peluncuran.

Contoh: Cadangan dan gunakan instance hpc6a.48xlarge dengan target ODCR

Perintah contoh ini membuat target ODCR untuk 32 instance hpc6a.48xlarge. Untuk meluncurkan instance cadangan dalam grup penempatan, tambahkan `--placement-group-arn` ke perintah. Anda dapat menentukan tanggal berhenti dengan `--end-date` dan`--end-date-type`, jika tidak, reservasi akan berlanjut hingga dihentikan secara manual.

```
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
```

Hasil dari perintah ini akan menjadi ARN untuk yang baruODCR. Untuk menggunakan ODCR with AWS PCS, itu harus ditambahkan ke grup Reservasi Kapasitas. Ini karena AWS PCS tidak mendukung individuODCRs. Untuk informasi selengkapnya, lihat grup Reservasi Kapasitas di Panduan Pengguna Amazon Elastic Compute Cloud.

Berikut adalah cara menambahkan ke grup Reservasi Kapasitas bernama`EXAMPLE-CR-GROUP`. ODCR

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

Dengan ODCR dibuat dan ditambahkan ke grup Reservasi Kapasitas, sekarang dapat dihubungkan ke grup node AWS PCS komputasi dengan menambahkannya ke template peluncuran. Berikut adalah contoh template peluncuran yang mereferensikan grup Reservasi Kapasitas.

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

Terakhir, buat atau perbarui grup node AWS PCS komputasi untuk menggunakan instance hpc6a.48xlarge dan gunakan templat peluncuran yang mereferensikan grup Reservasi Kapasitasnya. ODCR Untuk grup node statis, atur instance minimum dan maksimum ke ukuran reservasi (32). Untuk grup node dinamis, atur instance minimum ke 0 dan maksimum hingga ukuran reservasi.

Contoh ini adalah implementasi sederhana dari satu ODCR yang disediakan untuk satu grup node komputasi. Tapi, AWS PCS mendukung banyak desain lainnya. Misalnya, Anda dapat membagi grup besar ODCR atau Reservasi Kapasitas di antara beberapa grup node komputasi. Atau, Anda dapat menggunakan AWS akun lain ODCRs yang telah dibuat dan dibagikan dengan akun Anda. Kendala utama adalah bahwa ODCRs selalu harus terkandung dalam grup Reservasi Kapasitas.

Untuk informasi selengkapnya, lihat Reservasi Kapasitas Sesuai Permintaan dan Blok Kapasitas untuk ML di Panduan Pengguna Amazon Elastic Compute Cloud.

Parameter template peluncuran yang berguna

Bagian ini menjelaskan beberapa parameter template peluncuran yang mungkin berguna secara luas. AWS PCS

Aktifkan CloudWatch pemantauan terperinci

Anda dapat mengaktifkan kumpulan CloudWatch metrik pada interval yang lebih pendek menggunakan parameter template peluncuran.

AWS Management Console

Pada halaman konsol untuk membuat atau mengedit templat peluncuran, opsi ini ditemukan di bawah bagian Detail lanjutan. Atur CloudWatch Pemantauan terperinci ke Aktifkan.

YAML

```
Monitoring:
    Enabled: True
```

JSON

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

Untuk informasi selengkapnya, lihat Mengaktifkan atau menonaktifkan pemantauan terperinci untuk instans Anda di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Layanan Metadata Instance Versi 2 (v2) IMDS

Menggunakan IMDS v2 dengan EC2 instans menawarkan peningkatan keamanan yang signifikan dan membantu mengurangi potensi risiko yang terkait dengan mengakses metadata instans di lingkungan. AWS

AWS Management Console

Pada halaman konsol untuk membuat atau mengedit templat peluncuran, opsi ini ditemukan di bawah bagian Detail lanjutan. Setel Metadata yang dapat diakses ke Diaktifkan, versi Metadata ke V2 saja (diperlukan token), dan batas hop respons Metadata menjadi 4.

YAML

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

JSON

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

.

AWS PCSantrian

AWS PCSAntrian adalah abstraksi ringan atas implementasi asli penjadwal dari antrian kerja. Dalam kasus Slurm, AWS PCS antrian setara dengan partisi Slurm.

Pengguna mengirimkan pekerjaan ke antrian tempat mereka tinggal sampai mereka dapat dijadwalkan untuk berjalan pada node yang disediakan oleh satu atau lebih grup node komputasi. Sebuah AWS PCS cluster dapat memiliki beberapa antrian pekerjaan. Misalnya, Anda dapat membuat antrean yang menggunakan Instans EC2 Sesuai Permintaan Amazon untuk pekerjaan prioritas tinggi dan antrean lain yang menggunakan Instans EC2 Spot Amazon untuk pekerjaan dengan prioritas rendah.

Topik

- Membuat antrian di AWS PCS
- Memperbarui AWS PCS antrian
- Menghapus antrian di AWS PCS

Membuat antrian di AWS PCS

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda membuat antrian. AWS PCS

Prasyarat

- AWSPCSCluster - antrian hanya dapat dibuat dalam asosiasi dengan cluster tertentuPCS.
- Satu atau lebih Grup Node AWS PCS Komputasi - antrian harus dikaitkan dengan setidaknya satu grup node PCS komputasi.

Untuk membuat Antrian di AWS PCS

Anda dapat membuat antrian menggunakan AWS Management Console atau. AWS CLI

AWS Management Console

Untuk membuat antrian menggunakan konsol

1. Buka AWS PCS konsol di `https://console.aws.amazon.com/pcs/home#/clusters`
2. Pilih cluster tempat Anda ingin membuat antrian. Arahkan ke Antrian dan pilih Buat antrian.
3. Di bagian konfigurasi Antrian, berikan nilai berikut:
    a. Nama antrian — Nama untuk antrian Anda. Nama hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Itu harus dimulai dengan karakter alfabet dan tidak boleh lebih dari 25 karakter. Nama harus unik di dalam cluster.
    b. Compute node groups — Pilih satu atau beberapa grup node komputasi untuk melayani antrian ini. Grup node komputasi dapat dikaitkan dengan lebih dari satu antrian.
4. (Opsional) Di bawah Tag, tambahkan tag apa pun ke AWS PCS Antrian Anda
5. Pilih Buat antrean. Bidang Status akan menampilkan Membuat saat antrian sedang diatur. Pembuatan antrian dapat memakan waktu beberapa menit.

Direkomendasikan langkah selanjutnya

- Kirim pekerjaan ke antrian baru Anda

AWS CLI

Untuk membuat antrian menggunakan AWS CLI

Buat antrian Anda dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:

1. Ganti *region-code* dengan AWS Wilayah tempat Anda ingin membuat cluster Anda.
2. Ganti *my-queue* dengan nama untuk antrian Anda. Nama hanya dapat berisi karakter alfanumerik (peka huruf besar/kecil) dan tanda hubung. Itu harus dimulai dengan karakter alfabet dan tidak boleh lebih dari 25 karakter. Nama harus unik di dalam cluster.
3. Ganti *my-cluster* dengan nama atau clusterId klaster Anda.
4. Ganti nilainya `computeNodeGroupId` dengan pengidentifikasi grup node komputasi Anda sendiri. Perhatikan bahwa Anda tidak dapat menentukan nama grup node komputasi saat membuat antrian.

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

Diperlukan beberapa menit untuk membuat antrian. Anda dapat menanyakan status antrian Anda dengan perintah berikut. Anda tidak akan dapat mengirimkan pekerjaan ke antrian sampai statusnya tercapai`ACTIVE`.

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

Direkomendasikan langkah selanjutnya

- Kirim pekerjaan ke antrian baru Anda

Memperbarui AWS PCS antrian

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda memperbarui AWS PCS antrian.

Pertimbangan saat memperbarui antrian AWS PCS

Pembaruan antrian tidak akan memengaruhi pekerjaan yang sedang berjalan tetapi klaster mungkin tidak dapat menerima pekerjaan baru saat antrian sedang diperbarui.

Untuk memperbarui grup node AWS PCS komputasi

Anda dapat memperbarui grup node menggunakan AWS Management Console atau AWSCLI.

AWS Management Console

Untuk memperbarui antrian

1. Buka AWS PCS konsol di `https://console.aws.amazon.com/pcs/home#/clusters`
2. Pilih cluster tempat Anda ingin memperbarui antrian.
3. Arahkan ke Antrian, buka antrian yang ingin diperbarui, lalu pilih Edit.
4. Di bagian konfigurasi antrian, perbarui salah satu nilai berikut:
   - Grup node — Menambahkan atau menghapus grup node komputasi dari asosiasi dengan antrian.
   - Tag - Menambahkan atau menghapus tag untuk antrian.
5. Pilih Perbarui. Bidang Status akan menampilkan Memperbarui saat perubahan sedang diterapkan.

> **Important**
>
> Pembaruan antrian dapat memakan waktu beberapa menit.

AWS CLI

Untuk memperbarui antrian

1. Perbarui antrian Anda dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:
   a. Ganti *`region-code`* dengan Wilayah AWS yang Anda inginkan untuk membuat cluster Anda.
   b. Ganti *`my-queue`* dengan nama atau `computeNodeGroupId` antrian Anda.

c. Ganti *my-cluster* dengan nama atau `clusterId` klaster Anda.

d. Untuk mengubah asosiasi grup node komputasi, berikan daftar yang diperbarui untuk`--compute-node-group-configurations`.

- Misalnya, untuk menambahkan grup `computeNodeGroupExampleID2` node komputasi kedua:

```
--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. Diperlukan beberapa menit untuk memperbarui antrian. Anda dapat menanyakan status antrian Anda dengan perintah berikut. Anda tidak akan dapat mengirimkan pekerjaan ke antrian sampai statusnya tercapai`ACTIVE`.

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

Direkomendasikan langkah selanjutnya

- Kirim pekerjaan ke antrian Anda yang diperbarui.

Menghapus antrian di AWS PCS

Topik ini memberikan ikhtisar tentang cara menghapus antrian di AWS PCS.

Pertimbangan saat menghapus antrian

- Jika ada pekerjaan yang berjalan dalam antrian, mereka akan dihentikan oleh penjadwal saat antrian dihapus. Pekerjaan yang tertunda dalam antrian akan dibatalkan. Pertimbangkan untuk

menunggu pekerjaan dalam antrian untuk menyelesaikan atau menghentikan/membatalkannya secara manual menggunakan perintah asli penjadwal (seperti `scancel` untuk Slurm).

Hapus antrian

Anda dapat menggunakan AWS Management Console atau AWS CLI untuk menghapus antrian.

AWS Management Console

Untuk menghapus antrian

1. Buka AWS PCSkonsol.
2. Pilih cluster antrian.
3. Arahkan ke Antrian dan pilih antrian yang akan dihapus.
4. Pilih Hapus.
5. Bidang Status menunjukkan`Deleting`. Ini bisa memakan waktu beberapa menit untuk menyelesaikannya.

> **Note**
>
> Anda dapat menggunakan perintah asli penjadwal Anda untuk mengonfirmasi bahwa antrian dihapus. Misalnya, gunakan `sinfo` atau `squeue` untuk slurm.

AWS CLI

Untuk menghapus antrian

- Gunakan perintah berikut untuk menghapus antrian, dengan penggantian ini:
  - Ganti *`region-code`* dengan cluster Wilayah AWS Anda ada di.
  - Ganti *`my-queue`* dengan nama atau ID antrian Anda.
  - Ganti *`my-cluster`* dengan nama atau ID cluster Anda.

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

```
    --cluster-identifier my-cluster
```

Diperlukan beberapa menit untuk menghapus antrian.

> **Note**
>
> Anda dapat menggunakan perintah asli penjadwal Anda untuk mengonfirmasi bahwa antrian dihapus. Misalnya, gunakan `sinfo` atau `squeue` untuk slurm.

AWS PCSsimpul masuk

Sebuah AWS PCS cluster biasanya membutuhkan setidaknya 1 node login untuk mendukung akses interaktif dan manajemen pekerjaan. Cara untuk mencapai ini adalah dengan grup node AWS PCS komputasi statis yang dikonfigurasi untuk kemampuan node login. Anda juga dapat mengonfigurasi EC2 instance mandiri untuk bertindak sebagai node login.

Topik

- Menggunakan grup node AWS PCS komputasi untuk menyediakan node login
- Menggunakan instance mandiri sebagai node login AWS PCS

Menggunakan grup node AWS PCS komputasi untuk menyediakan node login

Topik ini memberikan ikhtisar opsi konfigurasi yang disarankan dan menjelaskan apa yang harus dipertimbangkan saat Anda menggunakan grup node AWS PCS komputasi untuk menyediakan akses interaktif yang persisten ke klaster Anda.

Membuat grup node AWS PCS komputasi untuk node login

Secara operasional, ini tidak jauh berbeda dengan membuat grup node komputasi biasa. Namun, ada beberapa pilihan konfigurasi utama yang dibuat:

- Tetapkan konfigurasi penskalaan statis setidaknya satu EC2 instance dalam grup node komputasi.
- Pilih opsi pembelian sesuai permintaan untuk menghindari instans Anda direklamasi.
- Pilih nama informatif untuk grup node komputasi, seperti login.

- Jika Anda ingin instance node login dapat diakses di luar AndaVPC, pertimbangkan untuk menggunakan subnet publik.
- Jika Anda bermaksud mengizinkan SSH akses, template peluncuran akan membutuhkan grup keamanan yang mengekspos SSH port ke alamat IP pilihan Anda.
- Profil IAM instance seharusnya hanya memiliki AWS izin yang Anda ingin pengguna akhir Anda miliki. Lihat IAMprofil instance untuk Layanan Komputasi AWS Paralel untuk detail.
- Pertimbangkan untuk mengizinkan AWS Systems Manager Session Manager mengelola instans login Anda.
- Pertimbangkan untuk membatasi akses ke AWS kredensi instans hanya untuk pengguna administratif
- Pilih jenis instance yang lebih murah daripada grup node komputasi biasa, karena node login akan berjalan terus menerus.
- Gunakan yang sama (atau turunan) AMI seperti untuk grup node komputasi Anda yang lain untuk membantu memastikan semua instance memiliki perangkat lunak yang sama yang diinstal. Untuk informasi selengkapnya tentang penyesuaianAMIs, lihat Gambar Mesin Amazon (AMIs) untuk AWS PCS
- Konfigurasikan sistem file jaringan yang sama (AmazonEFS, Amazon FSx untuk Lustre, dll.) Mount pada node login Anda seperti pada instance komputasi Anda. Untuk informasi selengkapnya, lihat Menggunakan sistem berkas jaringan dengan AWS PCS.

Akses node login Anda

Setelah grup node komputasi baru Anda mencapai ACTIVE status, Anda dapat menemukan EC2 instance yang telah dibuat dan masuk ke dalamnya. Untuk informasi selengkapnya, lihat Menemukan instance grup node komputasi di AWS PCS.

Memperbarui grup node AWS PCS komputasi untuk node login

Anda dapat memperbarui grup node login menggunakan UpdateComputeNodeGroup. Sebagai bagian dari proses pembaruan grup node, instance yang berjalan akan diganti. Perhatikan bahwa ini akan mengganggu sesi atau proses pengguna aktif apa pun pada instance. Menjalankan atau mengantri pekerjaan Slurm tidak akan terpengaruh. Untuk informasi selengkapnya, lihat Memperbarui grup node AWS PCS komputasi.

Anda juga dapat mengedit template peluncuran yang digunakan oleh grup node komputasi Anda. Anda harus menggunakan UpdateComputeNodeGroup untuk menerapkan template peluncuran

yang diperbarui ke grup node komputasi. EC2Instance baru yang diluncurkan di grup node komputasi menggunakan template peluncuran yang diperbarui. Untuk informasi selengkapnya, lihat Menggunakan template EC2 peluncuran Amazon dengan AWS PCS.

Menghapus grup node AWS PCS komputasi untuk node login

Anda dapat memperbarui grup node login menggunakan mekanisme grup node delete compute di AWS PCS. Menjalankan instance akan dihentikan sebagai bagian dari penghapusan grup node. Harap dicatat bahwa ini akan mengganggu sesi atau proses pengguna aktif apa pun pada instance. Menjalankan atau mengantri pekerjaan Slurm tidak akan terpengaruh. Untuk informasi selengkapnya, lihat Menghapus grup node komputasi di AWS PCS.

Menggunakan instance mandiri sebagai node login AWS PCS

Anda dapat mengatur EC2 instance independen untuk berinteraksi dengan penjadwal AWS PCS Slurm klaster. Ini berguna untuk membuat node login, workstation, atau host manajemen alur kerja khusus yang bekerja dengan AWS PCS cluster tetapi beroperasi di luar manajemen. AWS PCS Untuk melakukan ini, setiap instance mandiri harus:

1. Memiliki versi perangkat lunak Slurm yang kompatibel diinstal.
2. Dapat terhubung ke titik akhir Slurmctld AWS PCS cluster.
3. Minta Slurm Auth dan Cred Kiosk Daemon (`sackd`) dikonfigurasi dengan benar dengan titik akhir dan rahasia cluster. AWS PCS Untuk informasi lebih lanjut, lihat sackd di dokumentasi Slurm.

Tutorial ini membantu Anda mengonfigurasi instance independen yang terhubung ke AWS PCS cluster.

Daftar Isi

Langkah 1 - Ambil alamat dan rahasia untuk cluster target AWS PCS

Ambil detail tentang AWS PCS cluster target menggunakan AWS CLI dengan perintah yang mengikuti. Sebelum menjalankan perintah, buat penggantian berikut:

- Ganti *region-code* dengan Wilayah AWS tempat cluster target berjalan.
- Ganti *cluster-ident* dengan nama atau pengenal untuk cluster target

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

Perintah akan mengembalikan output yang mirip dengan contoh ini.

```
{
    "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"
        }
    }
}
```

Dalam contoh ini, titik akhir pengontrol Slurm cluster memiliki alamat IP `10.3.149.220` dan berjalan di port. 6817 Ini `secretArn` akan digunakan dalam langkah-langkah selanjutnya untuk mengambil rahasia cluster. Alamat IP dan port akan digunakan pada langkah-langkah selanjutnya untuk mengkonfigurasi `sackd` layanan.

Langkah 2 - Luncurkan sebuah EC2 instance

Untuk meluncurkan sebuah EC2 instance

1. Buka EC2konsol Amazon.
2. Di panel navigasi, pilih Instans, lalu pilih Luncurkan Instans untuk membuka wizard peluncuran instans baru.
3. (Opsional) Di bagian Nama dan tag, berikan nama untuk contoh, sepertiPCS-LoginNode. Nama ditetapkan ke instans sebagai tanda sumber daya (Name=PCS-LoginNode).
4. Di bagian Aplikasi dan Gambar OS, pilih AMI untuk salah satu sistem operasi yang didukung oleh AWS PCS. Untuk informasi selengkapnya, lihat Sistem operasi yang didukung.
5. Di bagian Jenis instans, pilih jenis instans yang didukung. Untuk informasi selengkapnya, lihat Tipe instans yang didukung.
6. Di bagian Key pair, pilih SSH key pair yang akan digunakan untuk instance.
7. Di bagian Pengaturan jaringan:

    - Pilih Edit.

        i. Pilih VPC AWS PCS cluster Anda.

        ii. Untuk Firewall (grup keamanan), pilih Pilih grup keamanan yang ada.

            A. Pilih grup keamanan yang mengizinkan lalu lintas antara instance dan pengontrol AWS PCS Slurm cluster target. Untuk informasi selengkapnya, lihat Persyaratan dan pertimbangan kelompok keamanan.

            B. (Opsional) Pilih grup keamanan yang memungkinkan SSH akses masuk ke instans Anda.

8. Di bagian Penyimpanan, konfigurasikan volume penyimpanan sesuai kebutuhan. Pastikan untuk mengonfigurasi ruang yang cukup untuk menginstal aplikasi dan pustaka untuk mengaktifkan kasus penggunaan Anda.
9. Di bagian Advanced, pilih IAM peran yang memungkinkan akses ke rahasia cluster. Untuk informasi selengkapnya, lihat Dapatkan rahasia cluster Slurm.
10. Di panel Ringkasan, pilih Launch instance.

Langkah 3 - Instal Slurm pada instance

Ketika instans telah diluncurkan dan menjadi aktif, sambungkan ke instans menggunakan mekanisme pilihan Anda. Gunakan installer Slurm yang disediakan oleh AWS untuk menginstal Slurm ke instance. Untuk informasi selengkapnya, lihat Pemasang slurm.

Unduh penginstal Slurm, buka kompres, dan gunakan `installer.sh` skrip untuk menginstal Slurm. Untuk informasi selengkapnya, lihat Langkah 3 - Instal Slurm.

Langkah 4 - Ambil dan simpan rahasia cluster

Instruksi ini membutuhkan AWS CLI. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui ke versi terbaru dari Panduan AWS Command Line Interface Pengguna untuk Versi 2. AWS CLI

Simpan rahasia cluster dengan perintah berikut.

- Buat direktori konfigurasi untuk Slurm.

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

- Mengambil, memecahkan kode, dan menyimpan rahasia cluster. Sebelum menjalankan perintah ini, ganti *region-code* dengan Wilayah tempat cluster target berjalan, dan ganti *secret-arn* dengan nilai untuk `secretArn` diambil pada Langkah 1.

```
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**
>
> Dalam lingkungan multipengguna, setiap pengguna dengan akses ke instance mungkin dapat mengambil rahasia cluster jika mereka dapat mengakses layanan metadata instance (). IMDS Ini, pada gilirannya, dapat memungkinkan mereka untuk meniru pengguna lain. Pertimbangkan untuk membatasi akses IMDS ke pengguna root atau administratif saja. Atau, pertimbangkan untuk menggunakan mekanisme berbeda yang tidak bergantung pada profil instance untuk mengambil dan mengonfigurasi rahasia.

- Tetapkan kepemilikan dan izin pada file kunci Slurm.

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

> **Note**
>
> Kunci Slurm harus dimiliki oleh pengguna dan grup tempat `sackd` layanan berjalan.

Langkah 5 - Konfigurasikan koneksi ke AWS PCS cluster

Untuk membuat koneksi ke AWS PCS cluster, luncurkan `sackd` sebagai layanan sistem dengan mengikuti langkah-langkah ini.

1. Siapkan file lingkungan untuk `sackd` layanan dengan perintah berikut. Sebelum menjalankan perintah, ganti *`ip-address`* and *`port`* dengan nilai yang diambil dari titik akhir pada Langkah 1.

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

2. Buat file `systemd` layanan untuk mengelola `sackd` proses.

```
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. Tetapkan kepemilikan file `sackd` layanan.

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

4. Aktifkan `sackd` layanan.

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

5. Mulai layanan `sackd`.

```
sudo systemctl start sackd
```

Langkah 6 - (Opsional) Uji koneksi

Konfirmasikan bahwa `sackd` layanan sedang berjalan. Berikut adalah contoh output. Jika ada kesalahan, biasanya akan muncul di sini.

```
[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
```

Konfirmasikan koneksi ke cluster bekerja menggunakan perintah klien Slurm seperti `sinfo` dan. `squeue` Berikut adalah contoh output dari`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]
```

Anda juga harus dapat mengirimkan pekerjaan. Misalnya, perintah yang mirip dengan contoh ini akan meluncurkan pekerjaan interaktif pada 1 node di cluster.

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

AWS PCSJaringan

AWS PCSCluster Anda dibuat di AmazonVPC. Bab ini mencakup topik-topik berikut tentang jaringan untuk penjadwal dan node cluster Anda.

Kecuali untuk memilih subnet untuk meluncurkan instance, Anda harus menggunakan template EC2 peluncuran untuk mengonfigurasi jaringan untuk grup node AWS PCS komputasi. Untuk informasi selengkapnya tentang template peluncuran, lihat[Menggunakan template EC2 peluncuran Amazon dengan AWS PCS](#).

Topik

- [AWS PCSVPCdan persyaratan dan pertimbangan subnet](#)
- [Membuat VPC untuk AWS PCS cluster Anda](#)
- [Kelompok keamanan di AWS PCS](#)
- [Beberapa antarmuka jaringan di AWS PCS](#)
- [Grup penempatan untuk EC2 instance di AWS PCS](#)

- Menggunakan Adaptor Kain Elastis (EFA) dengan AWS PCS

AWS PCSVPCdan persyaratan dan pertimbangan subnet

Saat Anda membuat AWS PCS cluster, Anda menentukan subnet di dalamnya. VPC VPC Topik ini memberikan gambaran umum tentang persyaratan dan pertimbangan AWS PCS khusus untuk VPC dan subnet yang Anda gunakan dengan cluster Anda. Jika Anda tidak VPC harus menggunakan AWS PCS, Anda dapat membuatnya menggunakan AWS CloudFormation template AWS yang disediakan. Untuk informasi selengkapnyaVPCs, lihat Virtual private cloud (VPC) di Panduan VPC Pengguna Amazon.

VPCpersyaratan dan pertimbangan

Saat Anda membuat cluster, VPC yang Anda tentukan harus memenuhi persyaratan dan pertimbangan berikut:

- VPCHarus memiliki cukup jumlah alamat IP yang tersedia untuk cluster, node apa pun, dan sumber daya cluster lain yang ingin Anda buat. Untuk informasi selengkapnya, lihat Pengalamatan IP untuk subnet Anda VPCs dan subnet di VPCPanduan Pengguna Amazon.
- VPCHarus memiliki DNS nama host dan dukungan DNS resolusi. Jika tidak, node tidak dapat mendaftarkan cluster pelanggan. Untuk informasi selengkapnya, lihat DNSatribut untuk Anda VPC di Panduan VPC Pengguna Amazon.
- VPCMungkin memerlukan VPC titik akhir yang digunakan AWS PrivateLink untuk dapat menghubungi. AWS PCS API Untuk informasi selengkapnya, lihat Menyambungkan VPC ke layanan yang digunakan AWS PrivateLink di Panduan VPC Pengguna Amazon.

Persyaratan dan pertimbangan subnet

Saat Anda membuat cluster Slurm, AWS PCS buat Elastic Network Interface (ENI) di subnet yang Anda tentukan. Antarmuka jaringan ini memungkinkan komunikasi antara pengontrol penjadwal dan pelangganVPC. Antarmuka jaringan juga memungkinkan Slurm untuk berkomunikasi dengan komponen yang digunakan di akun pelanggan. Anda hanya dapat menentukan subnet untuk cluster pada waktu pembuatan.

Persyaratan subnet untuk cluster

Subnet yang Anda tentukan saat membuat cluster harus memenuhi persyaratan berikut:

- Subnet harus memiliki setidaknya 1 alamat IP untuk digunakan oleh AWS PCS.
- Subnet tidak dapat berada di AWS Outposts, AWS Wavelength, atau Zona AWS Lokal.
- Subnet dapat bersifat publik atau pribadi. Kami menyarankan Anda menentukan subnet pribadi, jika memungkinkan. Subnet publik adalah subnet dengan tabel rute yang mencakup rute ke gateway internet; subnet pribadi adalah subnet dengan tabel rute yang tidak menyertakan rute ke gateway internet.

Persyaratan subnet untuk node

Anda dapat menyebarkan node dan sumber daya cluster lainnya ke subnet yang Anda tentukan saat membuat AWS PCS klaster, dan ke subnet lain yang sama. VPC

Setiap subnet yang Anda gunakan node dan sumber daya cluster harus memenuhi persyaratan berikut:

- Anda harus memastikan bahwa subnet memiliki cukup alamat IP yang tersedia untuk menyebarkan semua node dan sumber daya cluster.
- Jika Anda berencana untuk menyebarkan node ke subnet publik, subnet tersebut harus menetapkan alamat publik secara otomatisIPv4.
- Jika subnet tempat Anda menyebarkan node adalah subnet pribadi dan tabel rutenya tidak menyertakan rute ke perangkat terjemahan alamat jaringan (NAT) (IPv4), tambahkan VPC titik akhir menggunakan AWS PrivateLink ke pelanggan. VPC VPCendpoint diperlukan untuk semua AWS layanan yang dihubungi node. Satu-satunya titik akhir yang diperlukan adalah AWS PCS untuk mengizinkan node memanggil `registerNodeGroupInstances` API tindakan.
- Status subnet publik atau pribadi tidak berdampak AWS PCS; titik akhir yang diperlukan harus dapat dijangkau.

Membuat VPC untuk AWS PCS cluster Anda

Anda dapat membuat Amazon Virtual Private Cloud (AmazonVPC) untuk cluster Anda dalam AWS Parallel Computing Service (AWS PCS).

Gunakan Amazon VPC untuk meluncurkan VPC sumber daya ke jaringan virtual yang telah Anda tentukan. Jaringan virtual ini sangat mirip dengan jaringan tradisional yang mungkin Anda operasikan di pusat data Anda sendiri. Namun, ia datang dengan manfaat menggunakan infrastruktur yang dapat diskalakan dari Amazon Web Services. Kami menyarankan Anda memiliki pemahaman menyeluruh

tentang VPC layanan Amazon sebelum menerapkan VPC kluster produksi. Untuk informasi lebih lanjut, lihat Apa itu AmazonVPC? dalam mode visual penulis. Panduan VPC Pengguna Amazon.

PCSCluster, node, dan sumber daya pendukung (seperti sistem file dan layanan direktori) digunakan di Amazon VPC Anda. Jika Anda ingin menggunakan Amazon yang adaPCS, Amazon VPC harus memenuhi persyaratan yang dijelaskan dalamAWS PCSVPCdan persyaratan dan pertimbangan subnet. Topik ini menjelaskan cara membuat VPC yang memenuhi PCS persyaratan menggunakan AWS CloudFormation templat AWS yang disediakan. Setelah menerapkan template, Anda dapat melihat sumber daya yang dibuat oleh template untuk mengetahui dengan tepat sumber daya apa yang dibuatnya, dan konfigurasi sumber daya tersebut.

Prasyarat

Untuk membuat Amazon VPCPCS, Anda harus memiliki IAM izin yang diperlukan untuk membuat VPC sumber daya Amazon. Sumber daya ini adalahVPCs, subnet, grup keamanan, tabel rute dan rute, dan internet dan NAT gateway. Untuk informasi selengkapnya, lihat Membuat VPC dengan subnet publik di Panduan VPC Pengguna Amazon. Untuk meninjau daftar lengkap AmazonEC2, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon EC2 di Referensi Otorisasi Layanan.

Buat Amazon VPC

Buat VPC dengan menyalin dan menempelkan yang sesuai URL untuk Wilayah AWS tempat Anda akan menggunakanPCS. Anda juga dapat mengunduh AWS CloudFormation templat dan mengunggahnya sendiri ke AWS CloudFormation konsol.

- AS Timur (Virginia N.) (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
```

- AS Timur (Ohio) (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
```

- AS Barat (Oregon) (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
```

- Template saja

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

Untuk membuat Amazon VPC untuk PCS

1. Buka template di AWS CloudFormation konsol.

   > **Note**
   >
   > Ini sudah diisi sebelumnya dalam template sehingga Anda dapat membiarkannya sebagai nilai default.

2. Di bawah Berikan nama tumpukan, lalu nama Stack, masukkan`hpc-networking`.
3. Di bawah parameter, masukkan detail berikut:
   a. Di bawah VPC, lalu CidrBlock, masukkan `10.3.0.0/16`
   b. Di bawah Subnet A:
      i. Kemudian CidrPublicSubnetA, masukkan `10.3.0.0/20`
      ii. Kemudian CidrPrivateSubnetA, masukkan `10.3.128.0/20`
   c. Di bawah Subnet B:
      i. Kemudian CidrPublicSubnetB, masukkan `10.3.16.0/20`
      ii. Kemudian CidrPrivateSubnetA, masukkan `10.3.144.0/20`
   d. Di bawah Subnet C:
      i. Untuk ProvisionSubnetsC, pilih`True`.

> **Note**
>
> Jika Anda membuat VPC di Wilayah yang memiliki kurang dari tiga Availability Zone, opsi ini akan diabaikan jika disetel ke`True`.

ii. Kemudian CidrPublicSubnetB, masukkan `10.3.32.0/20`

iii. Kemudian CidrPrivateSubnetA, masukkan `10.3.160.0/20`

4. Di bawah Kemampuan, centang kotak untuk saya akui yang AWS CloudFormation mungkin membuat IAM sumber daya.

Pantau status AWS CloudFormation tumpukan. Ketika mencapai`CREATE_COMPLETE`, VPC sumber daya siap untuk Anda gunakan.

> **Note**
>
> Untuk melihat semua sumber daya yang dibuat AWS CloudFormation template, buka AWS CloudFormation konsol. Pilih `hpc-networking` tumpukan dan kemudian pilih tab Sumber Daya.

Kelompok keamanan di AWS PCS

Grup keamanan di Amazon EC2 bertindak sebagai firewall virtual untuk mengontrol lalu lintas masuk dan keluar ke instance. Gunakan template peluncuran untuk grup node AWS PCS komputasi untuk menambah atau menghapus grup keamanan ke instance-nya. Jika template peluncuran Anda tidak berisi antarmuka jaringan apa pun, gunakan `SecurityGroupIds` untuk menyediakan daftar grup keamanan. Jika template peluncuran Anda mendefinisikan antarmuka jaringan, Anda harus menggunakan `Groups` parameter untuk menetapkan grup keamanan ke setiap antarmuka jaringan. Untuk informasi selengkapnya tentang template peluncuran, lihatMenggunakan template EC2 peluncuran Amazon dengan AWS PCS.

> **Note**
>
> Perubahan pada konfigurasi grup keamanan dalam template peluncuran hanya memengaruhi instance baru yang diluncurkan setelah grup node komputasi diperbarui.

Persyaratan dan pertimbangan kelompok keamanan

AWS PCSmembuat Antarmuka Jaringan Elastis (ENI) lintas akun di subnet yang Anda tentukan saat membuat cluster. Ini memberikan HPC penjadwal, yang berjalan di akun yang dikelola oleh AWS, jalur untuk berkomunikasi dengan EC2 instance yang diluncurkan oleh. AWS PCS Anda harus menyediakan grup keamanan untuk itu ENI yang memungkinkan komunikasi 2 arah antara penjadwal ENI dan instance cluster EC2 Anda.

Cara mudah untuk mencapai ini adalah dengan membuat grup keamanan referensi mandiri permisif yang memungkinkan lalu lintas TCP /IP di semua port antara semua anggota grup. Anda dapat melampirkan ini ke cluster dan EC2 instance grup node.

Contoh konfigurasi grup keamanan permisif

Jenis aturan	Protokol	Port	Sumber	Tujuan
Ke dalam	Semua	Semua	Diri Sendiri	
Ke luar	Semua	Semua		0.0.0.0/0
Ke luar	Semua	Semua		Diri Sendiri

Aturan ini memungkinkan semua lalu lintas mengalir bebas antara pengontrol Slurm dan node, memungkinkan semua lalu lintas keluar ke tujuan mana pun, dan memungkinkan lalu lintas. EFA

Contoh konfigurasi grup keamanan yang membatasi

Anda juga dapat membatasi port terbuka antara cluster dan node komputasinya. Untuk penjadwal Slurm, grup keamanan yang dilampirkan ke cluster Anda harus mengizinkan port berikut:

- 6817 - aktifkan koneksi masuk ke `slurmctld` dari instance EC2
- 6818 - aktifkan koneksi keluar dari `slurmctld` untuk `slurmd` berjalan pada instance EC2

Grup keamanan yang dilampirkan ke node komputasi Anda harus mengizinkan port berikut:

- 6817 - aktifkan koneksi keluar `slurmctld` dari EC2 instance.
- 6818 - aktifkan koneksi masuk dan keluar ke `slurmd` dari `slurmctld` dan dari instance grup `slurmd` node

- 60001—63000 - koneksi masuk dan keluar antara instance grup node untuk mendukung `srun`
- EFAlalu lintas antar instance grup node. Untuk informasi selengkapnya, lihat Mempersiapkan grup keamanan yang EFA diaktifkan di Panduan Pengguna untuk Instans Linux
- Lalu lintas antar simpul lain yang diperlukan oleh beban kerja Anda

Beberapa antarmuka jaringan di AWS PCS

Beberapa EC2 contoh memiliki beberapa kartu jaringan. Hal ini memungkinkan mereka untuk memberikan kinerja jaringan yang lebih tinggi, termasuk kemampuan bandwidth di atas 100 Gbps dan peningkatan penanganan paket. Untuk informasi selengkapnya tentang instans dengan beberapa kartu jaringan, lihat Antarmuka jaringan elastis di Panduan Pengguna Amazon Elastic Compute Cloud.

Konfigurasikan kartu jaringan tambahan untuk instance dalam grup node AWS PCS komputasi dengan menambahkan antarmuka jaringan ke template peluncurannyaEC2. Di bawah ini adalah contoh template peluncuran yang memungkinkan dua kartu jaringan, seperti dapat ditemukan pada sebuah `hpc7a.96xlarge` instance. Perhatikan detail berikut:

- Subnet untuk setiap antarmuka jaringan harus sama dengan yang Anda pilih saat mengkonfigurasi grup node AWS PCS komputasi yang akan menggunakan template peluncuran.
- Perangkat jaringan utama, di mana komunikasi jaringan rutin seperti SSH dan HTTPS lalu lintas akan terjadi, ditetapkan dengan menetapkan a `DeviceIndex0`. Antarmuka jaringan lainnya memiliki file`DeviceIndex`. 1 Hanya ada satu antarmuka jaringan utama—semua antarmuka lainnya bersifat sekunder.
- Semua antarmuka jaringan harus memiliki yang unik`NetworkCardIndex`. Praktik yang disarankan adalah memberi nomor secara berurutan seperti yang ditentukan dalam templat peluncuran.
- Grup keamanan untuk setiap antarmuka jaringan diatur menggunakan`Groups`. Dalam contoh ini, grup SSH keamanan masuk (`sg-`*`SshSecurityGroupId`*) ditambahkan ke antarmuka jaringan utama, serta grup keamanan yang mengaktifkan komunikasi dalam-cluster (). `sg-`*`ClusterSecurityGroupId`* Akhirnya, grup keamanan yang memungkinkan koneksi keluar ke internet (`sg-`*`InternetOutboundSecurityGroupId`*) ditambahkan ke antarmuka primer dan sekunder.

```
{
```

```
    "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"]
        }
    ]
}
```

Grup penempatan untuk EC2 instance di AWS PCS

Anda dapat menggunakan grup penempatan untuk memengaruhi penempatan EC2 instance agar sesuai dengan kebutuhan beban kerja yang berjalan pada mereka.

Jenis grup penempatan

- Cluster — Paket instance berdekatan di Availability Zone untuk mengoptimalkan komunikasi latensi rendah.
- Partisi — Menyebarkan instance di seluruh partisi logis untuk membantu memaksimalkan ketahanan.
- Spread — Menegakkan secara ketat bahwa sejumlah kecil instance diluncurkan pada perangkat keras yang berbeda, yang juga dapat membantu ketahanan.

Untuk informasi selengkapnya, lihat Grup penempatan untuk EC2 instans Amazon Anda di Panduan Pengguna Amazon Elastic Compute Cloud.

Kami menyarankan Anda menyertakan grup penempatan klaster saat Anda mengonfigurasi grup node AWS PCS komputasi untuk menggunakan Elastic Fabric Adapter (EFA).

Untuk membuat grup penempatan cluster yang bekerja dengan EFA

1. Buat grup penempatan dengan cluster tipe untuk grup node komputasi.
   - Gunakan AWS CLI perintah berikut:

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

   - Anda juga dapat menggunakan CloudFormation template untuk membuat grup penempatan. Untuk informasi selengkapnya, lihat Bekerja dengan CloudFormation templat di Panduan AWS CloudFormation Pengguna. Unduh templat dari berikut ini URL dan unggah ke CloudFormation konsol.

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

2. Sertakan grup penempatan dalam template EC2 peluncuran untuk grup node AWS PCS komputasi.

Menggunakan Adaptor Kain Elastis (EFA) dengan AWS PCS

Elastic Fabric Adapter (EFA) adalah interkoneksi jaringan canggih berkinerja tinggi AWS yang dapat Anda lampirkan ke EC2 instans Anda untuk mempercepat aplikasi Komputasi Kinerja Tinggi (HPC) dan pembelajaran mesin. Mengaktifkan aplikasi Anda berjalan di AWS PCS cluster dengan EFA melibatkan konfigurasi instance grup node AWS PCS komputasi untuk digunakan sebagai berikut. EFA

Daftar Isi

- (Opsional) Gunakan CloudFormation templat untuk membuat templat peluncuran EFA yang diaktifkan

Instal EFA pada AWS PCS -kompatibel AMI

Yang AMI digunakan dalam grup node AWS PCS komputasi harus memiliki EFA driver diinstal dan dimuat. Untuk informasi tentang cara membuat kustom AMI dengan EFA perangkat lunak yang diinstal, lihatGambar Mesin Amazon Kustom (AMIs) untuk AWS PCS.

Identifikasi EFA instance yang diaktifkan EC2

Untuk menggunakanEFA, semua jenis instance yang diizinkan untuk grup AWS PCS komputasi harus mendukungEFA, dan harus memiliki jumlah yang sama vCPUs (dan GPUs jika sesuai). Untuk daftar instans EFA yang diaktifkan, lihat Elastic Fabric Adapter untuk HPC dan beban kerja ML di Amazon di Panduan Pengguna Amazon EC2 Elastic Compute Cloud. Anda juga dapat menggunakan AWS CLI untuk melihat daftar jenis instance yang mendukungEFA. Ganti *region-code* dengan Wilayah AWS tempat yang Anda gunakan AWS PCS, sepertius-east-1.

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

Tentukan berapa banyak antarmuka jaringan yang tersedia

Beberapa EC2 contoh memiliki beberapa kartu jaringan. Ini memungkinkan mereka untuk memiliki banyakEFAs. Untuk informasi selengkapnya, lihat Beberapa antarmuka jaringan di AWS PCS.

Buat grup keamanan untuk mendukung EFA komunikasi

AWS CLI

Anda dapat menggunakan AWS CLI perintah berikut untuk membuat grup keamanan yang mendukungEFA. Perintah tersebut mengeluarkan ID grup keamanan. Lakukan penggantian berikut:

- *region-code*— Tentukan Wilayah AWS di mana Anda menggunakan AWS PCS, sepertius-east-1.

- *vpc-id*— Tentukan ID VPC yang Anda gunakan untuk AWS PCS.
- *efa-group-name*— Berikan nama yang Anda pilih untuk grup keamanan.

```
aws ec2 create-security-group \
    --group-name efa-group-name \
    --description "Security group to enable EFA traffic" \
    --vpc-id vpc-id \
    --region region-code
```

Gunakan perintah berikut untuk melampirkan aturan grup keamanan masuk dan keluar. Lakukan penggantian berikut:

- *efa-secgroup-id*— Berikan ID grup EFA keamanan yang baru saja Anda buat.

```
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

Anda dapat menggunakan CloudFormation template untuk membuat grup keamanan yang mendukungEFA. Unduh template dari berikut iniURL, lalu unggah ke AWS CloudFormation konsol.

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

Dengan templat terbuka di AWS CloudFormation konsol, masukkan opsi berikut.

- Di bawah Berikan nama tumpukan
  - Di bawah nama Stack, masukkan nama seperti`efa-sg-stack`.
- Di bawah Parameter

- Di bawah SecurityGroupName, masukkan nama seperti`efa-sg`.
- Di bawah VPC, pilih VPC tempat yang akan Anda gunakan AWS PCS.

Selesai membuat CloudFormation tumpukan dan memantau statusnya. Ketika mencapai `CREATE_COMPLETE` grup EFA keamanan siap digunakan.

(Opsional) Buat grup penempatan

Disarankan untuk meluncurkan semua instance yang digunakan EFA dalam grup penempatan cluster untuk meminimalkan jarak fisik di antara mereka. Kami menyarankan Anda membuat grup penempatan untuk setiap grup node komputasi tempat Anda akan menggunakanEFA. Lihat Grup penempatan untuk EC2 instance di AWS PCS untuk membuat grup penempatan untuk grup node komputasi Anda.

Membuat atau memperbarui template EC2 peluncuran

EFAantarmuka jaringan diatur dalam template EC2 peluncuran untuk grup node AWS PCS komputasi. Jika ada beberapa kartu jaringan, beberapa EFAs dapat dikonfigurasi. Grup EFA keamanan dan grup penempatan opsional juga disertakan dalam template peluncuran.

Berikut adalah contoh template peluncuran untuk instance dengan dua kartu jaringan, seperti hpc7a.96xlarge. Instans akan diluncurkan di `subnet-`*`SubnetID1`* dalam grup `pg-`*`PlacementGroupId1`* penempatan cluster.

Grup keamanan harus ditambahkan secara khusus ke setiap EFA antarmuka. Setiap EFA kebutuhan kelompok keamanan yang memungkinkan EFA lalu lintas (`sg-`*`EfaSecGroupId`*). Kelompok keamanan lain, terutama yang menangani lalu lintas reguler seperti SSH atauHTTPS, hanya perlu dilampirkan ke antarmuka jaringan utama (ditunjuk oleh a `DeviceIndex0`). Peluncuran templat tempat antarmuka jaringan didefinisikan tidak mendukung pengaturan grup keamanan menggunakan `SecurityGroupIds` parameter—Anda harus menetapkan nilai untuk setiap antarmuka jaringan yang Anda `Groups` konfigurasikan.

```
{
    "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"]
        }
    ]
}
```

Membuat atau memperbarui grup node komputasi

Buat atau perbarui grup node AWS PCS komputasi dengan instance yang memiliki jumlah yang samavCPUs, arsitektur prosesor yang sama, dan yang semuanya mendukung. EFA Konfigurasikan grup node komputasi untuk menggunakan EFA perangkat lunak yang diinstal di dalamnya, dan untuk menggunakan template peluncuran yang mengatur antarmuka jaringan EFA yang diaktifkan. AMI

(Opsional) Tes EFA

Anda dapat mendemonstrasikan komunikasi EFA -enabled antara dua node dalam grup node komputasi dengan menjalankan `fi_pingpong` program, yang disertakan dalam instalasi EFA perangkat lunak. Jika tes ini berhasil, kemungkinan itu EFA dikonfigurasi dengan benar.

Untuk memulai, Anda memerlukan dua instance yang berjalan di grup node komputasi. Jika grup node komputasi Anda menggunakan kapasitas statis, seharusnya sudah ada instance yang tersedia. Untuk grup node komputasi yang menggunakan kapasitas dinamis, Anda dapat meluncurkan dua node menggunakan `salloc` perintah. Berikut adalah contoh dari cluster dengan grup node dinamis bernama `hpc7g` terkait dengan antrian bernama`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
```

Cari tahu alamat IP untuk dua node yang dialokasikan menggunakan`scontrol`. Dalam contoh berikut, alamatnya adalah `10.3.140.69` untuk `hpc7g-1` dan `10.3.132.211` untuk`hpc7g-2`.

```
% 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
```

Connect ke salah satu node (dalam kasus ini,`hpc7g-1`) menggunakan SSH (atauSSM). Perhatikan bahwa ini adalah alamat IP internal, jadi Anda mungkin perlu terhubung dari salah satu node login Anda jika Anda menggunakannyaSSH. Ketahuilah juga bahwa instance perlu dikonfigurasi dengan SSH kunci melalui templat peluncuran grup node komputasi.

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

Sekarang, luncurkan `fi_pingpong` dalam mode server.

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

Connect ke instance kedua (`hpc7g-2`).

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

Jalankan `fi_pingpong` dalam mode klien, sambungkan ke server aktif`hpc7g-1`. Anda akan melihat output yang menyerupai contoh di bawah ini.

```
% /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
```

(Opsional) Gunakan CloudFormation templat untuk membuat templat peluncuran EFA yang diaktifkan

Karena ada beberapa dependensi untuk menyiapkanEFA, CloudFormation template telah disediakan yang dapat Anda gunakan untuk mengkonfigurasi grup node komputasi. Ini mendukung contoh dengan hingga empat kartu jaringan. Untuk mempelajari lebih lanjut tentang instans dengan beberapa kartu jaringan, lihat Antarmuka jaringan elastis di Panduan Pengguna Amazon Elastic Compute Cloud.

Unduh CloudFormation template dari berikut iniURL, lalu unggah ke CloudFormation konsol di Wilayah AWS tempat Anda menggunakan AWS PCS.

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

Dengan templat terbuka di AWS CloudFormation konsol, masukkan nilai berikut. Perhatikan bahwa template akan memberikan beberapa nilai parameter default—Anda dapat membiarkannya sebagai nilai defaultnya.

- Di bawah Berikan nama tumpukan
  - Di bawah nama Stack, masukkan nama deskriptif. Kami merekomendasikan untuk memasukkan nama yang akan Anda pilih untuk grup node AWS PCS komputasi Anda, seperti. *NODEGROUPNAME*`-efa-lt`
- Di bawah Parameter
  - Di bawah NumberOfNetworkCards, pilih jumlah kartu jaringan dalam contoh yang akan ada di grup simpul Anda.
  - Di bawah VpcId, pilih VPC tempat AWS PCS klaster Anda digunakan.
  - Di bawah NodeGroupSubnetId, pilih subnet di cluster Anda VPC tempat instance EFA -enabled akan diluncurkan.
  - Di bawah PlacementGroupName, biarkan bidang kosong untuk membuat grup penempatan cluster baru untuk grup node. Jika Anda memiliki grup penempatan yang ingin Anda gunakan, masukkan namanya di sini.
  - Di bawah ClusterSecurityGroupId, pilih grup keamanan yang Anda gunakan untuk mengizinkan akses ke instance lain di klaster dan ke AWS PCSAPI. Banyak pelanggan memilih grup keamanan default dari cluster merekaVPC.
  - Di bawah SshSecurityGroupId, berikan ID untuk grup keamanan yang Anda gunakan untuk mengizinkan SSH akses masuk ke node di cluster Anda.
  - Untuk SshKeyName, pilih SSH keypair untuk akses ke node di cluster Anda.
  - Untuk LaunchTemplateName, masukkan nama deskriptif untuk template peluncuran seperti*NODEGROUPNAME*`-efa-lt`. Nama harus unik untuk Anda Akun AWS di Wilayah AWS mana Anda akan menggunakan AWS PCS.
- Di bawah Kemampuan
  - Centang kotak untuk saya akui yang AWS CloudFormation mungkin membuat IAM sumber daya.

Pantau status CloudFormation tumpukan. Ketika mencapai `CREATE_COMPLETE` template peluncuran siap untuk digunakan. Gunakan dengan grup node AWS PCS komputasi, seperti dijelaskan di atas dalamMembuat atau memperbarui grup node komputasi.

Menggunakan sistem berkas jaringan dengan AWS PCS

Anda dapat melampirkan volume penyimpanan jaringan ke node yang diluncurkan dalam grup node komputasi AWS Parallel Computing Service (AWS PCS) untuk menyediakan lokasi persisten di mana data dan file dapat ditulis dan diakses. Anda dapat menggunakan volume yang disediakan oleh AWS layanan. Volume termasuk Amazon Elastic File System (AmazonEFS), Amazon FSx for NetApp ONTAP, Amazon FSx for Open ZFS, Amazon FSx for Lustre, dan Amazon File Cache. Anda juga dapat menggunakan volume yang dikelola sendiri, seperti NFS server.

Topik ini mencakup pertimbangan dan contoh penggunaan sistem file jaringan dengan. AWS PCS

Pertimbangan untuk menggunakan sistem file jaringan

Detail implementasi untuk berbagai sistem file berbeda, tetapi ada beberapa pertimbangan umum.

- Perangkat lunak sistem file yang relevan harus diinstal pada instance. Misalnya, untuk menggunakan Amazon FSx untuk Lustre, Lustre paket yang sesuai harus ada. Ini dapat dicapai dengan memasukkannya ke dalam grup node komputasi AMI atau menggunakan skrip yang berjalan saat boot instance.
- Harus ada rute jaringan antara volume penyimpanan bersama dan instance grup node komputasi.
- Aturan grup keamanan pada volume penyimpanan bersama dan instance grup node komputasi harus mengizinkan koneksi ke port yang relevan.
- Anda harus mempertahankan namespace POSIX pengguna dan grup yang konsisten di seluruh sumber daya yang mengakses sistem file. Jika tidak, pekerjaan dan proses interaktif yang berjalan di PCS klaster Anda mungkin mengalami kesalahan izin.
- Pemasangan sistem file dilakukan dengan menggunakan templat EC2 peluncuran. Kesalahan atau batas waktu dalam memasang sistem file jaringan dapat mencegah instance menjadi tersedia untuk menjalankan pekerjaan. Ini, pada gilirannya, dapat menyebabkan biaya yang tidak terduga. Untuk informasi selengkapnya tentang men-debug template peluncuran, lihatMenggunakan template EC2 peluncuran Amazon dengan AWS PCS.

Contoh pemasangan jaringan

Anda dapat membuat sistem file menggunakan AmazonEFS, Amazon FSx untuk Lustre, Amazon FSx for OpenZFS, dan Amazon File Cache. Perluas bagian yang relevan di bawah ini untuk melihat contoh setiap pemasangan jaringan.

Amazon EFS

Pengaturan sistem file

Buat sistem EFS file Amazon. Pastikan ia memiliki target mount di setiap Availability Zone tempat Anda akan meluncurkan instance grup node PCS komputasi. Pastikan juga setiap target mount dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node PCS komputasi. Untuk informasi selengkapnya, lihat Memasang target dan grup keamanan di Panduan Pengguna Amazon Elastic File System.

Luncurkan template

Tambahkan grup keamanan dari pengaturan sistem file Anda ke template peluncuran yang akan Anda gunakan untuk grup node komputasi.

Sertakan data pengguna yang menggunakan `cloud-config` mekanisme untuk memasang sistem EFS file Amazon. Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *`mount-point-directory`*— Jalur pada setiap instance tempat Anda akan memasang Amazon EFS
- *`filesystem-id`*— ID sistem file untuk sistem EFS file

```
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==--
```

Amazon FSx untuk Lustre

Pengaturan sistem file

Buat sistem file FSx for Lustre di VPC mana Anda akan menggunakan. AWS PCS Untuk meminimalkan transfer antar zona, terapkan di subnet di Availability Zone yang sama di mana Anda akan meluncurkan sebagian besar instance grup node PCS komputasi Anda. Pastikan sistem file dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node PCS komputasi. Untuk informasi selengkapnya tentang grup keamanan, lihat Kontrol akses sistem berkas dengan Amazon VPC di Panduan Pengguna Amazon FSx for Lustre.

Luncurkan template

Sertakan data pengguna yang digunakan `cloud-config` untuk me-mount sistem file FSx for Lustre. Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *`mount-point-directory`*— Jalur pada contoh di mana Anda ingin me-mount FSx untuk Lustre
- *`filesystem-id`*— ID sistem file FSx untuk sistem file Lustre
- *`mount-name`*— Nama mount untuk sistem file FSx untuk Lustre
- *`region-code`*— Di Wilayah AWS mana sistem file FSx for Lustre digunakan (harus sama dengan sistem Anda) AWS PCS
- (Opsional) *`latest`* - Versi apa pun yang Lustre didukung oleh FSx untuk Lustre

```
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==
```

Amazon FSx untuk Terbuka ZFS

Pengaturan sistem file

Buat sistem ZFS file FSx for Open di VPC mana Anda akan menggunakan AWS PCS. Untuk meminimalkan transfer antar zona, terapkan di subnet di Availability Zone yang sama di mana Anda akan meluncurkan sebagian besar instance grup node AWS PCS komputasi Anda. Pastikan sistem file dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node AWS PCS komputasi. Untuk informasi selengkapnya tentang grup keamanan, lihat Mengelola akses sistem file dengan Amazon VPC di FSxPanduan ZFS Pengguna Terbuka.

Luncurkan template

Sertakan data pengguna yang digunakan `cloud-config` untuk me-mount volume root untuk sistem ZFS file FSx for Open. Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *`mount-point-directory`*— Jalur pada instance di mana Anda ingin memasang FSx untuk Open ZFS share
- *`filesystem-id`*— ID sistem file untuk sistem ZFS file FSx untuk Open
- *`region-code`*— Wilayah AWS Tempat sistem ZFS file FSx for Open digunakan (harus sama dengan AWS PCS sistem Anda)

```
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==
```

Cache File Amazon

Pengaturan sistem file

Buat Cache File Amazon di VPC tempat yang akan Anda gunakan AWS PCS. Untuk meminimalkan transfer antar zona, pilih subnet di Availability Zone yang sama di mana Anda akan meluncurkan sebagian besar instance grup node PCS komputasi Anda. Pastikan File Cache dikaitkan dengan grup keamanan yang memungkinkan lalu lintas masuk dan keluar pada port 988 antara PCS instance Anda dan File Cache. Untuk informasi selengkapnya tentang grup keamanan, lihat Kontrol akses cache dengan Amazon VPC di Panduan Pengguna Cache File Amazon.

Luncurkan template

Tambahkan grup keamanan dari pengaturan sistem file Anda ke template peluncuran yang akan Anda gunakan untuk grup node komputasi.

Sertakan data pengguna yang digunakan `cloud-config` untuk memasang Cache File Amazon. Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *`mount-point-directory`*— Jalur pada contoh di mana Anda ingin me-mount FSx untuk Lustre
- *`cache-dns-name`*— Nama Sistem Nama Domain (DNS) untuk File Cache
- *`mount-name`*— Nama mount untuk File Cache

```
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==
```

Gambar Mesin Amazon (AMIs) untuk AWS PCS

AWS PCSbekerja dengan AMIs yang Anda berikan, memberikan fleksibilitas besar dalam perangkat lunak dan konfigurasi yang ditemukan pada node di cluster Anda. Jika Anda mencoba AWS PCS, Anda dapat menggunakan sampel yang AMI disediakan oleh dan dikelola oleh AWS. Jika Anda menggunakan AWS PCS dalam produksi, kami sarankan Anda membangun sendiriAMIs. Topik ini

mencakup cara menemukan dan menggunakan sampelAMIs, serta cara membuat dan menggunakan kustomisasi Anda sendiriAMIs.

Topik

Menggunakan sampel Amazon Machine Images (AMIs) dengan AWS PCS

AWSmemberikan sampel AMIs yang dapat Anda gunakan sebagai titik awal untuk bekerja dengan AWS PCS.

> **Important**
>
> Sampel AMIs adalah untuk tujuan demonstrasi dan tidak direkomendasikan untuk beban kerja produksi.

Temukan AWS PCS sampel saat ini AMIs

AWS Management Console

AWSPCSsampel AMIs memiliki konvensi penamaan berikut:

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

Nilai yang diterima

- *OS* – `amzn2`
- *architecture* — `x86_64` atau `arm64`
- *scheduler* – `slurm`
- *scheduler-major-version* – `23.11`

Untuk menemukan AWS PCS sampel AMIs

1. Buka EC2konsol Amazon.

2. Navigasi ke AMIs.
3. Pilih Gambar publik.
4. Di Temukan AMI berdasarkan atribut atau tag, cari AMI menggunakan nama template.

   Contoh

   - Slurm AMI 23.11 mendukung Graviton

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

   - Sampel AMI untuk instance x86

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

   > **Note**
   >
   > Jika ada beberapaAMIs, gunakan AMI dengan stempel waktu terbaru.

5. Gunakan AMI ID saat Anda membuat atau memperbarui grup node komputasi.

AWS CLI

Anda dapat menemukan AWS PCS sampel terbaru AMI dengan perintah yang mengikuti. Ganti *region-code* dengan Wilayah AWS tempat yang Anda gunakan AWS PCS, seperti`us-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
```

Gunakan AMI ID saat Anda membuat atau memperbarui grup node komputasi.

Pelajari lebih lanjut tentang AWS PCS sampel AMIs

Untuk melihat konten, detail konfigurasi untuk rilis AWS PCS sampel saat ini dan sebelumnyaAMIs, lihatCatatan rilis untuk AWS PCS sampel AMIs.

Membangun sendiri yang AMIs kompatibel dengan AWS PCS

Untuk mempelajari cara membangun milik Anda sendiri AMIs yang bekerja dengannya AWS PCS, lihatGambar Mesin Amazon Kustom (AMIs) untuk AWS PCS.

Gambar Mesin Amazon Kustom (AMIs) untuk AWS PCS

AWS PCSdirancang untuk bekerja dengan Amazon Machine Images (AMI) yang Anda bawa ke layanan. Ini AMIs dapat memiliki perangkat lunak dan konfigurasi arbitrer yang diinstal pada mereka, selama mereka memiliki AWS PCS agen dan versi Slurm yang kompatibel diinstal dan dikonfigurasi dengan benar. Anda harus menggunakan installer AWS yang disediakan untuk menginstal AWS PCS perangkat lunak pada kustom Anda. AMI Kami menyarankan Anda menggunakan installer AWS yang disediakan untuk menginstal Slurm pada kustom Anda AMI tetapi Anda dapat menginstal Slurm sendiri jika Anda mau (tidak disarankan).

> **Note**
>
> Jika Anda ingin mencoba AWS PCS tanpa membuat kustomAMI, Anda dapat menggunakan sampel yang AMI disediakan oleh AWS. Untuk informasi selengkapnya, lihat Menggunakan sampel Amazon Machine Images (AMIs) dengan AWS PCS.

Tutorial ini membantu Anda membuat AMI yang dapat digunakan dengan grup node PCS komputasi untuk memberi daya pada beban kerja Anda HPC dan AI/ML.

Topik

- Langkah 1 - Luncurkan instance sementara
- Langkah 2 - Instal AWS PCS agen

- Langkah 3 - Instal Slurm
- Langkah 4 - (Opsional) Instal driver tambahan, perpustakaan, dan perangkat lunak aplikasi
- Langkah 5 - Buat yang AMI kompatibel dengan AWS PCS
- Langkah 6 - Gunakan kustom AMI dengan grup node AWS PCS komputasi
- Langkah 7 - Hentikan instance sementara

Langkah 1 - Luncurkan instance sementara

Luncurkan instance sementara yang dapat Anda gunakan untuk menginstal dan mengkonfigurasi AWS PCS perangkat lunak dan penjadwal Slurm. Anda menggunakan instance ini untuk membuat yang AMI kompatibel dengan AWS PCS.

Untuk meluncurkan instans sementara

1. Buka EC2konsol Amazon.
2. Di panel navigasi, pilih Instans, lalu pilih Launch instance untuk membuka wizard instance peluncuran baru.
3. (Opsional) Di bagian Nama dan tag, berikan nama untuk contoh, sepertiPCS-AMI-instance. Nama ditetapkan ke instans sebagai tanda sumber daya (Name=PCS-AMI-instance).
4. Di bagian Aplikasi dan Gambar OS, pilih AMI untuk salah satu sistem operasi yang didukung.
5. Di bagian Tipe instans, pilih tipe instans yang didukung.
6. Pada bagian Pasangan kunci, pilih pasangan kunci yang akan digunakan untuk instans.
7. Di bagian Pengaturan jaringan:
    - Untuk Firewall (grup keamanan), pilih Pilih grup keamanan yang ada, lalu pilih grup keamanan yang memungkinkan SSH akses masuk ke instans Anda.
8. Di bagian Penyimpanan, konfigurasikan volume sesuai kebutuhan. Pastikan untuk mengonfigurasi ruang yang cukup untuk menginstal aplikasi dan pustaka Anda sendiri.
9. Di panel Ringkasan, pilih Luncurkan instans.

Langkah 2 - Instal AWS PCS agen

Instal agen yang mengonfigurasi instance yang diluncurkan AWS PCS untuk digunakan dengan Slurm.

Untuk menginstal AWS PCS agen

1. Hubungkan ke instans yang Anda luncurkan. Untuk informasi selengkapnya, lihat Connect ke instans Linux Anda.
2. (Opsional) Untuk memastikan bahwa semua paket perangkat lunak Anda mutakhir, lakukan pembaruan perangkat lunak cepat pada instans Anda. Proses ini mungkin memerlukan waktu beberapa menit.

   - Amazon Linux 2, RHEL 9, Rocky Linux 9

     ```
     sudo yum update -y
     ```

   - Ubuntu 22.04

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

3. Boot ulang dan terhubung kembali ke instans Anda.
4. Unduh file instalasi AWS PCS agen. File instalasi dikemas ke dalam file tarball () `.tar.gz` terkompresi. Untuk mengunduh versi stabil terbaru, gunakan perintah berikut. Pengganti *`region`* dengan Wilayah AWS tempat Anda meluncurkan instance sementara Anda, sepertius`-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
   ```

   Anda juga bisa mendapatkan versi terbaru dengan mengganti nomor versi `latest` dengan perintah sebelumnya (misalnya:`aws-pcs-agent-v1-latest.tar.gz`).

   > **Note**
   >
   > Ini mungkin berubah dalam rilis perangkat lunak AWS PCS agen di masa mendatang.

5. (Opsional) Verifikasi keaslian dan integritas tarball AWS PCS perangkat lunak. Kami menyarankan Anda melakukan hal ini untuk memverifikasi identitas penerbit perangkat lunak dan memeriksa apakah file tersebut tidak diubah atau rusak sejak file tersebut diterbitkan.

   a. Unduh GPG kunci publik untuk AWS PCS dan impor ke keyring Anda. Pengganti *`region`* dengan Wilayah AWS tempat Anda meluncurkan instance sementara Anda. Perintah tersebut harus mengembalikan nilai kunci. Catat nilai kunci; Anda menggunakannya di langkah berikutnya.

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

b. Jalankan perintah berikut untuk memverifikasi sidik jari GPG kunci.

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

Perintah harus mengembalikan sidik jari yang identik dengan yang berikut:

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

> **Important**
>
> Jangan jalankan skrip instalasi AWS PCS agen jika sidik jari tidak cocok. Hubungi AWS Support.

c. Unduh file tanda tangan dan verifikasi tanda tangan file tarball AWS PCS perangkat lunak. Ganti *region* dengan Wilayah AWS tempat Anda meluncurkan instance sementara Anda, seperti`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
```

Output harus serupa dengan yang berikut ini:

```
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
```

Jika hasilnya termasuk `Good signature` dan sidik jari cocok dengan sidik jari yang dikembalikan pada langkah sebelumnya, lanjutkan ke langkah berikutnya.

> **Important**
>
> Jangan jalankan skrip instalasi AWS PCS perangkat lunak jika sidik jari tidak cocok. Hubungi AWS Support.

6. Ekstrak file dari file terkompresi `.tar.gz` dan arahkan ke direktori yang diekstrak.

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

7. Instal perangkat AWS PCS lunak.

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

8. Periksa file versi AWS PCS perangkat lunak untuk mengonfirmasi instalasi yang berhasil.

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

Output harus serupa dengan yang berikut ini:

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

Langkah 3 - Instal Slurm

Instal versi Slurm yang kompatibel dengan. AWS PCS

Untuk menginstal Slurm

1. Connect ke instance sementara yang sama di mana Anda menginstal AWS PCS perangkat lunak.
2. Unduh perangkat lunak penginstal Slurm. Penginstal Slurm dikemas ke dalam file tarball () terkompresi. `.tar.gz` Untuk mengunduh versi stabil terbaru, gunakan perintah berikut. Pengganti *`region`* dengan Wilayah AWS contoh sementara Anda, sepertius-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
```

Anda juga bisa mendapatkan versi terbaru dengan mengganti nomor versi `latest` dengan perintah sebelumnya (misalnya:`aws-pcs-slurm-23.11-installer-latest.tar.gz`).

> **Note**
>
> Ini mungkin berubah dalam rilis future dari perangkat lunak installer Slurm.

3. (Opsional) Verifikasi keaslian dan integritas tarball installer Slurm. Kami menyarankan Anda melakukan hal ini untuk memverifikasi identitas penerbit perangkat lunak dan memeriksa apakah file tersebut tidak diubah atau rusak sejak file tersebut diterbitkan.

   a. Unduh GPG kunci publik untuk AWS PCS dan impor ke keyring Anda. Pengganti *`region`* dengan Wilayah AWS tempat Anda meluncurkan instance sementara Anda. Perintah tersebut harus mengembalikan nilai kunci. Catat nilai kunci; Anda menggunakannya di langkah berikutnya.

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

   b. Jalankan perintah berikut untuk memverifikasi sidik jari GPG kunci.

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

   Perintah harus mengembalikan sidik jari yang identik dengan yang berikut:

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

   > **Important**
   >
   > Jangan jalankan skrip instalasi Slurm jika sidik jari tidak cocok. Hubungi AWS Support.

   c. Unduh file tanda tangan dan verifikasi tanda tangan file tarball installer Slurm. Ganti *`region`* dengan Wilayah AWS tempat Anda meluncurkan instance sementara Anda, seperti`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
```

Output harus serupa dengan yang berikut ini:

```
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
```

Jika hasilnya termasuk `Good signature` dan sidik jari cocok dengan sidik jari yang dikembalikan pada langkah sebelumnya, lanjutkan ke langkah berikutnya.

> **Important**
>
> Jangan jalankan skrip instalasi Slurm jika sidik jari tidak cocok. Hubungi AWS Support.

4. Ekstraksi file dari file `.tar.gz` yang dikompresi dan navigasi ke dalam direktori yang diekstraksi.

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

5. Instal slurm. Penginstal mengunduh, mengkompilasi, dan menginstal Slurm dan dependensinya. Dibutuhkan beberapa menit, tergantung pada spesifikasi instance sementara yang Anda pilih.

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

6. Periksa file versi penjadwal untuk mengonfirmasi penginstalan.

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

Output harus serupa dengan yang berikut ini:

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

Langkah 4 - (Opsional) Instal driver tambahan, perpustakaan, dan perangkat lunak aplikasi

Instal driver tambahan, pustaka, dan perangkat lunak aplikasi pada instance sementara. Prosedur instalasi akan bervariasi tergantung pada aplikasi dan pustaka tertentu. Jika Anda belum membuat kustom AMI untuk AWS PCS sebelumnya, kami sarankan Anda terlebih dahulu membangun dan menguji AMI dengan hanya AWS PCS perangkat lunak dan Slurm diinstal, kemudian secara bertahap menambahkan perangkat lunak dan konfigurasi Anda sendiri setelah Anda mengkonfirmasi keberhasilan awal.

Contoh

- Perangkat lunak Adaptor Kain Elastis (EFA). Untuk informasi selengkapnya, lihat Memulai EFA dan MPI untuk HPC beban kerja di Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud.
- Klien Amazon Elastic File System (AmazonEFS). Untuk informasi selengkapnya, lihat Menginstal EFS klien Amazon secara manual di Panduan Pengguna Amazon Elastic File System.
- Klien Lustre, untuk menggunakan Amazon FSx untuk Lustre dan Amazon File Cache. Untuk informasi selengkapnya, lihat Menginstal klien Lustre di FSxfor Lustre User Guide.
- CloudWatch Agen Amazon, untuk menggunakan CloudWatch Log dan Metrik. Untuk informasi selengkapnya, lihat Menginstal CloudWatch agen di Panduan CloudWatch Pengguna Amazon.
- AWS Neuron, untuk menggunakan tipe instance trn* dan inf*. Untuk informasi lebih lanjut, lihat dokumentasi AWS Neuron.
- NVIDIADriver,CUDA, danDCGM, untuk menggunakan tipe instans p* atau g*.

Langkah 5 - Buat yang AMI kompatibel dengan AWS PCS

Setelah Anda menginstal komponen perangkat lunak yang diperlukan, Anda membuat AMI yang dapat Anda gunakan kembali untuk meluncurkan instance di grup node AWS PCS komputasi.

Untuk membuat AMI dari instance sementara Anda

1. Buka EC2konsol Amazon.
2. Di panel navigasi, pilih Instans.
3. Pilih instance sementara yang Anda buat. Pilih Tindakan, Gambar, Buat gambar.
4. Untuk Buat gambar, lakukan hal berikut:
   a. Untuk nama Gambar, masukkan nama deskriptif untuk. AMI
   b. (Opsional) Untuk deskripsi Gambar, masukkan deskripsi singkat tentang tujuanAMI.
   c. Pilih Buat citra.
5. Di panel navigasi, pilih AMIs.
6. Temukan AMI tht yang Anda buat dalam daftar. Tunggu statusnya berubah dari Pending ke Available, lalu gunakan dengan grup node AWS PCS komputasi.

Langkah 6 - Gunakan kustom AMI dengan grup node AWS PCS komputasi

Anda dapat menggunakan kustom Anda AMI dengan grup node AWS PCS komputasi baru atau yang sudah ada.

New compute node group

Untuk menggunakan kustom AMI

1. Buka AWS PCSkonsol.
2. Pada panel navigasi, silakan pilih Klaster.
3. Pilih cluster tempat Anda akan menggunakan kustomAMI, lalu pilih Compute node groups.
4. Buat grup node komputasi baru. Untuk informasi selengkapnya, lihat Membuat grup node komputasi di AWS PCS. Di bawah AMIID, cari nama atau ID kustom yang ingin AMI Anda gunakan. Selesai mengkonfigurasi grup node komputasi, lalu pilih Buat grup node komputasi.
5. (Opsional) Konfirmasikan peluncuran instance AMI pendukung. Luncurkan instance di grup node komputasi. Anda dapat melakukan ini dengan mengonfigurasi grup node komputasi untuk memiliki satu instance statis, atau Anda dapat mengirimkan pekerjaan ke antrian yang menggunakan grup node komputasi.

a. Periksa EC2 konsol Amazon hingga muncul instance yang ditandai dengan ID grup node komputasi baru. Untuk informasi lebih lanjut tentang ini, lihatMenemukan instance grup node komputasi di AWS PCS..

b. Ketika Anda melihat peluncuran instance dan menyelesaikan proses bootstrap, konfirmasikan itu menggunakan yang diharapkanAMI. Untuk melakukan ini, pilih instance, lalu periksa AMIID di bawah Detail. Ini harus cocok dengan yang AMI Anda konfigurasikan dalam pengaturan grup node komputasi.

c. (Opsional) Perbarui konfigurasi penskalaan grup node komputasi ke nilai pilihan Anda.

Existing compute node group

Untuk menggunakan kustom AMI

1. Buka AWS PCSkonsol.
2. Pada panel navigasi, silakan pilih Klaster.
3. Pilih cluster tempat Anda akan menggunakan kustomAMI, lalu pilih Compute node groups.
4. Pilih grup simpul yang ingin Anda konfigurasikan dan pilih Edit. Di bawah AMIID, cari nama atau ID kustom yang ingin AMI Anda gunakan. Selesai mengkonfigurasi grup node komputasi, lalu pilih Perbarui. Instance baru yang diluncurkan di grup node komputasi akan menggunakan ID yang diperbaruiAMI. Instance yang ada akan terus menggunakan yang lama AMI sampai AWS PCS menggantikannya. Untuk informasi selengkapnya, lihat Memperbarui grup node AWS PCS komputasi.
5. (Opsional) Konfirmasikan peluncuran instance AMI pendukung. Luncurkan instance di grup node komputasi. Anda dapat melakukan ini dengan mengonfigurasi grup node komputasi untuk memiliki satu instance statis, atau Anda dapat mengirimkan pekerjaan ke antrian yang menggunakan grup node komputasi.

   a. Periksa EC2 konsol Amazon hingga muncul instance yang ditandai dengan ID grup node komputasi baru. Untuk informasi lebih lanjut tentang ini, lihatMenemukan instance grup node komputasi di AWS PCS..

   b. Ketika Anda melihat peluncuran instance dan menyelesaikan proses bootstrap, konfirmasikan itu menggunakan yang diharapkanAMI. Untuk melakukan ini, pilih instance, lalu periksa AMIID di bawah Detail. Ini harus cocok dengan yang AMI Anda konfigurasikan dalam pengaturan grup node komputasi.

   c. (Opsional) Perbarui konfigurasi penskalaan grup node komputasi ke nilai pilihan Anda.

Langkah 7 - Hentikan instance sementara

Setelah Anda mengonfirmasi bahwa AMI pekerjaan Anda sebagaimana dimaksud AWS PCS, Anda dapat menghentikan instans sementara untuk menghentikan biaya untuk itu.

Untuk mengakhiri instans sementara

1. Buka EC2konsol Amazon.
2. Di panel navigasi, pilih Instans.
3. Pilih instance sementara yang Anda buat dan pilih Actions, Instance state, Terminate instance.
4. Saat diminta untuk mengonfirmasi, pilih Hentikan.

Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS

AWS menyediakan file yang dapat diunduh yang dapat menginstal AWS PCS perangkat lunak pada sebuah instance. AWS juga menyediakan perangkat lunak yang dapat mengunduh, mengkompilasi, dan menginstal versi Slurm yang relevan dan dependensinya. Anda dapat menggunakan petunjuk ini untuk membuat kustom AMIs untuk digunakan dengan AWS PCS atau Anda dapat menggunakan metode Anda sendiri.

Daftar Isi

AWS PCSpenginstal perangkat lunak

Penginstal AWS PCS perangkat lunak mengonfigurasi instance untuk bekerja dengan AWS PCS selama proses bootstrap instance. Anda harus menggunakan installer AWS yang disediakan untuk menginstal AWS PCS perangkat lunak pada kustom Anda. AMI

Pemasang slurm

Penginstal Slurm mengunduh, mengkompilasi, dan menginstal versi Slurm yang relevan dan dependensinya. Anda dapat menggunakan installer Slurm untuk membangun kustom untuk. AMIs AWS PCS Anda juga dapat menggunakan mekanisme Anda sendiri jika mereka konsisten dengan konfigurasi perangkat lunak yang disediakan oleh penginstal Slurm.

Perangkat lunak AWS yang disediakan menginstal yang berikut ini:

- Slurm pada versi mayor dan pemeliharaan yang diminta (saat ini versi 23.11.8) - Lisensi 2 GPL
  - Slurm dibangun dengan `--sysconfdir` set ke `/etc/slurm`
  - Slurm dibangun dengan opsi dan `--enable-pam` `--without-munge`
  - Slurm dibangun dengan opsi `--sharedstatedir=/run/slurm/`
  - Slurm dibangun dengan PMIX dan mendukung JWT
  - Slurm dipasang di `/opt/aws/pcs/schedulers/slurm-23.11`
- Buka PMIX (versi 4.2.6) - Lisensi
  - Open PMIX diinstal sebagai subdirektori dari `/opt/aws/pcs/scheduler/`
- libjwt (versi 1.15.3) - Lisensi -2.0 MPL
  - libjwt diinstal sebagai subdirektori dari `/opt/aws/pcs/scheduler/`

Perangkat lunak AWS yang disediakan mengubah konfigurasi sistem sebagai berikut:

- `systemd`File Slurm yang dibuat oleh build disalin `/etc/systemd/system/` dengan nama file. `slurmd-23.11.service`
- Jika tidak ada, pengguna Slurm dan grup (`slurm:slurm`) dibuat dengan`UID/GID`dari. `401`
- Di Amazon Linux 2 dan Rocky Linux 9 instalasi menambahkan EPEL repositori untuk menginstal perangkat lunak yang diperlukan untuk membangun Slurm atau dependensinya.
- Pada RHEL9 instalasi akan mengaktifkan `codeready-builder-for-rhel-9-rhui-rpms` dan `epel-release-latest-9` dari `fedoraproject` untuk menginstal perangkat lunak yang diperlukan untuk membangun Slurm atau dependensinya.

Sistem operasi yang didukung

Perangkat AWS PCS lunak dan installer Slurm mendukung sistem operasi berikut:

- Amazon Linux 2
- RedHat Perusahaan Linux 9
- Berbatu Linux 9
- Ubuntu 22.04

> **Note**
>
> AWS Deep Learning AMIs (DLAMI) versi berbasis Amazon Linux 2 dan Ubuntu 22.04 harus kompatibel dengan AWS PCS perangkat lunak dan installer Slurm. Untuk informasi selengkapnya, lihat Memilih Panduan AWS Deep Learning AMIs Pengembang Anda DLAMI.

Tipe instans yang didukung

AWS PCSperangkat lunak dan penginstal Slurm mendukung jenis instans x86_64 atau arm64 apa pun yang dapat menjalankan salah satu sistem operasi yang didukung.

Versi Slurm yang didukung

Versi utama Slurm berikut didukung:

- Buburan 23.11

Verifikasi penginstal menggunakan checksum

Anda dapat menggunakan SHA256 checksum untuk memverifikasi file tarball installer (.tar.gz). Kami menyarankan Anda melakukan hal ini untuk memverifikasi identitas penerbit perangkat lunak dan memeriksa apakah aplikasi tersebut belum diubah atau rusak sejak file tersebut diterbitkan.

Untuk memverifikasi tarball

Gunakan utilitas sha256sum untuk SHA256 checksum dan tentukan nama file tarball. Anda harus menjalankan perintah dari direktori tempat Anda menyimpan file tarball.

- SHA256

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

Perintah harus mengembalikan nilai checksum dalam format berikut.

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

Bandingkan nilai checksum yang dikembalikan oleh perintah dengan nilai checksum yang disediakan dalam tabel berikut. Jika checksum cocok, maka aman untuk menjalankan skrip instalasi.

> **Important**
>
> Jika checksum tidak cocok, jangan jalankan skrip instalasi. Hubungi AWS Support.

Misalnya, perintah berikut menghasilkan SHA256 checksum untuk tarball Slurm 23.11.9.

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

Contoh output:

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

Tabel berikut mencantumkan checksum untuk versi terbaru dari installer. Ganti *us-east-1* dengan Wilayah AWS tempat yang Anda gunakan AWS PCS.

Penginstal	Unduh URL	SHA256checksum
Buburan 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 PCSagen 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`

Versi slurm di AWS PCS

SchedMD terus meningkatkan Slurm dengan kemampuan baru, optimasi, dan patch keamanan. SchedMD merilis versi utama baru secara berkala dan berencana untuk mendukung hingga 3 versi pada waktu tertentu. AWS PCSawalnya mendukung Slurm 23.11. Anda dapat memutakhirkan versi utama Slurm Anda setelah versi baru dirilis. AWS PCSdirancang untuk memperbarui pengontrol Slurm secara otomatis dengan versi patch.

Ketika SchedMD mengakhiri dukungan untuk versi utama tertentu, AWS PCS juga mengakhiri dukungan untuk versi utama itu. AWS PCSmengirimkan pemberitahuan terlebih dahulu jika versi utama Slurm mendekati akhir masa pakainya, untuk membantu pelanggan mengetahui kapan harus meningkatkan cluster mereka ke versi yang didukung yang lebih baru.

Kami menyarankan Anda menggunakan versi Slurm terbaru yang didukung untuk menyebarkan klaster Anda, untuk mengakses kemajuan dan peningkatan terbaru.

Pertanyaan yang sering diajukan tentang versi Slurm

Berapa lama AWS PCS mendukung versi Slurm?

AWS PCSmengikuti siklus dukungan SchedMD untuk versi utama. AWS PCSmendukung hingga 3 versi utama pada waktu tertentu. Setelah SchedMD merilis versi mayor baru, AWS PCS pensiun versi tertua yang didukung. AWS PCSmerilis versi utama baru Slurm sesegera mungkin, tetapi mungkin ada penundaan antara rilis schedMD dan ketersediaannya di. AWS PCS

Kapan AWS PCS memberi tahu saya tentang End of Support Life (EOSL) untuk versi Slurm?

AWS PCSmemberi tahu Anda beberapa kali, dalam irama yang telah ditentukan sebelumnya, sebelum tanggal. EOSL

Apa yang harus saya lakukan ketika versi Slurm mendekat? EOSL

Anda harus memperbarui versi Slurm Anda sebelumnya EOSL untuk membantu menjaga lingkungan yang aman dan didukung.

Bagaimana cara memperbarui cluster saya untuk menggunakan versi utama Slurm yang baru?

Untuk memperbarui versi Slurm, Anda harus membuat cluster baru. Anda juga harus meningkatkan ke AWS PCS perangkat lunak yang setara di Anda AMI dan menggunakannya untuk membuat grup node komputasi untuk cluster baru Anda.

Bagaimana cluster saya mendapatkan rilis versi patch Slurm baru?

AWS PCSdirancang untuk secara otomatis menerapkan tambalan untuk mengatasi Slurm Common Vulnerabilities and Exposures (). CVEs AWS PCSmenerapkan tambalan ke pengontrol cluster yang berjalan di akun milik layanan internal. Anda harus menggunakan AWS PCS API tindakan AWS Management Console atau untuk menginstal tambalan pada EC2 instance di Anda. Akun AWS

Bagaimana jika saya tidak memperbarui Slurm berdasarkan tanggal? EOSL

AWS PCSdirancang untuk menghentikan cluster yang memiliki versi Slurm yang tidak didukung. Anda harus memperbarui versi utama Slurm dari pengontrol cluster dan AWS PCS perangkat lunak yang diinstal pada grup node komputasi.

Berapa banyak versi Slurm yang mendukung? AWS PCS

AWS PCSmendukung hingga 3 versi Slurm utama pada waktu tertentu, termasuk versi utama saat ini dan 2 sebelumnya.

Pembaruan versi Slurm apa yang harus saya terapkan?

Kami sangat menyarankan Anda menggunakan versi utama yang sama di semua komponen di cluster Anda dan menginstal tambalan terbaru segera setelah dirilis. AMIsUntuk grup node komputasi Anda harus menggunakan versi perangkat lunak Slurm yang kompatibel dengan versi Slurm dari pengontrol cluster. Versi utama Slurm di Anda AMIs harus berada dalam 2 versi versi utama Slurm pada pengontrol cluster. Versi Slurm yang diinstal di AMI dan pada EC2 instance yang sedang berjalan di cluster tidak bisa lebih baru dari versi Slurm pada pengontrol cluster. Untuk mempertahankan dukungan untuk klaster Anda, Anda AMIs harus menggunakan versi AWS PCS perangkat lunak yang didukung.

Bagaimana jika saya memperbarui versi utama Slurm tetapi menggunakan perangkat lunak Slurm yang lebih lama di grup node komputasi sayaAMI?

Anda harus memperbarui AWS PCS perangkat lunak ke versi yang sama untuk menggunakan fungsionalitas Slurm baru. Untuk AWS PCS dukungan penuh, semua komponen Slurm harus menggunakan versi yang didukung. Ringkasnya:

- Kami dapat memberikan dukungan penuh ketika pengontrol cluster dan semua komponen (AWS PCSpaket) di Anda Akun AWS berdua menggunakan versi yang didukung.
- AWS PCSdirancang untuk menghentikan cluster jika versi Slurm dari pengontrolnya mencapai. EOSL

- Jika versi Slurm komponen dalam Akun AWS jangkauan AndaEOSL, klaster Anda tidak akan didukung.

Dalam urutan apa saya harus memperbarui komponen di Cluster saya?

Anda harus memperbarui versi Slurm dari pengontrol cluster Anda sebelum Anda menggunakan versi Slurm AMI dengan yang lebih baru. Anda memperbarui grup node komputasi untuk menggunakan. AMI AWS PCSmenggunakan AMI untuk meluncurkan EC2 instance baru di grup node komputasi. AWS PCStidak memperbarui EC2 instance yang ada yang menjalankan pekerjaan; AWS PCS dirancang untuk menghentikan instance tersebut setelah pekerjaan mereka selesai.

Apakah AWS PCS menawarkan dukungan tambahan untuk versi Slurm?

Tidak. Kami akan mengkomunikasikan informasi terperinci tentang opsi dukungan yang diperluas, termasuk biaya tambahan dan cakupan dukungan khusus yang disediakan.

Keamanan dalam Layanan Komputasi AWS Paralel

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Layanan Komputasi AWS Paralel, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS PCS. Topik berikut menunjukkan cara mengonfigurasi AWS PCS untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS PCS sumber daya Anda.

Topik

Perlindungan data dalam Layanan Komputasi AWS Paralel

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Layanan Komputasi AWS Paralel. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat Privasi Data FAQ. Untuk informasi tentang perlindungan data di Eropa, lihat Model Tanggung Jawab AWS Bersama dan posting GDPR blog di Blog AWS Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management ()IAM. Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- GunakanSSL/TLSuntuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir. API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat Federal Information Processing Standard (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk ketika Anda bekerja dengan AWS PCS atau lainnya Layanan AWS menggunakan konsol,API, AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Jika Anda memberikan URL ke server eksternal, kami sangat menyarankan agar

Anda tidak menyertakan informasi kredensional dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi diam

Enkripsi diaktifkan secara default untuk data saat istirahat saat Anda membuat cluster AWS Parallel Computing Service (AWS PCS) dengan AWS Management Console AWS CLI, AWS PCSAPI, atau AWS SDKs. AWS PCSmenggunakan KMSkunci yang AWS dimiliki untuk mengenkripsi data saat istirahat. Untuk informasi selengkapnya, lihat Kunci dan AWS kunci pelanggan di Panduan AWS KMS Pengembang. Rahasia cluster disimpan AWS Secrets Manager dan dienkripsi dengan kunci KMS terkelola Secrets Manager. Untuk informasi selengkapnya, lihat Bekerja dengan rahasia cluster di AWS PCS.

Dalam sebuah AWS PCS cluster, data berikut diam:

- Status penjadwal - Ini mencakup data tentang pekerjaan yang sedang berjalan dan node yang disediakan di cluster. Ini adalah data yang Slurm bertahan dalam yang `StateSaveLocation` ditentukan dalam Anda. `slurm.conf` Untuk informasi lebih lanjut, lihat deskripsi StateSaveLocationdalam dokumentasi Slurm. AWS PCSmenghapus data pekerjaan setelah pekerjaan selesai.
- Rahasia autentikasi penjadwal — AWS PCS menggunakannya untuk mengautentikasi semua komunikasi penjadwal di cluster.

Untuk informasi status scheduler, AWS PCS secara otomatis mengenkripsi data dan metadata sebelum menuliskannya ke sistem file. Sistem file terenkripsi menggunakan algoritma enkripsi AES -256 standar industri untuk data saat istirahat.

Enkripsi bergerak

Koneksi Anda ke TLS enkripsi AWS PCS API penggunaan dengan proses penandatanganan Signature Version 4, terlepas dari apakah Anda menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDKs. Untuk informasi selengkapnya, lihat Menandatangani AWS API permintaan di Panduan AWS Identity and Access Management Pengguna. AWS mengelola kontrol akses melalui API dengan IAM kebijakan untuk kredenal keamanan yang Anda gunakan untuk terhubung.

AWS PCSdigunakan TLS untuk terhubung ke AWS layanan lain.

Dalam cluster Slurm, scheduler dikonfigurasi dengan plug-in otentikasi yang menyediakan `auth/slurm` otentikasi untuk semua komunikasi scheduler. Slurm tidak menyediakan enkripsi pada tingkat aplikasi untuk komunikasinya, semua data yang mengalir di seluruh instance cluster tetap lokal EC2 VPC dan oleh karena itu tunduk pada VPC enkripsi jika instance tersebut mendukung enkripsi dalam perjalanan. Untuk informasi selengkapnya, lihat Enkripsi dalam perjalanan di Panduan Pengguna Amazon Elastic Compute Cloud. Komunikasi dienkripsi antara pengontrol (disediakan dalam akun layanan) node cluster di akun Anda.

Manajemen kunci

AWS PCSmenggunakan KMSkunci yang AWS dimiliki untuk mengenkripsi data. Untuk informasi selengkapnya, lihat Kunci dan AWS kunci pelanggan di Panduan AWS KMS Pengembang. Rahasia cluster disimpan AWS Secrets Manager dan dienkripsi dengan kunci KMS terkelola Secrets Manager. Untuk informasi selengkapnya, lihat Bekerja dengan rahasia cluster di AWS PCS.

Privasi lalu lintas antar jaringan

AWS PCSmenghitung sumber daya untuk klaster berada dalam 1 VPC di akun pelanggan. Oleh karena itu, semua lalu lintas AWS PCS layanan internal dalam sebuah cluster tetap berada dalam AWS jaringan dan tidak melakukan perjalanan melalui internet. Komunikasi antara pengguna dan AWS PCS node dapat melakukan perjalanan melalui internet dan kami sarankan menggunakan SSH atau Systems Manager untuk terhubung ke node. Untuk informasi lebih lanjut, lihat Apa itu AWS Systems Manager? dalam AWS Systems Manager User Guide.

Anda juga dapat menggunakan penawaran berikut untuk menghubungkan jaringan lokal Anda ke: AWS

- AWS Site-to-Site VPN. Untuk informasi lebih lanjut, lihat Apa itu AWS Site-to-Site VPN? dalam AWS Site-to-Site VPN User Guide.
- Sebuah AWS Direct Connect. Untuk informasi lebih lanjut, lihat Apa itu AWS Direct Connect? dalam AWS Direct Connect User Guide.

Anda mengakses AWS PCS API untuk melakukan tugas-tugas administratif untuk layanan. Anda dan pengguna Anda mengakses port endpoint Slurm untuk berinteraksi dengan penjadwal secara langsung.

Mengenkripsi lalu lintas API

Untuk mengakses AWS PCSAPI, klien harus mendukung Transport Layer Security (TLS) 1.2 atau yang lebih baru. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3. Klien juga harus mendukung cipher suite dengan Perfect Forward Secrecy (PFS), seperti Ephemeral Diffie-Hellman () atau Elliptic Curve Diffie-Hellman Ephemeral (). DHE ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini. Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Anda juga dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredenal keamanan sementara untuk menandatangani permintaan.

Mengenkripsi lalu lintas data

Enkripsi data dalam perjalanan diaktifkan dari EC2 instance yang didukung yang mengakses titik akhir penjadwal dan antar ComputeNodeGroup instance dari dalam. AWS Cloud Untuk informasi selengkapnya, lihat Enkripsi bergerak.

Akses Layanan Komputasi AWS Paralel menggunakan titik akhir antarmuka ()AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara Anda VPC dan AWS Parallel Computing Service (AWS PCS). Anda dapat mengakses AWS PCS seolah-olah itu ada di AndaVPC, tanpa menggunakan gateway internet, NAT perangkat, VPN koneksi, atau AWS Direct Connect koneksi. Contoh di Anda VPC tidak memerlukan alamat IP publik untuk mengakses AWS PCS.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditakdirkan. AWS PCS

Untuk informasi selengkapnya, lihat Akses Layanan AWS melalui AWS PrivateLink di AWS PrivateLink Panduan.

Pertimbangan untuk AWS PCS

Sebelum menyiapkan titik akhir antarmuka AWS PCS, tinjau Akses AWS layanan menggunakan VPC titik akhir antarmuka di AWS PrivateLink Panduan.

AWS PCSmendukung membuat panggilan ke semua API tindakannya melalui titik akhir antarmuka.

Jika Anda VPC tidak memiliki akses internet langsung, Anda harus mengonfigurasi VPC titik akhir untuk mengaktifkan instance grup node komputasi Anda untuk memanggil tindakan. AWS PCS `RegisterComputeNodeGroupInstance`API

Buat titik akhir antarmuka untuk AWS PCS

Anda dapat membuat titik akhir antarmuka untuk AWS PCS menggunakan VPC konsol Amazon atau AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di AWS PrivateLink Panduan.

Buat titik akhir antarmuka untuk AWS PCS menggunakan nama layanan berikut:

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

Ganti *region* dengan ID Wilayah AWS untuk membuat titik akhir di, sepertius-east-1.

Jika Anda mengaktifkan private DNS untuk titik akhir antarmuka, Anda dapat membuat API permintaan untuk AWS PCS menggunakan DNS nama Regional default. Misalnya, `pcs.us-east-1.amazonaws.com`.

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah IAM sumber daya yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh AWS PCS melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan AWS PCS dari AndaVPC, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, IAM pengguna, dan IAM peran).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan menggunakan kebijakan titik akhir di Panduan AWS PrivateLink .

Contoh: kebijakan VPC endpoint untuk tindakan AWS PCS

Berikut ini adalah contoh kebijakan endpoint kustom. Saat Anda melampirkan kebijakan ini ke titik akhir antarmuka Anda, kebijakan ini akan memberikan akses ke AWS PCS tindakan yang tercantum untuk semua prinsipal ke klaster dengan yang ditentukan *cluster-id*. Ganti *region* dengan ID Wilayah AWS dari cluster, seperti`us-east-1`. Ganti *account-id* dengan Akun AWS jumlah cluster.

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

Identity and Access Management untuk Layanan Komputasi AWS Paralel

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. IAMadministrator mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS PCS IAMadalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

- AWS kebijakan terkelola untuk Layanan Komputasi AWS Paralel
- Peran yang terhubung dengan layanan untuk AWS PCS
- Peran Amazon EC2 Spot untuk AWS PCS
- Izin minimum untuk AWS PCS
- IAMprofil instance untuk Layanan Komputasi AWS Paralel
- Pemecahan Masalah Identitas dan akses Layanan Komputasi AWS Paralel

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan AWS PCS.

Pengguna layanan — Jika Anda menggunakan AWS PCS layanan untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS PCS fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur AWS PCS, lihatPemecahan Masalah Identitas dan akses Layanan Komputasi AWS Paralel.

Administrator layanan — Jika Anda bertanggung jawab atas AWS PCS sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS PCS. Tugas Anda adalah menentukan AWS PCS fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke IAM administrator Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasarIAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakannya IAM AWS PCS, lihatBagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM.

IAMadministrator - Jika Anda seorang IAM administrator, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses AWS PCS. Untuk melihat contoh kebijakan AWS PCS berbasis identitas yang dapat Anda gunakan, lihat. IAM Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai IAM pengguna, atau dengan mengambil peranIAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (Pusat IAM Identitas), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan IAM peran. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang menggunakan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani AWS API permintaan di Panduan IAM Pengguna.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat Autentikasi multi-faktor di Panduan AWS IAM Identity Center Pengguna dan Menggunakan otentikasi multi-faktor (MFA) AWS di Panduan Pengguna. IAM

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensi pengguna root di IAMPanduan Pengguna.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat IAM Identitas, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat IAM Identitas, lihat Apa itu Pusat IAM Identitas? dalam AWS IAM Identity Center User Guide.

Pengguna dan grup IAM

IAMPengguna adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensi sementara daripada membuat IAM pengguna yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensil jangka panjang dengan IAM pengguna, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensi jangka panjang di IAMPanduan Pengguna.

IAMGrup adalah identitas yang menentukan kumpulan IAM pengguna. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup bernama IAMAdminsdan memberikan izin grup tersebut untuk mengelola sumber dayaIAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari lebih lanjut, lihat Kapan membuat IAM pengguna (bukan peran) di Panduan IAM Pengguna.

IAMperan

IAMPeran adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Ini mirip dengan IAM pengguna, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil IAM peran sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil AWS CLI atau AWS API operasi atau dengan menggunakan kustomURL. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat Menggunakan IAM peran di Panduan IAM Pengguna.

IAMperan dengan kredensi sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga di Panduan IAM Pengguna. Jika Anda menggunakan Pusat IAM Identitas, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, Pusat IAM Identitas menghubungkan izin yang disetel ke peran. IAM Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin IAM pengguna sementara — IAM Pengguna atau peran dapat mengambil IAM peran untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun — Anda dapat menggunakan IAM peran untuk memungkinkan seseorang (prinsipal tepercaya) di akun lain mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM Panduan Pengguna. IAM
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FASmenggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan

hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat Meneruskan sesi akses.

- Peran layanan — Peran layanan adalah IAMperan yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam IAMPanduan Pengguna.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan IAM peran untuk mengelola kredenal sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS API meminta. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan IAM peran untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di IAMPanduan Pengguna.

Untuk mempelajari apakah akan menggunakan IAM peran atau IAM pengguna, lihat Kapan membuat IAM peran (bukan pengguna) di Panduan IAM Pengguna.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai JSON dokumen. Untuk informasi selengkapnya tentang struktur dan isi dokumen JSON kebijakan, lihat Ringkasan JSON kebijakan di Panduan IAM Pengguna.

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

IAMkebijakan menentukan izin untuk tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasi. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan itu bisa mendapatkan informasi peran dari AWS Management Console, AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna. IAM

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan sebaris, lihat Memilih antara kebijakan terkelola dan kebijakan sebaris di IAMPanduan Pengguna.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLsmirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnyaACLs, lihat Ikhtisar daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batas izin — Batas izin adalah fitur lanjutan tempat Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas (pengguna atau peran). IAM IAM Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batas izin, lihat Batas izin untuk IAM entitas di IAMPanduan Pengguna.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah JSON kebijakan yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCPMembatasi izin untuk entitas di akun anggota, termasuk masing-masing Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations danSCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan secara tegas dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat Kebijakan sesi di Panduan IAM Pengguna.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan IAM Pengguna.

Bagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses AWS PCS, pelajari IAM fitur apa yang tersedia untuk digunakan AWS PCS.

IAMfitur yang dapat Anda gunakan dengan Layanan Komputasi AWS Paralel

IAMfitur	AWS PCSdukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC(tag dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin prinsipal	Ya
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS PCS dan AWS layanan lain bekerja dengan sebagian besar IAM fitur, lihat AWS layanan yang berfungsi IAM di Panduan IAM Pengguna.

Kebijakan berbasis identitas untuk AWS PCS

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan JSON izin yang dapat Anda lampirkan ke identitas, seperti pengguna, grup IAM pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat IAM kebijakan di Panduan Pengguna. IAM

Dengan kebijakan IAM berbasis identitas, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak serta kondisi di mana tindakan diizinkan atau ditolak. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam JSON kebijakan, lihat referensi elemen IAM JSON kebijakan di Panduan IAM Pengguna.

Contoh kebijakan berbasis identitas untuk AWS PCS

Untuk melihat contoh kebijakan AWS PCS berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

Kebijakan berbasis sumber daya dalam AWS PCS

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen JSON kebijakan yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan IAM peran dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau IAM entitas di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan.

Ketika prinsipal dan sumber daya berbeda Akun AWS, IAM administrator di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun IAM di Panduan IAM Pengguna.

Tindakan kebijakan untuk AWS PCS

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

`Action`Elemen JSON kebijakan menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan AWS API operasi terkait. Ada beberapa pengecualian, seperti tindakan khusus izin yang tidak memiliki operasi yang cocok. API Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar AWS PCS tindakan, lihat Tindakan yang Ditentukan oleh Layanan Komputasi AWS Paralel di Referensi Otorisasi Layanan.

Tindakan kebijakan AWS PCS menggunakan awalan berikut sebelum tindakan:

```
pcs
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan AWS PCS berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

Sumber daya kebijakan untuk AWS PCS

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Resource` JSON kebijakan menentukan objek atau objek yang tindakan tersebut berlaku. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis AWS PCS sumber daya dan jenisnyaARNs, lihat Sumber Daya yang Ditentukan oleh Layanan Komputasi AWS Paralel di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang Ditentukan oleh Layanan Komputasi AWS Paralel.

Untuk melihat contoh kebijakan AWS PCS berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

Kunci kondisi kebijakan untuk AWS PCS

Mendukung kunci kondisi kebijakan khusus layanan: Ya

Administrator dapat menggunakan AWS JSON kebijakan untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin IAM pengguna untuk mengakses sumber daya hanya jika ditandai dengan nama IAM pengguna mereka. Untuk informasi selengkapnya, lihat elemen IAM kebijakan: variabel dan tag di Panduan IAM Pengguna.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan IAM Pengguna.

Untuk melihat daftar kunci kondisi, lihat Kunci AWS PCS Kondisi untuk Layanan Komputasi AWS Paralel di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang Ditentukan oleh Layanan Komputasi AWS Paralel.

Untuk melihat contoh kebijakan AWS PCS berbasis identitas, lihat. Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

ACLsdi AWS PCS

MendukungACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLsmirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan. JSON

ABACdengan AWS PCS

Mendukung ABAC (tag dalam kebijakan): Ya

Attribute-based access control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke IAM entitas (pengguna

atau peran) dan ke banyak AWS sumber daya. Menandai entitas dan sumber daya adalah langkah pertama dari. ABAC Kemudian Anda merancang ABAC kebijakan untuk mengizinkan operasi ketika tag prinsipal cocok dengan tag pada sumber daya yang mereka coba akses.

ABACmembantu dalam lingkungan yang berkembang pesat dan membantu dengan situasi di mana manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi lebih lanjut tentangABAC, lihat Apa ituABAC? dalam IAMUser Guide. Untuk melihat tutorial dengan langkah-langkah penyiapanABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) di IAMPanduan Pengguna.

Menggunakan kredensi sementara dengan AWS PCS

Mendukung kredensi sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensil sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang berfungsi IAM di IAMPanduan Pengguna.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan link sign-on (SSO) tunggal perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat Beralih ke peran (konsol) di Panduan IAM Pengguna.

Anda dapat secara manual membuat kredensi sementara menggunakan atau. AWS CLI AWS API Anda kemudian dapat menggunakan kredensil sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensi keamanan sementara di. IAM

Izin utama lintas layanan untuk AWS PCS

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan IAM pengguna atau peran untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FASmenggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. FASPermintaan hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat FAS permintaan, lihat Meneruskan sesi akses.

Peran layanan untuk AWS PCS

Mendukung peran layanan: Tidak

Peran layanan adalah IAMperan yang diasumsikan layanan untuk melakukan tindakan atas nama Anda. IAMAdministrator dapat membuat, memodifikasi, dan menghapus peran layanan dari dalamIAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam IAMPanduan Pengguna.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak AWS PCS fungsionalitas. Edit peran layanan hanya jika AWS PCS memberikan panduan untuk melakukannya.

Peran terkait layanan untuk AWS PCS

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. IAMAdministrator dapat melihat, tetapi tidak mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan, lihat AWS layanan yang berfungsi dengannya. IAM Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi AWS PCS sumber daya. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, IAM administrator dapat membuat IAM kebijakan. Administrator kemudian dapat menambahkan IAM kebijakan ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan IAM berbasis identitas menggunakan contoh dokumen kebijakan ini, lihat Membuat JSON IAM kebijakan di Panduan Pengguna. IAM

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AWS PCS, termasuk format ARNs untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk Layanan Komputasi AWS Paralel di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol AWS PCS
- Mengizinkan pengguna melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS PCS sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan AWSAWS terkelola atau kebijakan terkelola untuk fungsi pekerjaan di Panduan IAM Pengguna.
- Menerapkan izin hak istimewa paling sedikit — Saat Anda menetapkan izin dengan IAM kebijakan, berikan hanya izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan

mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat Kebijakan dan izin IAM di IAM Panduan Pengguna.

- Gunakan ketentuan dalam IAM kebijakan untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakanSSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat elemen IAM JSON kebijakan: Kondisi dalam Panduan IAM Pengguna.
- Gunakan IAM Access Analyzer untuk memvalidasi IAM kebijakan Anda guna memastikan izin yang aman dan fungsional — IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa IAM kebijakan () JSON dan praktik terbaik. IAM IAMAccess Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan IAM Access Analyzer di IAMPanduan Pengguna.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan IAM pengguna atau pengguna root di Anda Akun AWS, aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA kapan API operasi dipanggil, tambahkan MFA kondisi ke kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi API akses MFA yang dilindungi di IAMPanduan Pengguna.

Untuk informasi selengkapnya tentang praktik terbaik diIAM, lihat Praktik terbaik keamanan IAM di Panduan IAM Pengguna.

Menggunakan konsol AWS PCS

Untuk mengakses konsol Layanan Komputasi AWS Paralel, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang AWS PCS sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau. AWS API Sebagai gantinya, izinkan akses hanya ke tindakan yang cocok dengan API operasi yang mereka coba lakukan.

Untuk informasi selengkapnya tentang izin minimum yang diperlukan untuk menggunakan AWS PCS konsol, lihatIzin minimum untuk AWS PCS.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan IAM pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau secara terprogram menggunakan atau. AWS CLI 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": "*"
        }
    ]
}
```

AWS kebijakan terkelola untuk Layanan Komputasi AWS Paralel

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau API operasi baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat kebijakan AWS terkelola di Panduan IAM Pengguna.

AWS kebijakan terkelola: AWSPCSServiceRolePolicy

Anda tidak dapat melampirkan AWSPCSServiceRolePolicy ke IAM entitas Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan AWS PCS untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Peran yang terhubung dengan layanan untuk AWS PCS.

Detail izin

Kebijakan ini mencakup izin berikut.

- `ec2`— Memungkinkan AWS PCS untuk membuat dan mengelola EC2 sumber daya Amazon.
- `iam`— Memungkinkan AWS PCS untuk membuat peran terkait layanan untuk EC2 armada Amazon dan meneruskan peran tersebut ke Amazon. EC2

- `cloudwatch`— Memungkinkan AWS PCS untuk mempublikasikan metrik layanan ke Amazon CloudWatch.
- `secretsmanager`— Memungkinkan AWS PCS untuk mengelola rahasia untuk sumber daya AWS PCS cluster.

```
{
    "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 PCSpembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola AWS PCS sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan RSS feed di halaman Riwayat AWS PCS dokumen.

Perubahan	Deskripsi	Tanggal
AWS PCS mulai melacak perubahan	AWS PCSmulai melacak perubahan untuk kebijakan yang AWS dikelola.	Agustus 28, 2024

Peran yang terhubung dengan layanan untuk AWS PCS

AWS Layanan Komputasi Paralel menggunakan AWS Identity and Access Management (IAM) peran terkait layanan. Peran terkait layanan adalah jenis peran unik yang ditautkan langsung ke. IAM AWS PCS Peran terkait layanan telah ditentukan sebelumnya oleh AWS PCS dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan AWS PCS lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS PCSmendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya AWS PCS dapat mengambil perannya. Izin yang ditetapkan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas lain mana pun. IAM

Anda dapat menghapus peran tertaut layanan hanya setelah terlebih dahulu menghapus sumber dayanya yang terkait. Ini melindungi AWS PCS sumber daya Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat AWS layanan yang bekerja dengan IAM dan cari layanan yang memiliki Ya di kolom Peran terkait layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk AWS PCS

AWS PCSmenggunakan peran terkait layanan bernama AWSServiceRoleForPCS— Izinkan AWS PCS untuk mengelola sumber daya AmazonEC2.

Peran AWSServiceRoleForPCS terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `pcs.amazonaws.com`

Kebijakan izin peran bernama AWSPCSServiceRolePolicymemungkinkan AWS PCS untuk menyelesaikan tindakan pada sumber daya tertentu.

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran terkait layanan di Panduan Pengguna. IAM

Membuat peran terkait layanan untuk AWS PCS

Anda tidak perlu membuat peran terkait layanan secara manual. AWS PCSmembuat peran terkait layanan untuk Anda saat membuat klaster.

Mengedit peran terkait layanan untuk AWS PCS

AWS PCStidak memungkinkan Anda untuk mengedit peran AWSServiceRoleForPCS terkait layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit deskripsi peran menggunakanIAM. Untuk informasi selengkapnya, lihat Mengedit peran terkait layanan di IAMPanduan Pengguna.

Menghapus peran terkait layanan untuk AWS PCS

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran yang terhubung dengan layanan sebelum menghapusnya secara manual.

> **Note**
>
> Jika AWS PCS layanan menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus AWS PCS sumber daya yang digunakan oleh AWSServiceRoleForPCS

Anda harus menghapus semua klaster untuk menghapus peran AWSServiceRoleForPCS terkait layanan. Untuk informasi selengkapnya, lihat Menghapus klaster.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan IAM konsol, AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForPCS terkait layanan. Untuk informasi selengkapnya, lihat Menghapus peran terkait layanan di Panduan Pengguna. IAM

Wilayah yang didukung untuk peran yang terhubung dengan layanan AWS PCS

AWS PCSmendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat AWS Wilayah dan titik akhir.

Peran Amazon EC2 Spot untuk AWS PCS

Jika Anda ingin membuat grup node AWS PCS komputasi yang menggunakan Spot sebagai opsi pembeliannya, Anda juga harus memiliki peran AWSServiceRoleForEC2Spotterkait layanan di situs Anda. Akun AWS Anda dapat menggunakan AWS CLI perintah berikut untuk membuat peran. Untuk informasi selengkapnya, lihat Membuat peran terkait layanan dan Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna.AWS Identity and Access Management

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

> **Note**
>
> Anda menerima kesalahan berikut jika Anda Akun AWS sudah memiliki `AWSServiceRoleForEC2Spot` IAM peran.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
> operation: Service role name AWSServiceRoleForEC2Spot has been taken in this
> account, please try a different suffix.
> ```

Izin minimum untuk AWS PCS

Bagian ini menjelaskan IAM izin minimum yang diperlukan untuk IAM identitas (pengguna, grup, atau peran) untuk menggunakan layanan.

Daftar Isi

Izin minimum untuk menggunakan tindakan API

APIaksi	Izin minimum	Izin tambahan untuk konsol
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`	

APIaksi	Izin minimum	Izin tambahan untuk konsol
CreateComputeNodeGroup	`ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags, iam:PassRole, iam:GetInstanceProfile, pcs:CreateComputeNodeGroup`	`iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster`
ListComputerNodeGroups	`pcs:ListComputeNodeGroups`	`pcs:GetCluster`
GetComputeNodeGroup	`pcs:GetComputeNodeGroup`	`ec2:DescribeSubnets`

APIaksi	Izin minimum	Izin tambahan untuk konsol
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`	
CreateQueue	`pcs:CreateQueue`	`pcs:ListComputeNodeGroups, pcs:GetCluster`
ListQueues	`pcs:ListQueues`	`pcs:GetCluster`
GetQueue	`pcs:GetQueue`	
UpdateQueue	`pcs:UpdateQueue`	`pcs:ListComputeNodeGroups, pcs:GetQueue`

APIaksi	Izin minimum	Izin tambahan untuk konsol
DeleteQueue	`pcs:DeleteQueue`	

Izin minimum yang diperlukan untuk menggunakan tag

Izin berikut diperlukan untuk menggunakan tag dengan sumber daya Anda di AWS PCS.

```
pcs:ListTagsForResource
pcs:TagResource
pcs:UntagResource
```

Izin minimum yang diperlukan untuk mendukung log

AWS PCSmengirimkan data log ke Amazon CloudWatch Logs (CloudWatch Log). Anda harus memastikan bahwa identifikasi Anda memiliki izin minimum untuk menggunakan CloudWatch Log. Untuk informasi selengkapnya, lihat Ringkasan mengelola izin akses ke sumber daya CloudWatch Log Anda di Panduan Pengguna Amazon CloudWatch Logs.

Untuk informasi tentang izin yang diperlukan bagi layanan untuk mengirim CloudWatch log ke Log, lihat Mengaktifkan logging dari AWS layanan di Panduan Pengguna CloudWatch Log Amazon.

Izin minimum untuk administrator layanan

IAMKebijakan berikut menentukan izin minimum yang diperlukan untuk IAM identitas (pengguna, grup, atau peran) untuk mengonfigurasi dan mengelola layanan. AWS PCS

> **Note**
>
> Pengguna yang tidak mengonfigurasi dan mengelola layanan tidak memerlukan izin ini. Pengguna yang hanya menjalankan pekerjaan menggunakan secure shell (SSH) untuk terhubung ke cluster. AWS Identity and Access Management (IAM) tidak menangani otentikasi atau otorisasi untuk. SSH

```
{
    "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": "*"
        }
    ]
}
```

Anda dapat mengecualikan izin berikut dari kebijakan dan sebagai gantinya menggunakan kebijakan terkelola terkait dilAM:

- `"firehose:*"`

  ```
  AmazonKinesisFirehoseFullAccess
  ```

- `"kms:*"`

  ```
  AWSKeyManagementServicePowerUser
  ```

- `"logs:*"`

  ```
  CloudWatchLogsFullAccess
  ```

- `"s3:*"`

```
AmazonS3FullAccess
```

IAMprofil instance untuk Layanan Komputasi AWS Paralel

Aplikasi yang berjalan pada sebuah EC2 instance harus menyertakan AWS kredensil dalam AWS API permintaan apa pun yang mereka buat. Kami menyarankan Anda menggunakan IAM peran untuk mengelola kredensil sementara pada instans. EC2 Anda dapat menentukan profil instance untuk melakukan ini, dan melampirkannya ke instance Anda. Untuk informasi selengkapnya, lihat IAMperan untuk Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud.

> **Note**
>
> Saat Anda menggunakan AWS Management Console untuk membuat IAM peran untuk AmazonEC2, konsol akan membuat profil instance secara otomatis dan memberinya nama yang sama dengan IAM peran tersebut. Jika Anda menggunakan AWS CLI, AWS API tindakan, atau AWS SDK untuk membuat IAM peran, Anda membuat profil instance sebagai tindakan terpisah. Untuk informasi selengkapnya, lihat Profil instans di Panduan Pengguna Amazon Elastic Compute Cloud.

Anda harus menentukan ARN profil instance saat membuat grup node komputasi. Anda dapat memilih profil instance yang berbeda untuk beberapa atau semua grup node komputasi.

Persyaratan Profil Instance

Nama Profil Instance

Profil IAM instance ARN harus dimulai dengan `AWSPCS` atau berisi `/aws-pcs/` di jalurnya.

Example

- `arn:aws:iam::*:instance-profile/AWSPCS-example-role-1`dan
- `arn:aws:iam::*:instance-profile/aws-pcs/example-role-2`.

Izin

Minimal, profil instans untuk AWS PCS harus menyertakan kebijakan berikut. Ini memungkinkan node komputasi untuk memberi tahu AWS PCS layanan ketika mereka beroperasi.

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

Kebijakan tambahan

Anda dapat mempertimbangkan untuk menambahkan kebijakan terkelola ke profil instans. Sebagai contoh:

- AmazonS3 ReadOnlyAccess menyediakan akses hanya-baca ke semua bucket S3.
- A mazonSSMManaged InstanceCore memungkinkan fungsionalitas inti layanan AWS Systems Manager, seperti akses jarak jauh langsung dari Amazon Management Console.
- CloudWatchAgentServerPolicyberisi izin yang diperlukan untuk digunakan AmazonCloudWatchAgent di server.

Anda juga dapat menyertakan IAM kebijakan Anda sendiri yang mendukung kasus penggunaan spesifik Anda.

Membuat profil instans

Anda dapat membuat profil instance langsung dari EC2 konsol Amazon. Untuk informasi selengkapnya, lihat Menggunakan profil instans di Panduan AWS Identity and Access Management Pengguna.

Pemecahan Masalah Identitas dan akses Layanan Komputasi AWS Paralel

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS PCS danIAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di AWS PCS
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS PCS sumber daya saya

Saya tidak berwenang untuk melakukan tindakan di AWS PCS

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika `mateojackson` IAM pengguna mencoba menggunakan konsol untuk melihat detail tentang *`my-example-widget`* sumber daya fiksi tetapi tidak memiliki izin `pcs:`*`GetWidget`* fiksi.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 pcs:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `pcs:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS PCS.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika IAM pengguna bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AWS PCS. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS PCS sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah AWS PCS mendukung fitur-fitur ini, lihatBagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke IAM pengguna lain Akun AWS yang Anda miliki di Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan IAM Pengguna.
- Untuk mempelajari cara menyediakan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna yang diautentikasi secara eksternal (federasi identitas) di Panduan Pengguna. IAM
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM Panduan Pengguna. IAM

Validasi kepatuhan untuk Layanan Komputasi AWS Paralel

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- Arsitektur untuk HIPAA Keamanan dan Kepatuhan di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat HIPAA aplikasi yang memenuhi syarat.

  > **Note**
  >
  > Tidak semua Layanan AWS HIPAA memenuhi syarat. Untuk informasi selengkapnya, lihat Referensi Layanan yang HIPAA Memenuhi Syarat.

- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi ()). ISO
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCIDSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.

- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan dalam Layanan Komputasi AWS Paralel

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan Infrastruktur dalam Layanan Komputasi AWS Paralel

Sebagai layanan terkelola, AWS Parallel Computing Service dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan API panggilan yang AWS dipublikasikan untuk mengakses AWS PCS melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Transportasi (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju yang sempurna (PFS) seperti (Ephemeral Diffie-Hellman) atau DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda dapat menggunakan AWS Security Token

Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Saat AWS PCS membuat cluster, layanan meluncurkan pengontrol Slurm di akun milik layanan, terpisah dari node komputasi di akun Anda. Untuk menjembatani komunikasi antara pengontrol dan node komputasi, AWS PCS buat Antarmuka Jaringan Elastis (ENI) lintas akun di akun Anda. VPC Pengontrol Slurm menggunakan ENI untuk mengelola dan berkomunikasi dengan node komputasi di berbagai tempat Akun AWS, menjaga keamanan dan isolasi sumber daya sambil memfasilitasi operasi yang efisien HPC dan AI/ML.

Analisis dan manajemen kerentanan dalam Layanan Komputasi AWS Paralel

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara Anda AWS dan Anda. Untuk informasi selengkapnya, lihat model tanggung jawab AWS bersama. AWS menangani tugas keamanan dasar untuk infrastruktur yang mendasari di akun layanan, seperti menambal sistem operasi pada instance pengontrol, konfigurasi firewall, dan pemulihan bencana AWS infrastruktur. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai. Untuk detail selengkapnya, lihat Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan.

Anda bertanggung jawab atas keamanan infrastruktur yang mendasari di Akun AWS:

- Pertahankan kode Anda, termasuk pembaruan dan patch keamanan.
- Menambal dan memperbarui sistem operasi pada instance grup node.
- Perbarui penjadwal untuk menyimpannya dalam versi yang didukung.
- Mengautentikasi dan mengenkripsi komunikasi antara klien pengguna dan node yang mereka sambungkan.

Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara

yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan global dalam kebijakan sumber daya untuk membatasi izin yang diberikan AWS Parallel Computing Service (AWS PCS) ke layanan lain ke sumber daya. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan penuh ARN sumber daya. Jika Anda tidak tahu sumber daya penuh ARN atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian yang tidak diketahui dari file. ARN Misalnya, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti bucket Amazon S3ARN, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin.

Nilai `aws:SourceArn` harus berupa clusterARN.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan AWS PCS untuk mencegah masalah wakil yang membingungkan.

```
{
"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"
      }
    }
  }
}
```

IAMperan untuk EC2 instans Amazon yang disediakan sebagai bagian dari grup node komputasi

AWS PCSsecara otomatis mengatur kapasitas EC2 Amazon untuk setiap grup node komputasi yang dikonfigurasi dalam sebuah cluster. Saat membuat grup node komputasi, pengguna harus memberikan profil IAM instance melalui `iamInstanceProfileArn` bidang. Profil instance menentukan izin yang terkait dengan instance yang disediakanEC2. AWS PCSmenerima peran apa pun yang memiliki `AWSPCS` awalan nama peran atau `/aws-pcs/` sebagai bagian dari jalur peran. `iam:PassRole`Izin diperlukan pada IAM identitas (pengguna atau peran) yang membuat atau memperbarui grup node komputasi. Ketika pengguna memanggil `CreateComputeNodeGroup` atau `UpdateComputeNodeGroup` API tindakan, AWS PCS periksa untuk melihat apakah pengguna diizinkan untuk melakukan `iam:PassRole` tindakan.

Contoh kebijakan berikut memberikan izin untuk hanya meneruskan IAM peran yang namanya dimulai. `AWSPCS`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/AWSPCS*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Praktik terbaik keamanan untuk Layanan Komputasi AWS Paralel

Bagian ini menjelaskan praktik terbaik keamanan yang khusus untuk Layanan Komputasi AWS Paralel (AWS PCS). Untuk mempelajari lebih lanjut tentang praktik terbaik keamanan di AWS, lihat Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan.

AMIkeamanan terkait

- Jangan gunakan AWS PCS sampel AMIs untuk beban kerja produksi. Sampel AMIs tidak didukung dan hanya ditujukan untuk pengujian.
- Secara teratur memperbarui sistem operasi dan perangkat lunak AWS PCS instance untuk mengurangi kerentanan.
- Gunakan AWS Systems Manager untuk mengotomatiskan penambalan dan menjaga kepatuhan terhadap kebijakan keamanan Anda.
- Hanya gunakan AWS PCS paket resmi yang diautentikasi yang diunduh dari AWS sumber resmi.
- Perbarui AWS PCS paket secara teratur pada node komputasi untuk menerima tambalan dan peningkatan keamanan. Pertimbangkan untuk mengotomatiskan proses ini untuk meminimalkan kerentanan.

Keamanan Manajer Beban Kerja Slurm

- Menerapkan kontrol akses dan pembatasan jaringan untuk mengamankan kontrol Slurm dan node komputasi. Hanya izinkan pengguna dan sistem tepercaya untuk mengirimkan pekerjaan dan mengakses perintah manajemen Slurm.
- Gunakan fitur keamanan bawaan Slurm, seperti otentikasi Slurm, untuk memastikan bahwa pengiriman pekerjaan dan komunikasi diautentikasi.
- Perbarui versi Slurm untuk menjaga kelancaran operasi dan dukungan cluster.

> **Important**
>
> Setiap cluster yang menggunakan versi Slurm yang telah mencapai akhir support life (EOSL) dihentikan segera. Gunakan tautan di bagian atas halaman panduan pengguna untuk berlangganan RSS umpan AWS PCS dokumentasi untuk menerima pemberitahuan saat versi Slurm mendekat. EOSL

Pemantauan dan pencatatan

- Gunakan Amazon CloudWatch Logs dan AWS CloudTrail untuk memantau dan merekam tindakan di cluster Anda dan Akun AWS. Gunakan data untuk pemecahan masalah dan audit.

Keamanan jaringan

- Terapkan AWS PCS cluster Anda secara terpisah VPC untuk mengisolasi HPC lingkungan Anda dari lalu lintas jaringan lainnya.
- Gunakan grup keamanan dan daftar kontrol akses jaringan (ACLs) untuk mengontrol lalu lintas masuk dan keluar ke AWS PCS instance dan subnet.
- Gunakan AWS PrivateLink atau VPC titik akhir untuk menjaga lalu lintas jaringan antara kluster Anda dan AWS layanan lain di dalam jaringan. AWS

Penebangan dan pemantauan untuk AWS PCS

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja dan AWS sumber daya Anda yang lain. AWS PCS AWSmenyediakan alat pemantauan berikut untuk menonton AWS PCS, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak CPU penggunaan atau metrik lain dari EC2 instans Amazon Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari EC2 instans Amazon CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- AWS CloudTrailmenangkap API panggilan dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

AWS PCSlog penjadwal

Anda dapat mengonfigurasi AWS PCS untuk mengirim data pencatatan terperinci dari penjadwal klaster ke Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3), dan Amazon Data Firehose. Ini dapat membantu pemantauan dan pemecahan masalah. Anda dapat mengatur log AWS PCS penjadwal menggunakan AWS PCS konsol, serta secara terprogram menggunakan or. AWS CLI SDK

Daftar Isi

- Menyiapkan log penjadwal menggunakan AWS CLI
  - Buat tujuan pengiriman
  - Aktifkan AWS PCS cluster sebagai sumber pengiriman
  - Connect sumber pengiriman cluster ke tujuan pengiriman
- Jalur dan nama aliran log penjadwal
- Contoh catatan log AWS PCS penjadwal

Prasyarat

IAMPrinsipal yang digunakan untuk mengelola AWS PCS cluster harus memungkinkan`pcs:AllowVendedLogDeliveryForResource`. Berikut adalah contoh AWS IAM kebijakan yang memungkinkannya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PcsAllowVendedLogsDelivery",
            "Effect": "Allow",
            "Action": ["pcs:AllowVendedLogDeliveryForResource"],
            "Resource": [
                "arn:aws:pcs:::cluster/*"
            ]
        }
    ]
}
```

Menyiapkan log penjadwal menggunakan konsol AWS PCS

Untuk mengatur log AWS PCS penjadwal di konsol, ikuti langkah-langkah berikut:

1. Buka AWS PCSkonsol.
2. Pilih Cluster dan navigasikan ke halaman detail untuk AWS PCS cluster tempat Anda akan mengaktifkan logging.
3. Pilih Log.
4. Di bawah pengiriman log - Log Penjadwal - opsional

a. Tambahkan hingga tiga tujuan pengiriman log. Pilihannya termasuk CloudWatch Log, Amazon S3, atau Firehose.

b. Pilih Perbarui pengiriman log.

Anda dapat mengkonfigurasi ulang, menambah, atau menghapus pengiriman log dengan mengunjungi kembali halaman ini.

Menyiapkan log penjadwal menggunakan AWS CLI

Untuk mencapai hal ini, Anda memerlukan setidaknya satu tujuan pengiriman, satu sumber pengiriman (PCScluster), dan satu pengiriman, yang merupakan hubungan yang menghubungkan sumber ke tujuan.

Buat tujuan pengiriman

Anda memerlukan setidaknya satu tujuan pengiriman untuk menerima log penjadwal dari sebuah AWS PCS cluster. Anda dapat mempelajari lebih lanjut tentang topik ini di PutDeliveryDestination bagian Panduan CloudWatch API Pengguna.

Untuk membuat tujuan pengiriman menggunakan AWS CLI

- Buat tujuan dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:
  - Ganti *region-code* dengan Wilayah AWS tempat Anda akan membuat tujuan Anda. Ini umumnya akan menjadi wilayah yang sama dengan tempat AWS PCS cluster digunakan.
  - Ganti *pcs-logs-destination* dengan nama pilihan Anda. Ini harus unik untuk semua tujuan pengiriman di akun Anda.
  - Ganti *resource-arn* dengan ARN untuk grup log yang ada di CloudWatch Log, bucket S3, atau aliran pengiriman di Firehose. Contohnya termasuk:
    - CloudWatch Grup log

      ```
      arn:aws:logs:region-code:account-id:log-group:/log-group-name:*
      ```

    - Ember S3

      ```
      arn:aws:s3:::bucket-name
      ```

    - Aliran pengiriman Firehose

```
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
```

Catat ARN untuk tujuan pengiriman baru, karena Anda akan memerlukannya untuk mengonfigurasi pengiriman.

Aktifkan AWS PCS cluster sebagai sumber pengiriman

Untuk mengumpulkan log penjadwal dari AWSPCS, konfigurasikan cluster sebagai sumber pengiriman. Untuk informasi selengkapnya, lihat PutDeliverySourcedi APIReferensi CloudWatch Log Amazon.

Untuk mengonfigurasi klaster sebagai sumber pengiriman menggunakan AWS CLI

- Aktifkan pengiriman log dari klaster Anda dengan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:
  - Ganti *region-code* dengan Wilayah AWS tempat cluster Anda digunakan.
  - Ganti *cluster-logs-source-name* dengan nama untuk sumber ini. Ini harus unik untuk semua sumber pengiriman di Anda Akun AWS. Pertimbangkan untuk memasukkan nama atau ID AWS PCS cluster.
  - Ganti *cluster-arn* dengan ARN untuk AWS PCS cluster Anda

```
aws logs put-delivery-source \
    --region region-code \
    --name cluster-logs-source-name \
    --resource-arn cluster-arn \
    --log-type PCS_SCHEDULER_LOGS
```

Connect sumber pengiriman cluster ke tujuan pengiriman

Agar data log penjadwal mengalir dari cluster ke tujuan, Anda harus mengonfigurasi pengiriman yang menghubungkannya. Untuk informasi selengkapnya, lihat CreateDeliverydi APIReferensi CloudWatch Log Amazon.

Untuk membuat pengiriman menggunakan AWS CLI

- Buat pengiriman menggunakan perintah berikut. Sebelum menjalankan perintah, buat penggantian berikut:
  - Ganti *region-code* dengan di Wilayah AWS mana sumber dan tujuan Anda ada.
  - Ganti *cluster-logs-source-name* dengan nama sumber pengiriman Anda dari atas.
  - Ganti *destination-arn* dengan ARN dari tujuan pengiriman tempat Anda ingin log dikirimkan.

```
aws logs create-delivery \
    --region region-code \
    --delivery-source-name cluster-logs-source \
    --delivery-destination-arn destination-arn
```

Jalur dan nama aliran log penjadwal

Jalur dan nama untuk log AWS PCS penjadwal bergantung pada jenis tujuan.

- CloudWatch Log
  - Aliran CloudWatch Log mengikuti konvensi penamaan ini.

    ```
    AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
    ```

    Example

    ```
    AWSLogs/PCS/abcdef0123/slurmctld_24.05.log
    ```
- Ember S3
  - Jalur keluaran bucket S3 mengikuti konvensi penamaan ini:

    ```
    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.
    ```
  - Nama objek S3 mengikuti konvensi ini:

```
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
```

Contoh catatan log AWS PCS penjadwal

AWSPCSlog scheduler terstruktur. Mereka termasuk bidang seperti pengidentifikasi cluster, jenis penjadwal, versi mayor dan patch, selain pesan log yang dipancarkan dari proses pengontrol Slurm. Inilah contohnya.

```
{
    "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"
}
```

Memantau Layanan Komputasi AWS Paralel dengan Amazon CloudWatch

Amazon CloudWatch menyediakan pemantauan kesehatan dan kinerja klaster AWS Parallel Computing Service (AWS PCS) Anda dengan mengumpulkan metrik dari klaster secara berkala. Metrik ini dipertahankan, memungkinkan Anda mengakses data historis dan mendapatkan wawasan tentang kinerja klaster Anda dari waktu ke waktu.

CloudWatch juga memungkinkan Anda memantau EC2 instans yang diluncurkan AWS PCS untuk memenuhi persyaratan penskalaan Anda. Meskipun Anda dapat memeriksa log pada instance yang sedang berjalan, CloudWatch metrik dan data logging biasanya dihapus setelah instance

dihentikan. Namun, Anda dapat mengonfigurasi CloudWatch agen pada instance menggunakan templat EC2 peluncuran untuk mempertahankan metrik dan log bahkan setelah penghentian instans, memungkinkan pemantauan dan analisis jangka panjang.

Jelajahi topik di bagian ini untuk mempelajari lebih lanjut tentang pemantauan AWS PCS menggunakan CloudWatch.

Topik

Memantau AWS PCS metrik menggunakan CloudWatch

Anda dapat memantau kesehatan AWS PCS klaster menggunakan Amazon CloudWatch, yang mengumpulkan data dari klaster Anda dan mengubahnya menjadi metrik mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang kinerja klaster Anda. Metrik klaster dikirim ke periode CloudWatch 1 menit. Untuk informasi selengkapnya CloudWatch, lihat Apa itu Amazon CloudWatch? di Panduan CloudWatch Pengguna Amazon.

AWS PCSmenerbitkan metrik berikut ke dalam PCS namespace AWS/di. CloudWatch Mereka memiliki dimensi tunggal,`ClusterId`.

Nama	Penjelasan	Unit
ActualCapacity	IdleCapacity + UtilizedCapacity	Hitung
CapacityUtilization	UtilizedCapacity / ActualCapacity	Hitung
DesiredCapacity	ActualCapacity + PendingCapacity	Hitung
IdleCapacity	Hitungan instance yang berjalan tetapi tidak dialokasikan untuk pekerjaan	Hitung

Nama	Penjelasan	Unit
UtilizedCapacity	Hitungan instance yang berjalan dan dialokasikan untuk pekerjaan	Hitung

Memantau AWS PCS instans menggunakan Amazon CloudWatch

AWSPCSmeluncurkan EC2 instans Amazon sesuai kebutuhan untuk memenuhi persyaratan penskalaan yang ditentukan dalam grup node PCS komputasi Anda. Anda dapat memantau instance ini saat dijalankan menggunakan Amazon CloudWatch. Anda dapat memeriksa log instance yang sedang berjalan dengan masuk ke dalamnya dan menggunakan alat baris perintah interaktif. Namun, secara default, data CloudWatch metrik hanya disimpan untuk jangka waktu terbatas setelah instance dihentikan, dan log instance biasanya dihapus bersama dengan EBS volume yang mendukung instance. Untuk menyimpan metrik atau data pencatatan dari instans yang diluncurkan PCS setelah dihentikan, Anda dapat mengonfigurasi CloudWatch agen pada instans Anda dengan templat peluncuran. EC2 Topik ini memberikan ikhtisar pemantauan instance yang sedang berjalan dan memberikan contoh cara mengonfigurasi metrik dan log instance persisten.

Memantau instance yang sedang berjalan

Menemukan AWS PCS contoh

Untuk memantau instance yang diluncurkan olehPCS, temukan instance yang sedang berjalan yang terkait dengan cluster atau grup node komputasi. Kemudian, di EC2 konsol untuk contoh tertentu, periksa bagian Status dan alarm dan Pemantauan. Jika akses login dikonfigurasi untuk instans tersebut, Anda dapat terhubung ke mereka dan memeriksa berbagai file log pada instance. Untuk informasi selengkapnya tentang mengidentifikasi instance mana yang dikelola olehPCS, lihatMenemukan instance grup node komputasi di AWS PCS.

Mengaktifkan metrik terperinci

Secara default, metrik instans dikumpulkan pada interval 5 menit. Untuk mengumpulkan metrik pada interval satu menit, aktifkan CloudWatch pemantauan terperinci dalam templat peluncuran grup node komputasi Anda. Untuk informasi selengkapnya, lihat Aktifkan CloudWatch pemantauan terperinci.

Mengonfigurasi metrik dan log instance persisten

Anda dapat menyimpan metrik dan log dari instans Anda dengan menginstal dan mengonfigurasi CloudWatch agen Amazon di dalamnya. Ini terdiri dari tiga langkah utama:

1. Buat konfigurasi CloudWatch agen.
2. Simpan konfigurasi di mana ia dapat diambil oleh PCS instance.
3. Tulis template EC2 peluncuran yang menginstal perangkat lunak CloudWatch agen, mengambil konfigurasi Anda, dan memulai CloudWatch agen menggunakan konfigurasi.

Untuk informasi selengkapnya, lihat Mengumpulkan metrik, log, dan jejak dengan CloudWatch agen di Panduan CloudWatch Pengguna Amazon, danMenggunakan template EC2 peluncuran Amazon dengan AWS PCS.

Buat konfigurasi CloudWatch Agen

Sebelum menerapkan CloudWatch agen pada instance Anda, Anda harus membuat file JSON konfigurasi yang menentukan metrik, log, dan jejak yang akan dikumpulkan. File konfigurasi dapat dibuat menggunakan wizard atau secara manual, menggunakan editor teks. File konfigurasi akan dibuat secara manual untuk demonstrasi ini.

Di komputer tempat Anda AWS CLI menginstal, buat file CloudWatch konfigurasi bernama config.json dengan konten yang mengikuti. Anda juga dapat menggunakan yang berikut ini URL untuk mengunduh salinan file.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

Catatan

- Jalur log dalam file sampel adalah untuk Amazon Linux 2. Jika instans Anda akan menggunakan sistem operasi dasar yang berbeda, ubah jalur yang sesuai.
- Untuk menangkap log lain, tambahkan entri tambahan di bawah`collect_list`.
- Nilai dalam `{brackets}` adalah variabel template. Untuk daftar lengkap variabel yang didukung, lihat Membuat atau mengedit file konfigurasi CloudWatch agen secara manual di Panduan CloudWatch Pengguna Amazon.
- Anda dapat memilih untuk menghilangkan `logs` atau `metrics` jika Anda tidak ingin mengumpulkan jenis informasi ini.

```
{
    "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
            }
        }
    }
}
```

File ini menginstruksikan CloudWatch agen untuk memantau beberapa file yang dapat membantu dalam mendiagnosis kesalahan dalam bootstrap misalnya, otentikasi dan login, dan domain pemecahan masalah lainnya. Ini termasuk:

- `/var/log/cloud-init.log`— Output dari tahap awal konfigurasi instance
- `/var/log/cloud-init-output.log`— Output dari perintah yang berjalan selama konfigurasi instance
- `/var/log/amazon/pcs/bootstrap.log`— Output dari operasi PCS -spesifik yang berjalan selama konfigurasi instance
- `/var/log/slurmd.log`— Output dari slurmd daemon manajer beban kerja Slurm
- `/var/log/messages`— Pesan sistem dari kernel, layanan sistem, dan aplikasi

- `/var/log/secure`— Log yang terkait dengan upaya otentikasi, sepertiSSH, sudo, dan peristiwa keamanan lainnya

File log dikirim ke grup CloudWatch log bernama`/PCSLogs/instances`. Aliran log adalah kombinasi dari ID instance dan nama dasar file log. Grup log memiliki waktu retensi 30 hari.

Selain itu, file menginstruksikan CloudWatch agen untuk mengumpulkan beberapa metrik umum, menggabungkannya dengan ID instance.

Simpan konfigurasi

File konfigurasi CloudWatch agen harus disimpan di tempat yang dapat diakses oleh instance node PCS komputasi. Ada dua cara umum untuk melakukan ini. Anda dapat mengunggahnya ke bucket Amazon S3 yang dapat diakses oleh instans grup node komputasi Anda melalui profil instansnya. Atau, Anda dapat menyimpannya sebagai parameter SSM di Amazon Systems Manager Parameter Store.

Unggah ke bucket S3

Untuk menyimpan file Anda di S3, gunakan AWS CLI perintah yang mengikuti. Sebelum menjalankan perintah, buat penggantian ini:

- Ganti *DOC-EXAMPLE-BUCKET* dengan nama bucket S3 Anda sendiri

Pertama, (ini opsional jika Anda memiliki bucket yang sudah ada), buat bucket untuk menyimpan file konfigurasi Anda.

```
aws s3 mb s3://DOC-EXAMPLE-BUCKET
```

Selanjutnya, unggah file ke ember.

```
aws s3 cp ./config.json s3://DOC-EXAMPLE-BUCKET/
```

Simpan sebagai SSM parameter

Untuk menyimpan file Anda sebagai SSM parameter, gunakan perintah berikut. Sebelum menjalankan perintah, buat penggantian ini:

- Ganti *region-code* dengan AWS wilayah tempat Anda bekerja dengan AWSPCS.

- (Opsional) Ganti *AmazonCloudWatch-PCS* dengan nama Anda sendiri untuk parameter. Perhatikan bahwa jika Anda mengubah awalan nama dari `AmazonCloudWatch-` Anda perlu secara khusus menambahkan akses baca ke SSM parameter di profil instance grup node Anda.

```
aws ssm put-parameter \
   --region region-code \
   --name "AmazonCloudWatch-PCS" \
   --type String \
   --value file://config.json
```

Tulis template EC2 peluncuran

Detail spesifik untuk template peluncuran tergantung pada apakah file konfigurasi Anda disimpan di S3 atauSSM.

Gunakan konfigurasi yang disimpan di S3

Skrip ini menginstal CloudWatch agen, mengimpor file konfigurasi dari bucket S3, dan meluncurkan agen dengannya. CloudWatch Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- *DOC-EXAMPLE-BUCKET* — Nama bucket S3 yang dapat dibaca akun Anda
- */config.json* — Jalur relatif terhadap root bucket S3 tempat konfigurasi disimpan

```
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==--
```

Profil IAM instance untuk grup node harus memiliki akses ke bucket. Berikut adalah contoh IAM kebijakan untuk bucket dalam skrip data pengguna di atas.

```
{
    "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/*"
            ]
        }
    ]
}
```

Perhatikan juga bahwa instance harus mengizinkan lalu lintas keluar ke S3 dan titik akhir. CloudWatch Ini dapat dilakukan dengan menggunakan grup keamanan atau VPC titik akhir, tergantung pada arsitektur cluster Anda.

Gunakan konfigurasi yang disimpan di SSM

Skrip ini menginstal CloudWatch agen, mengimpor file konfigurasi dari SSM parameter, dan meluncurkan CloudWatch agen dengannya. Ganti nilai berikut dalam skrip ini dengan detail Anda sendiri:

- (Opsional) Ganti *AmazonCloudWatch-PCS* dengan nama Anda sendiri untuk parameter.

```
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==--
```

Kebijakan IAM instance untuk grup node harus memiliki CloudWatchAgentServerPolicylampiran padanya.

Jika nama parameter Anda tidak dimulai dengan `AmazonCloudWatch-` Anda perlu secara khusus menambahkan akses baca ke SSM parameter di profil instance grup node Anda. Berikut adalah contoh IAM kebijakan yang menggambarkan ini untuk awalan *DOC-EXAMPLE-PREFIX*.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "CustomCwSsmMParamReadOnly",
      "Effect" : "Allow",
      "Action" : [
        "ssm:GetParameter"
      ],
      "Resource" : "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

Perhatikan juga bahwa instance harus memungkinkan lalu lintas keluar ke SSM dan CloudWatch titik akhir. Ini dapat dilakukan dengan menggunakan grup keamanan atau VPC titik akhir, tergantung pada arsitektur cluster Anda.

Logging API panggilan Layanan Komputasi AWS Paralel menggunakan AWS CloudTrail

AWS PCSterintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS PCS. CloudTrail menangkap semua API panggilan untuk AWS PCS sebagai acara. Panggilan yang diambil termasuk panggilan dari AWS PCS konsol dan panggilan kode ke AWS PCS API operasi. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk. AWS PCS Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat

peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat AWS PCS, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS PCSinformasi di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di AWS PCS, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk AWS PCS, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi SNS notifikasi Amazon untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua AWS PCS tindakan dicatat oleh CloudTrail dan didokumentasikan dalam APIReferensi Layanan Komputasi AWS Paralel. Misalnya, panggilan ke`CreateComputeNodeGroup`,`UpdateQueue`, dan `DeleteCluster` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan root atau AWS Identity and Access Management (IAM) kredensional pengguna.

- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lebih lanjut, lihat CloudTrail userIdentityelemen.

Memahami entri file CloudTrail log dari AWS PCS

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari API panggilan publik, sehingga tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log untuk `CreateQueue` tindakan.

```
{
    "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"
```

```
}
```

Titik akhir dan kuota layanan untuk AWS PCS

Bagian berikut menjelaskan titik akhir dan kuota layanan untuk AWS Parallel Computing Service ()AWS PCS. Kuota layanan, sebelumnya disebut sebagai batas, adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda. Akun AWS

Anda Akun AWS memiliki kuota default untuk setiap AWS layanan. Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah. Anda dapat meminta peningkatan untuk beberapa kuota dan kuota lainnya tidak dapat ditingkatkan.

Untuk informasi selengkapnya, lihat Service Quotas AWS di Referensi Umum AWS .

Daftar Isi

Titik akhir layanan

Nama Wilayah	Wilayah	Titik Akhir	Protokol
US East (N. Virginia)	us-east-1	pcs.us-east-1.amazonaws.com	HTTPS
AS Timur (Ohio)	us-east-2	pcs.us-east-2.amazonaws.com	HTTPS
AS Barat (Oregon)	us-west-2	pcs.us-west-2.amazonaws.com	HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	pcs.ap-southeast-1.amazonaws.com	HTTPS
Asia Pasifik (Sydney)	ap-southeast-2	pcs.ap-southeast-2.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Tokyo)	ap-northeast-1	pcs.ap-northeast-1.amazonaws.com	HTTPS
Eropa (Frankfurt)	eu-central-1	pcs.eu-central-1.amazonaws.com	HTTPS
Eropa (Irlandia)	eu-west-1	pcs.eu-west-1.amazonaws.com	HTTPS
Eropa (Stockholm)	eu-north-1	pcs.eu-north-1.amazonaws.com	HTTPS

Kuota layanan

Nama	Default	Dapat disesuaikan	Deskripsi
Klaster	5	Ya	Jumlah maksimum cluster per. Wilayah AWS

> **Note**
>
> Nilai default adalah kuota awal yang ditetapkan oleh AWS. Nilai default ini terpisah dari nilai kuota yang diterapkan aktual dan kuota layanan maksimum yang mungkin. Untuk informasi selengkapnya, lihat Terminologi dalam Service Quotas di Panduan Pengguna Service Quotas.

Kuota layanan ini tercantum di bawah Layanan Komputasi AWS Paralel (PCS) di. AWS Management Console Untuk meminta peningkatan kuota untuk nilai yang ditampilkan sebagai dapat disesuaikan, lihat Meminta Peningkatan Kuota di Panduan Pengguna Service Quotas.

> **Important**
>
> Ingatlah untuk memeriksa Wilayah AWS pengaturan saat ini di AWS Management Console.

Kuota internal

Kuota berikut bersifat internal dan tidak dapat disesuaikan.

Nama	Default	Dapat disesuaikan	Deskripsi
Pembuatan cluster bersamaan	1	Tidak	Jumlah maksimum cluster di `Creating` negara bagian per Wilayah AWS.

Kuota yang relevan untuk layanan lain AWS

AWS PCSmenggunakan AWS layanan lain. Kuota layanan Anda untuk layanan tersebut memengaruhi penggunaan AWS PCS Anda.

Kuota EC2 layanan Amazon yang berdampak AWS PCS

- Permintaan instans spot
- Menjalankan instans sesuai permintaan
- Templat peluncuran
- Luncurkan versi template
- EC2APIPermintaan Amazon

Untuk informasi selengkapnya, lihat kuota EC2 layanan Amazon di Panduan Pengguna Amazon Elastic Compute Cloud.

Catatan rilis untuk AWS PCS sampel AMIs

AWS PCSsampel AMIs memiliki irama rilis malam untuk patch keamanan. Patch keamanan tambahan ini tidak termasuk dalam catatan rilis resmi.

> **Important**
>
> Sampel AMIs adalah untuk tujuan demonstrasi dan tidak direkomendasikan untuk beban kerja produksi.

Daftar Isi

AWS PCScontoh x86_64 AMI untuk Slurm 23.11 (Amazon Linux 2)

Dokumen ini menjelaskan perubahan terbaru, penambahan, masalah yang diketahui, dan perbaikan untuk AWS PCS Sample x86_64 AMI (Amazon Linux 2).

- Tanggal Dibuat: Juli 15, 2024
- Tanggal Rilis: 22 Aug 2024
- Terakhir Diperbarui: 22 Agt 2024

AMInama

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

EC2Contoh yang didukung

- Semua instance dengan prosesor x86 64-bit. Untuk menemukan instance yang kompatibel, navigasikan ke EC2konsol Amazon. Pilih Jenis Instance, lalu cari`Architectures=x86_64`.

AMIisi

- AWS Layanan yang Didukung: AWS PCS

- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: x86_64
- Kernel Linux: 5.10.220-209.867.amzn2.x86_64
- EBSjenis volume: gp2
- AWS PCSPenginstal Slurm 23.11:23.11.9-1
- AWS PCSpenginstal perangkat lunak: 1.0.0-1
- EFAPemasang: 1.33.0
- GDRCopy: 2.4
- NVIDIAPengemudi: 535.154.05
- NVIDIACUDA: 12.2.2_535.104.05

Pemberitahuan

- Tidak ada

Tanggal rilis: 2024-08-22

Diperbarui

- Tidak ada. Rilis pertama.

Ditambahkan

- Tidak ada. Rilis pertama.

Dihapus

- Tidak ada. Rilis pertama.

AWS PCScontoh Arm64 AMI untuk Slurm 23.11 (Amazon Linux 2)

Dokumen ini menjelaskan perubahan terbaru, penambahan, masalah yang diketahui, dan perbaikan untuk AWS PCS Sample Arm64 (AMIAmazon Linux 2).

- Tanggal Dibuat: Juli 15, 2024

- Tanggal Rilis: 22 Aug 2024
- Terakhir Diperbarui: 22 Agt 2024

AMInama

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

EC2Contoh yang didukung

- Semua instance dengan prosesor Arm 64-bit. Untuk menemukan instance yang kompatibel, navigasikan ke EC2konsol Amazon. Pilih Jenis Instance, lalu cari`Architectures=arm64`.

AMIisi

- AWS Layanan yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: arm64
- Kernel Linux: 5.10.220-209.867.amzn2.aarch64
- EBSjenis volume: gp2
- AWS PCSPenginstal Slurm 23.11:23.11.9-1
- AWS PCSpenginstal perangkat lunak: 1.0.0-1
- EFAPemasang: 1.33.0
- GDRCopy: 2.4
- NVIDIAPengemudi: 535.154.05
- NVIDIACUDA: 12.2.2_535.104.05

Pemberitahuan

- Tidak ada

Tanggal rilis: 2024-08-22

Diperbarui

- Tidak ada. Rilis pertama.

Ditambahkan

- Tidak ada. Rilis pertama.

Dihapus

- Tidak ada. Rilis pertama.

Riwayat dokumen untuk Panduan Pengguna AWS PCS

Tabel berikut menjelaskan rilis dokumentasi untuk AWS PCS.

Tanggal	Perubahan	Pembaruan dokumentasi	APIversi diperbarui
Agustus 28, 2024	Halaman kebijakan terkelola ditambahkan	Untuk informasi selengkapnya, lihat AWS kebijakan terkelola untuk Layanan Komputasi AWS Paralel.	N/A
Agustus 28, 2024	AWS PCSrilis	Rilis awal panduan AWS PCS pengguna.	AWS SDK: 2024-08-28

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.