



Panduan Pengguna

AWS PCS



AWS PCS: Panduan Pengguna

Copyright © 2026 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 merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS PCS?	1
Konsep	1
Memulai dengan AWS PCS	3
Prasyarat	4
Mendaftar AWS dan membuat pengguna administratif	5
Instal AWS CLI untuk AWS PCS	7
Izin IAM yang diperlukan	7
Menggunakan CloudFormation	8
Buat VPC dan subnet	8
Temukan grup keamanan default untuk VPC cluster	10
Buat grup keamanan	10
Buat grup keamanan	10
Membuat klaster	11
Buat penyimpanan bersama di Amazon EFS	12
Buat penyimpanan bersama FSx untuk Lustre	13
Buat grup node komputasi	14
Buat profil instance	14
Buat template peluncuran	16
Buat grup node komputasi untuk node login	18
Buat grup node komputasi untuk pekerjaan	19
Membuat antrean	20
Connect ke cluster Anda	21
Jelajahi lingkungan cluster	22
Ubah pengguna	22
Bekerja dengan sistem file bersama	22
Berinteraksi dengan Slurm	23
Jalankan pekerjaan node tunggal	24
Jalankan pekerjaan MPI multi-node dengan Slurm	26
Hapus AWS sumber daya Anda	28
Memulai dengan CloudFormation dan AWS PCS	31
Gunakan CloudFormation untuk membuat cluster	31
Connect ke sebuah cluster	33
Bersihkan cluster	34
Bagian dari CloudFormation template untuk AWS PCS	34

Header	35
Metadata	35
Parameter	36
Pemetaan	38
Sumber Daya	38
Output	42
Template untuk membuat cluster sampel	43
Klaster	45
Membuat kluster	45
Prasyarat	46
Buat cluster AWS PCS	46
Memperbarui kluster	50
Manfaat pembaruan kluster	50
Perubahan konfigurasi yang didukung	51
Batasan	51
Prasyarat untuk pembaruan cluster	51
Perbarui proses dan dampak pekerjaan	52
Penagihan selama pembaruan	52
Perbarui kluster	52
Pertanyaan yang Sering Diajukan	54
Penyelesaian Masalah	55
Menghapus kluster	56
Pertimbangan saat menghapus cluster PCS AWS	57
Hapus cluster	57
Ukuran cluster	58
Rahasia cluster	59
Gunakan AWS Secrets Manager untuk menemukan rahasia cluster	59
Gunakan AWS PCS untuk menemukan rahasia cluster	60
Dapatkan rahasia cluster Slurm	61
Rotasi rahasia	62
Hitung grup simpul	67
Membuat grup node komputasi	67
Prasyarat	68
Buat grup node komputasi di AWS PCS	68
Memperbarui grup node komputasi	74
Opsi untuk memperbarui grup node komputasi AWS PCS	74

Pertimbangan saat memperbarui grup node komputasi AWS PCS	74
Untuk memperbarui grup node komputasi AWS PCS	76
Menghapus grup node komputasi	77
Pertimbangan saat menghapus grup node komputasi	78
Hapus grup node komputasi	78
Dapatkan detail grup node komputasi	79
Menemukan instance grup node komputasi	82
Menggunakan template peluncuran	85
Ikhtisar	85
Buat template peluncuran dasar	87
Bekerja dengan data pengguna Amazon EC2	89
Contoh: Instal perangkat lunak dari repositori paket	91
Contoh: Jalankan skrip dari bucket S3	91
Contoh: Mengatur variabel lingkungan global	93
Contoh: Gunakan sistem file EFS sebagai direktori home bersama	93
Reservasi Kapasitas	95
Menggunakan ODCRs dengan AWS PCS	95
Blok Kapasitas	98
Parameter template peluncuran yang berguna	103
Aktifkan CloudWatch pemantauan terperinci	103
Layanan Metadata Instans Versi 2 (IMDS v2)	104
Antrean	106
Membuat antrian	106
Prasyarat	106
Untuk membuat antrian di AWS PCS	107
Memperbarui antrian	109
Pertimbangan saat memperbarui antrian AWS PCS	109
Untuk memperbarui antrian AWS PCS	109
Menghapus antrian	111
Pertimbangan saat menghapus antrian	111
Hapus antrian	111
Node masuk	113
Menggunakan grup node komputasi untuk login	113
Membuat grup node komputasi AWS PCS untuk node login	113
Memperbarui grup node komputasi AWS PCS untuk node login	114
Menghapus grup node komputasi AWS PCS untuk node login	115

Menggunakan instance mandiri sebagai node login	115
Langkah 1 - Ambil alamat dan rahasia untuk cluster AWS PCS target	115
Langkah 2 - Luncurkan instans EC2	117
Langkah 3 - Instal Slurm pada instance	118
Langkah 4 - Ambil dan simpan rahasia cluster	118
Langkah 5 - Konfigurasi koneksi ke cluster AWS PCS	119
Langkah 6 - (Opsional) Uji koneksi	121
Menghubungkan node login mandiri ke beberapa cluster	121
Prasyarat	122
Kode skrip	124
Menggunakan skrip	132
Jaringan	135
Persyaratan VPC dan subnet	135
Persyaratan dan pertimbangan VPC	135
Persyaratan dan pertimbangan subnet	136
Membuat VPC	138
Prasyarat	139
Buat VPC Amazon	139
Grup keamanan	141
Persyaratan grup keamanan	141
Beberapa antarmuka jaringan	143
Grup penempatan	144
Menggunakan Elastic Fabric Adapter (EFA)	145
Identifikasi instans EC2 yang mendukung EFA	146
Buat grup keamanan untuk mendukung komunikasi EFA	146
(Opsional) Buat grup penempatan	148
Membuat atau memperbarui template peluncuran EC2	148
Membuat atau memperbarui grup node komputasi untuk EFA	149
(Opsional) Uji EFA	149
(Opsional) Gunakan CloudFormation templat untuk membuat templat peluncuran berkemampuan EFA	151
Sistem file jaringan	154
Pertimbangan untuk menggunakan sistem file jaringan	154
Contoh pemasangan jaringan	154
Gambar Mesin Amazon (AMIs)	160
Menggunakan sampel AMIs	160

Temukan sampel AWS PCS saat ini AMIs	161
Pelajari lebih lanjut tentang sampel AWS PCS AMIs	162
Bangun sendiri yang AMIs kompatibel dengan AWS PCS	162
Kustom AMIs	162
Langkah 1 - Luncurkan instance sementara	163
Langkah 2 - Instal agen AWS PCS	164
Langkah 3 - Instal Slurm	167
Langkah 4 - (Opsional) Instal driver tambahan, perpustakaan, dan perangkat lunak aplikasi	170
Langkah 5 - Buat AMI yang kompatibel dengan AWS PCS	170
Langkah 6 - Gunakan AMI kustom dengan grup node komputasi AWS PCS	171
Langkah 7 - Hentikan instance sementara	173
Installer untuk membangun AMIs	173
AWS Penginstal perangkat lunak agen PCS	174
Pemasang slurm	174
Sistem operasi yang didukung	175
Tipe instans yang didukung	175
Versi Slurm yang didukung	175
Verifikasi penginstal menggunakan checksum	175
Catatan rilis untuk AMIs	181
Contoh AMIs untuk x86_64 () AL2	182
Contoh AMIs untuk Arm64 () AL2	185
Sistem operasi yang didukung	189
AWS Versi agen PCS	191
Slurm	195
Versi slurm	195
Versi Slurm yang didukung di PCS AWS	196
Versi Slurm yang tidak didukung di PCS AWS	197
Catatan rilis	197
Pertanyaan umum	199
Akuntansi slurm	202
Memodifikasi pengaturan akuntansi	203
Konsep utama	203
Dapatkan konfigurasi akuntansi untuk cluster AWS PCS yang ada	205
Slurm REST API	205
Kasus penggunaan umum	206

Persyaratan dan pembatasan	206
Aktifkan REST API	207
Otentikasi REST API	209
Gunakan REST API	213
SISA API FAQ	215
Slurm reboot	218
Manfaat reboot Slurm	218
Kapan menggunakan reboot Slurm	218
Batasan	219
Reboot node komputasi	219
Batalkan reboot	221
Pertanyaan yang Sering Diajukan	221
Pemecahan masalah	223
Pengaturan Slurm kustom	224
Manfaat pengaturan Slurm kustom	224
Mengkonfigurasi pengaturan khusus	224
Validasi dan penanganan kesalahan	226
Batasan	226
Pengaturan cluster	227
Hitung pengaturan grup node	229
Pengaturan antrian	229
Pemecahan masalah	230
Plugin SPANK	231
Instal plugin SPANK	231
Konfigurasikan plugin SPANK	232
FAQ plugin SPANK	233
Plugin Filter CLI Slurm	234
Persyaratan	234
Keterbatasan dan pertimbangan keamanan	234
Konfigurasikan Plugin Filter CLI	235
Menggunakan Amazon S3 untuk menyebarkan skrip Plugin Filter CLI	239
Terjemahkan skrip plugin Job Submit	240
Pertanyaan yang Sering Diajukan	241
Pemecahan masalah	243
Keamanan	245
Perlindungan data	246

Enkripsi saat diam	247
Enkripsi saat bergerak	247
Manajemen kunci	248
Privasi lalu lintas antar jaringan	248
Menkripsi lalu lintas API	249
Menkripsi lalu lintas data	249
Kebijakan kunci KMS untuk volume EBS terenkripsi	249
Titik akhir antarmuka VPC (AWS PrivateLink)	256
Pertimbangan-pertimbangan	256
Membuat sebuah titik akhir antarmuka	256
Membuat kebijakan titik akhir	257
Identity and Access Management	258
Audiens	258
Mengautentikasi dengan identitas	259
Mengelola akses menggunakan kebijakan	260
Bagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM	262
Contoh kebijakan berbasis identitas	267
AWS kebijakan terkelola	271
Peran terkait layanan	273
Peran EC2 Spot	275
Izin minimum	276
Profil instans	284
Pemecahan masalah	288
Validasi kepatuhan	290
Ketahanan	290
Keamanan Infrastruktur	291
Analisis dan manajemen kerentanan	291
Pencegahan "confused deputy" lintas layanan	292
Peran IAM untuk instans Amazon EC2 disediakan sebagai bagian dari grup node komputasi	294
Praktik terbaik keamanan	295
Keamanan terkait AMI	295
Keamanan Manajer Beban Kerja Slurm	295
Pencatatan dan pemantauan	296
Keamanan jaringan	296
Pencatatan log dan pemantauan	297

Log penyelesaian Job	297
Prasyarat	298
Siapkan log penyelesaian pekerjaan	299
Cara menemukan log penyelesaian pekerjaan	301
Bidang log penyelesaian pekerjaan	301
Contoh log penyelesaian pekerjaan	305
Log penjadwal	308
Prasyarat	309
Siapkan log penjadwal	309
Jalur dan nama aliran log penjadwal	311
Contoh catatan log penjadwal	312
Pemantauan CloudWatch dengan	313
Metrik pemantauan	313
Pemantauan instans	314
CloudTrail log	323
AWS Informasi PCS di CloudTrail	323
Memahami entri file CloudTrail log dari AWS PCS	324
Titik akhir dan kuota layanan	327
Titik akhir layanan	327
Kuota layanan	330
Kuota internal	331
Kuota yang relevan untuk layanan lain AWS	331
Pemecahan masalah	333
Instans EC2 dihentikan dan diganti setelah reboot	333
Memecahkan masalah bootstrap node komputasi dan masalah pendaftaran di PCS AWS	334
Bagaimana Slurm bekerja pada PCS AWS	335
Ambil log contoh	336
Mengambil VPC/Subnet/Security Grup dari ID instance	337
Masalah pendaftaran node	338
Masalah bergabung dengan kluster slurm	340
Riwayat dokumen	344
AWS Glosarium	368
.....	ccclxix

Apa itu Layanan Komputasi AWS Paralel?

AWS Parallel Computing Service (AWS PCS) adalah layanan terkelola yang memudahkan menjalankan dan menskalakan beban kerja komputasi kinerja tinggi (HPC), dan membangun model ilmiah dan teknik AWS menggunakan Slurm. Gunakan AWS PCS untuk membangun cluster komputasi yang paling terintegrasi dalam AWS komputasi, penyimpanan, jaringan, dan visualisasi kelas. Jalankan simulasi atau bangun model ilmiah dan teknik. Rampingkan 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.

Topik

- [Konsep dalam AWS PCS](#)

Konsep dalam AWS PCS

Sebuah cluster di 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 arsitektur HPC yang canggih.

Klaster

Cluster adalah sumber daya untuk mengelola sumber daya dan menjalankan beban kerja. Cluster adalah sumber daya AWS PCS 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 VPC mana Anda ingin sumber daya cluster diluncurkan. Penjadwal menerima dan menjadwalkan pekerjaan, dan juga meluncurkan node komputasi (EC2 instance) yang memproses pekerjaan tersebut.

Hitung grup simpul

Grup node komputasi adalah kumpulan node komputasi yang digunakan AWS PCS 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, subnet VPC target, Amazon Machine Image (AMI), opsi pembelian, dan konfigurasi

peluncuran kustom. AWS PCS menggunakan 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 secara langsung ke node komputasi atau grup node komputasi.

Administrator sistem

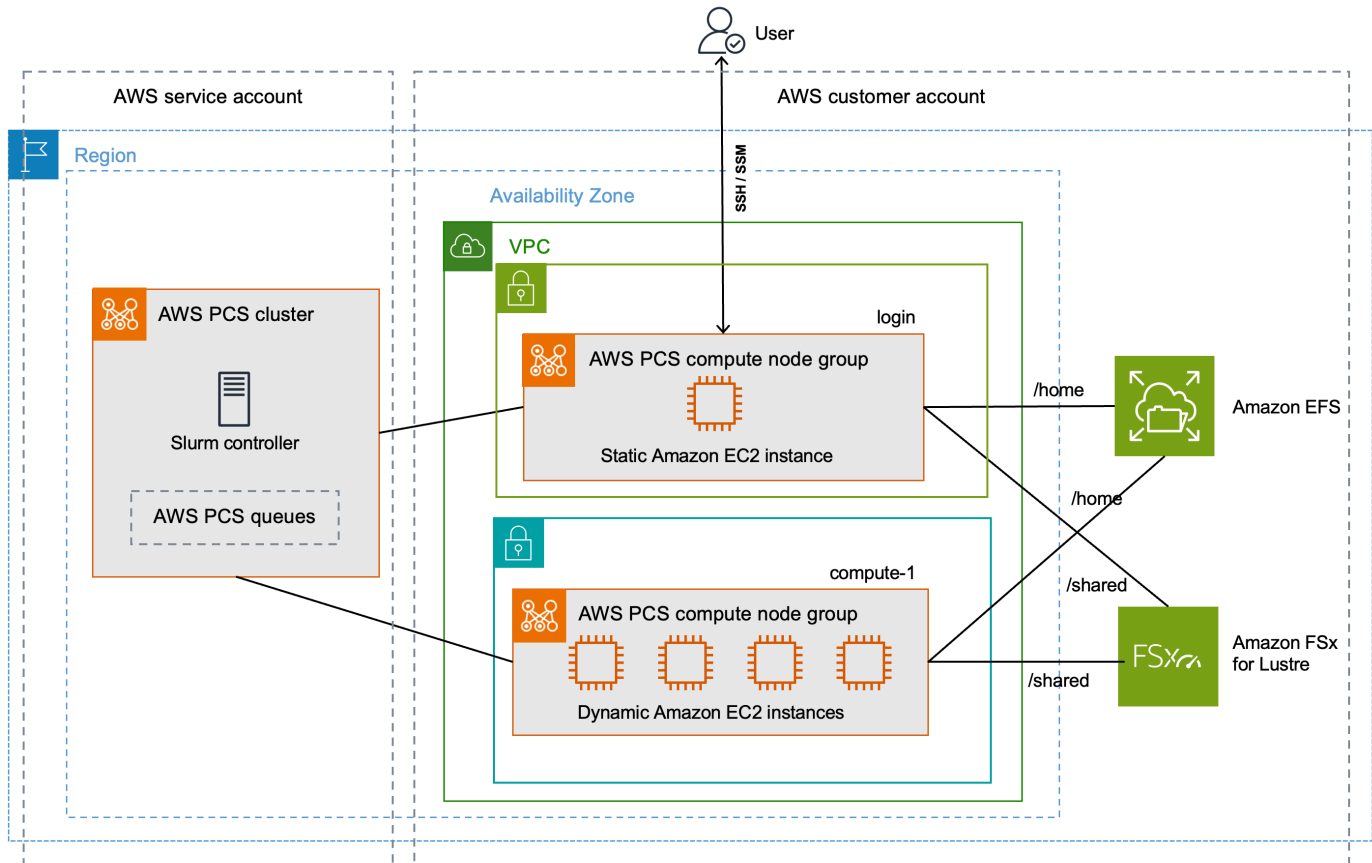
Administrator sistem menyebarkan, memelihara, dan mengoperasikan cluster. Mereka dapat mengakses AWS PCS melalui Konsol Manajemen AWS, AWS PCS API, dan AWS SDK. Mereka memiliki akses ke cluster tertentu melalui SSH atau AWS Systems Manager, di mana mereka dapat menjalankan tugas administratif, menjalankan 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 (seperti SSH) untuk mengakses sumber daya cluster, menjalankan pekerjaan, mengelola data, dan melakukan aktivitas berbasis shell lainnya.

Memulai Layanan Komputasi AWS Paralel

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:

- VPC dan subnet yang memenuhi persyaratan jaringan [AWS PCS](#).
- Sistem file Amazon EFS, yang akan digunakan sebagai direktori home bersama.
- Sistem file Amazon FSx untuk Lustre, yang menyediakan direktori kinerja tinggi bersama.
- Cluster AWS PCS, yang menyediakan pengontrol Slurm.
- 2 AWS PCS 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, peran IAM, dan templat EC2 peluncuran, yang tidak ditampilkan dalam diagram.

Note

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](#).

Topik

- [Prasyarat untuk memulai dengan PCS AWS](#)
- [Menggunakan AWS CloudFormation dengan tutorial AWS PCS](#)
- [Buat VPC dan subnet untuk PCS AWS](#)
- [Buat grup keamanan untuk AWS PCS](#)
- [Buat cluster di AWS PCS](#)
- [Buat penyimpanan bersama untuk AWS PCS di Amazon Elastic File System](#)
- [Buat penyimpanan bersama untuk AWS PCS di Amazon FSx untuk Lustre](#)
- [Buat grup node komputasi di AWS PCS](#)
- [Buat antrian untuk mengelola pekerjaan di AWS PCS](#)
- [Connect ke klaster AWS PCS Anda](#)
- [Jelajahi lingkungan cluster di AWS PCS](#)
- [Jalankan pekerjaan node tunggal di AWS PCS](#)
- [Jalankan pekerjaan MPI multi-node dengan Slurm di PCS AWS](#)
- [Hapus AWS sumber daya Anda untuk AWS PCS](#)

Prasyarat untuk memulai dengan PCS AWS

Lihat topik-topik berikut untuk mempersiapkan lingkungan pengembangan Anda Akun AWS dan lokal untuk AWS PCS.

Topik

- [Mendaftar AWS dan membuat pengguna administratif](#)
- [Instal AWS CLI untuk AWS PCS](#)
- [Izin IAM yang diperlukan untuk PCS AWS](#)

Mendaftar AWS dan membuat pengguna administratif

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

Topik

- [Mendaftar untuk Akun AWS](#)
- [Buat pengguna dengan akses administratif](#)

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 menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. 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 mengirimkan 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.com/ke/> dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftarkan 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 [Konsol Manajemen AWS](#) sebagai 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. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

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

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, 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 Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

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

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, 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 untuk AWS PCS

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

Anda harus mengkonfigurasi file AWS CLI. Untuk informasi selengkapnya, lihat [Mengkonfigurasi AWS CLI](#) dalam Panduan AWS Command Line Interface Pengguna untuk Versi 2.

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

```
aws pcs help
```

Izin IAM yang diperlukan untuk PCS AWS

Prinsip keamanan IAM yang Anda gunakan harus memiliki izin untuk bekerja dengan peran AWS PCS IAM, peran terkait layanan,, VPC, AWS CloudFormation dan sumber daya terkait. Untuk informasi selengkapnya [Identity 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
```

Menggunakan AWS CloudFormation dengan tutorial AWS PCS

Tutorial AWS PCS memiliki banyak langkah dan dimaksudkan untuk membantu Anda memahami bagian-bagian dari cluster AWS PCS dan prosedur yang diperlukan untuk membuatnya. Kami menyarankan Anda melalui langkah-langkah tutorial setidaknya 1 kali. Setelah Anda memiliki pemahaman yang baik tentang apa yang terlibat, Anda dapat menggunakan AWS CloudFormation untuk membuat cluster sampel dengan cepat dengan otomatisasi.

CloudFormation adalah AWS layanan yang memungkinkan Anda membuat dan menyediakan penyebaran AWS infrastruktur yang dapat diprediksi dan berulang kali. Anda dapat menggunakan CloudFormation template untuk secara otomatis menyediakan AWS sumber daya untuk cluster sampel sebagai satu unit, yang disebut tumpukan. Anda dapat menghapus tumpukan ketika Anda selesai dengan itu.

Untuk informasi selengkapnya, lihat [Memulai dengan CloudFormation dan AWS PCS](#).

Buat VPC dan subnet untuk PCS AWS

Anda dapat membuat VPC dan subnet dengan template. CloudFormation Gunakan URL berikut untuk mengunduh CloudFormation templat, lalu unggah templat di [CloudFormation konsol](#) untuk membuat CloudFormation tumpukan baru. Untuk informasi selengkapnya, lihat [Menggunakan 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 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 sumber daya IAM.

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

Temukan grup keamanan default untuk VPC cluster

Untuk menemukan ID untuk grup keamanan default di VPC baru, ikuti prosedur ini:

- Arahkan ke [konsol VPC Amazon](#).
- Di bawah Dasbor VPC, pilih Filter berdasarkan VPC.
 - Pilih VPC tempat 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 peluncuran EC2.

Buat grup keamanan untuk AWS PCS

AWS PCS bergantung pada grup keamanan untuk mengelola lalu lintas jaringan masuk dan keluar dari cluster dan grup node komputasinya. Untuk informasi rinci tentang topik ini, lihat [Persyaratan dan pertimbangan kelompok keamanan](#).

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

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

Buat grup keamanan untuk AWS PCS

Anda dapat menggunakan CloudFormation template untuk membuat grup keamanan. Gunakan URL berikut untuk mengunduh CloudFormation templat, lalu unggah templat di [CloudFormation konsol](#) untuk membuat CloudFormation tumpukan baru. Untuk informasi selengkapnya, lihat [Menggunakan 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 dimulai. `hpc-networking`
 - (Opsional) Di bawah ClientIpCidr, masukkan rentang IP yang lebih ketat untuk grup keamanan SSH 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 PUBLIK Anda sendiri. Anda bisa mendapatkan IP publik Anda menggunakan alat-alat seperti <https://ifconfig.co/>)

Pantau status CloudFormation tumpukan. Ketika mencapai kelompok keamanan `CREATE_COMPLETE`, sumber daya sudah 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 SSH masuk


Buat cluster di AWS PCS

Di AWS PCS, cluster adalah sumber daya persisten untuk mengelola sumber daya dan menjalankan beban kerja. Anda membuat cluster untuk penjadwal tertentu (AWS PCS saat ini mendukung Slurm) di subnet VPC baru atau yang sudah ada. Cluster menerima dan menjadwalkan pekerjaan, dan juga meluncurkan node komputasi (EC2 instance) yang memproses pekerjaan tersebut.

Untuk membuat klaster Anda

1. Buka [konsol AWS PCS](#) dan pilih Buat cluster.
2. Di bagian Cluster details, masukkan bidang-bidang berikut:
 - Nama cluster — Enter `get-started`
 - Scheduler - Pilih Slurm Versi 25.05
 - Ukuran pengontrol - Pilih Kecil

3. Di bagian Jaringan, pilih nilai untuk bidang berikut:
 - VPC — Pilih 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 untuk AWS PCS di Amazon Elastic File System

Amazon Elastic File System (Amazon EFS) 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 demonstrasi AWS PCS menggunakan sistem file EFS untuk menyediakan direktori home bersama antara node cluster. Buat sistem file EFS di VPC yang sama dengan cluster Anda.

Untuk membuat sistem file Amazon EFS Anda

1. Buka [konsol Amazon EFS](#).
2. Pastikan itu diatur ke AWS Region 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 bernama `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. Gunakan Lustre untuk beban kerja di mana kecepatan penting, seperti machine learning, komputasi performa 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 demonstrasi AWS PCS dapat menggunakan sistem file FSx for Lustre untuk menyediakan direktori bersama berkinerja tinggi antara node cluster. Buat sistem file FSx for Lustre di VPC yang sama dengan cluster Anda.

Untuk membuat sistem FSx file Lustre Anda

1. Pergi ke [FSx konsol Amazon](#).
2. Pastikan konsol diatur untuk menggunakan yang AWS Region 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 bernama `hpc-networking:Large-Scale-HPC`
 - Untuk Grup Keamanan VPC, biarkan grup keamanan bernama `default`
 - Untuk Subnet, pilih subnet tempat nama dimulai `hpc-networking:PrivateSubnetA`
 - Biarkan opsi lain disetel 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 (instans EC2) yang diluncurkan dan dikelola AWS PCS. Saat menentukan grup node komputasi, Anda menentukan ciri umum seperti tipe instans EC2, jumlah instans minimum dan maksimum, subnet VPC target, opsi pembelian pilihan, dan konfigurasi peluncuran kustom. AWS PCS secara 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](#)
- [Buat template peluncuran untuk AWS PCS](#)
- [Buat grup node komputasi untuk node login di AWS PCS](#)
- [Buat grup node komputasi untuk menjalankan pekerjaan komputasi di PCS AWS](#)

Buat profil instance untuk AWS PCS

Grup node komputasi memerlukan profil instance saat dibuat. Jika Anda menggunakan Konsol Manajemen AWS untuk membuat peran Amazon EC2, konsol akan secara otomatis membuat profil

instans 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 Konsol Manajemen AWS untuk membuat peran untuk Amazon EC2, yang juga membuat profil instance untuk grup node komputasi Anda.

Untuk membuat profil peran dan contoh

- Arahkan ke [konsol IAM](#).
- 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:

JSON

```
{
  "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.
 - Dalam kebijakan Izin, cari Amazon SSManaged InstanceCore.
 - Centang kotak di samping Amazon SSManaged 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 peluncuran EC2 yang digunakan AWS PCS untuk mengonfigurasi instans EC2 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 templat peluncuran EC2. 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 akses SSH masuk.

Akses CloudFormation template

Gunakan URL berikut untuk mengunduh CloudFormation templat, lalu unggah templat di [CloudFormation konsol](#) untuk membuat CloudFormation tumpukan baru. Untuk informasi selengkapnya, lihat [Menggunakan CloudFormation konsol](#) di Panduan AWS CloudFormation Pengguna.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-1t-efs-fsx1.yaml
```

Gunakan CloudFormation template untuk membuat template peluncuran EC2

Gunakan prosedur berikut untuk menyelesaikan CloudFormation template di CloudFormation konsol

- Di bawah Berikan nama tumpukan:
 - Di bawah nama Stack, masukkan `getstarted-1t`.
- Di bawah Parameter:
 - Di bawah Keamanan
 - Untuk `VpcSecurityGroupId`, pilih grup keamanan yang disebutkan default di VPC cluster Anda.
 - Untuk `ClusterSecurityGroupId`, pilih grup bernama `cluster-getstarted-sg`
 - Untuk `SshSecurityGroupId`, pilih grup bernama `inbound-ssh-getstarted-sg`
 - Untuk `SshKeyName`, pilih key pair SSH pilihan Anda.
 - Di bawah sistem File
 - Untuk `EfsFileSystemId`, masukkan ID sistem file dari sistem file EFS 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 [CloudFormation konsol](#). Pilih `getstarted-1t` 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 (instans EC2) yang diluncurkan dan dikelola AWS PCS. Saat menentukan grup node komputasi, Anda menentukan ciri umum seperti tipe instans EC2, jumlah instans minimum dan maksimum, subnet VPC target, opsi pembelian pilihan, dan konfigurasi peluncuran kustom. AWS PCS secara 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 [konsol AWS PCS](#) 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 — `Enterlogin`.
- Di bawah konfigurasi Komputasi, masukkan atau pilih nilai-nilai ini:
 - Template peluncuran EC2 - Pilih template peluncuran di mana namanya berada `login-getstarted-1t`
 - Profil instans IAM - Pilih profil contoh 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, masukkan1.
- Di bawah Pengaturan tambahan, tentukan yang berikut ini:
 - ID AMI - Pilih AMI yang ingin Anda gunakan, yang memiliki nama dalam format berikut:

```
aws-pcs-sample_ami-amzn2-platform-slurm-version
```

Untuk informasi lebih lanjut tentang sampel AMIs, lihat [Menggunakan sampel Amazon Machine Images \(AMIs\) dengan AWS PCS](#).

- 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 PCS AWS

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 [konsol AWS PCS](#) dan arahkan ke Cluster.
- Pilih klaster 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:
 - Template peluncuran EC2 - Pilih template peluncuran di mana namanya berada `compute-getstarted-1t`
 - Profil instans IAM - Pilih profil contoh 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:
 - ID AMI - Pilih AMI yang ingin Anda gunakan, yang memiliki nama dalam format berikut:

```
aws-pcs-sample_ami-amzn2-platform-slurm-version
```

Untuk informasi lebih lanjut tentang sampel AMIs, lihat [Menggunakan sampel Amazon Machine Images \(AMIs\) dengan AWS PCS](#).

- 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 [konsol AWS PCS](#).
- 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 klaster AWS PCS Anda

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

Untuk terhubung ke node login

- Buka [konsol AWS PCS](#) 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 [EC2 konsol 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 atau SSH.


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 klien SSH.
- 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 `sm-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 menggunakan SSH.

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

Bekerja dengan sistem file bersama

Anda dapat mengonfirmasi bahwa sistem file EFS 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:/z1shxbev  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 file EFS 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 yang dikembalikan darisbatch.

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

Example

```
squeue --job 1
```

squeuePerintah 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 selesai 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 pekerjaan MPI multi-node dengan Slurm di PCS AWS

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

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 modul OpenMpi.

```
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 isimulti.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 sumber daya AWS PCS yang Anda buat untuk tutorial ini

- Buka [konsol AWS PCS](#).
- Arahkan ke cluster bernama get-started.
- Arahkan ke bagian Antrian.
- Pilih antrian bernama demo.
- Pilih Hapus.

 Important


Tunggu hingga antrian 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 [konsol IAM](#).
 - Pilih Peran.
 - Pilih peran bernama AWSPCS-getstarted-role lalu pilih Delete.
 - Setelah peran dihapus, pilih Kebijakan.

- Pilih kebijakan bernama AWSPCS-getstarted-policy lalu pilih Delete.
- Buka [konsol CloudFormation](#).
 - Pilih tumpukan bernama getstarted-It.
 - Pilih Hapus.

 Important

Tunggu tumpukan dihapus sebelum melanjutkan.

- Buka [konsol Amazon EFS](#).
 - Pilih Sistem file.
 - Pilih sistem file bernama getstarted-efs.
 - Pilih Hapus.

 Important

Tunggu hingga sistem file dihapus sebelum melanjutkan.

- Buka [FSx konsol Amazon](#).
 - Pilih Sistem file.
 - Pilih sistem file bernama getstarted-fsx.
 - Pilih Hapus.

 Important

Tunggu hingga sistem file dihapus sebelum melanjutkan.

- Buka [konsol CloudFormation](#).
 - Pilih tumpukan bernama getstarted-sg.
 - Pilih Hapus.
- Buka [konsol CloudFormation](#).
 - Pilih tumpukan bernama hpc-networking.
 - Pilih Hapus.

Memulai dengan CloudFormation dan AWS PCS

Anda dapat menggunakan AWS CloudFormation untuk membuat cluster AWS PCS. CloudFormation memungkinkan Anda untuk membuat dan menyediakan penyebaran AWS infrastruktur yang dapat diprediksi dan berulang kali. Anda dapat menggunakan CloudFormation untuk secara otomatis menyediakan sumber daya dari banyak AWS layanan untuk membangun aplikasi yang sangat andal, terukur, dan hemat biaya AWS Cloud tanpa membuat dan mengonfigurasi infrastruktur yang mendasarinya. AWS CloudFormation memungkinkan Anda untuk menggunakan file template untuk membuat dan menghapus koleksi sumber daya bersama-sama sebagai satu unit, yang disebut tumpukan. Untuk informasi lebih lanjut tentang CloudFormation, lihat [Apa itu CloudFormation?](#) dalam AWS CloudFormation User Guide. Untuk informasi selengkapnya tentang jenis sumber daya AWS [AWS PCS CloudFormation](#), lihat [referensi jenis sumber daya PCS](#) di Panduan AWS CloudFormation Pengguna.

Topik

- [Gunakan CloudFormation untuk membuat contoh cluster AWS PCS](#)
- [Connect ke cluster AWS PCS yang dibuat dengan CloudFormation](#)
- [Bersihkan cluster AWS PCS di CloudFormation](#)
- [Bagian dari CloudFormation template untuk AWS PCS](#)
- [CloudFormation template untuk membuat contoh cluster AWS PCS](#)


Gunakan CloudFormation untuk membuat contoh cluster AWS PCS

Prosedur berikut menggunakan CloudFormation template dalam Konsol Manajemen AWS untuk membuat contoh AWS PCS cluster. Untuk informasi lebih lanjut tentang CloudFormation, lihat [Apa itu CloudFormation?](#) dalam AWS CloudFormation User Guide. Untuk informasi selengkapnya tentang jenis sumber daya AWS [AWS PCS CloudFormation](#), lihat [referensi jenis sumber daya PCS](#) di Panduan AWS CloudFormation Pengguna.

Untuk membuat cluster sampel

1. Pilih AWS Region untuk membuat cluster di (link membuka CloudFormation konsol dengan template):
 - [AS Timur \(Virginia N.\) \(us-east-1\)](#)
 - [AS Timur \(Ohio\) \(us-east-2\)](#)

- [AS Barat \(Oregon\) \(us-west-2\)](#)
 - [Asia Pasifik \(Singapura\) \(ap-southeast-1\)](#)
 - [Asia Pasifik \(Sydney\) \(ap-southeast-2\)](#)
 - [Asia Pasifik \(Tokyo\) \(ap-northeast-1\)](#)
 - [Eropa \(Frankfurt\) \(eu-central-1\)](#)
 - [Eropa \(Irlandia\) \(eu-west-1\)](#)
 - [Eropa \(London\) \(eu-west-2\)](#)
 - [Eropa \(Stockholm\) \(eu-north-1\)](#)
 - [AWS GovCloud \(AS-Timur\) \(us-gov-east-1\)](#)
 - [AWS GovCloud \(AS-Barat\) \(us-gov-west-1\)](#)
2. Di bawah Berikan nama tumpukan, masukkan nama deskriptif. Ini adalah nama untuk CloudFormation tumpukan Anda. Template menggunakan nilai ini sebagai nama untuk cluster AWS PCS Anda.
 3. Di bawah Parameter:
 - a. Di bawah SlurmVersion, pilih versi Slurm yang ingin digunakan cluster Anda.
 - b. Di bawah NodeArchitecture, pilih x86 untuk menerapkan cluster yang menggunakan instance x86_64-kompatibel, atau pilih Graviton untuk menggunakan instance Arm64.
 - c. Untuk KeyName, pilih key pair SSH untuk mengakses node login cluster. Pastikan Anda memiliki file PEM untuk key pair yang Anda pilih.
 - d. Untuk ClientIpCidr, masukkan rentang IP dalam format CIDR untuk mengontrol akses ke node login.

 Warning

Nilai default `0.0.0.0/0` memungkinkan akses dari semua alamat IP.

- e. Biarkan nilai untuk HpcRecipesS3Bucket dan HpcRecipesBranch sebagai nilai defaultnya.

4. Di bawah Kemampuan dan transformasi:

- a. Pilih kotak centang untuk mengakui bahwa CloudFormation akan membuat sumber daya IAM.
- b. Pilih kotak centang untuk mengakui bahwa CloudFormation akan membuat sumber daya

~~IAM dengan nama khusus.~~

- c. Pilih kotak centang `CAPABILITY_AUTO_EXPAND` untuk mengakui tumpukan baru. Untuk informasi selengkapnya, lihat [CreateStack](#) di dalam Referensi API AWS CloudFormation .
5. Pilih Buat tumpukan.
6. Pantau status tumpukan Anda. Anda dapat terhubung ke cluster setelah status tumpukan `CREATE_COMPLETE`.

Connect ke cluster AWS PCS yang dibuat dengan CloudFormation

Setelah Anda membuat cluster AWS PCS dari CloudFormation template, Anda dapat menggunakan konsol AWS PCS (dalam Konsol Manajemen AWS) untuk mengelola cluster. Anda juga dapat terhubung ke 1 node login cluster untuk mengelola cluster, menjalankan pekerjaan, dan mengelola data. CloudFormation Tumpukan menyediakan tautan yang dapat Anda gunakan untuk terhubung ke cluster Anda.

Untuk terhubung ke cluster Anda

1. Buka [konsol CloudFormation](#)
2. Pilih tumpukan yang Anda buat.
3. Pilih tab Output dari tumpukan.

Tumpukan menyediakan tautan berikut:

- `PcsConsoleUrl`— Pilih tautan ini untuk membuka konsol AWS PCS dengan cluster yang dipilih. Anda dapat menggunakannya untuk menjelajahi konfigurasi cluster, grup node, dan antrian.
- `Ec2 ConsoleUrl` — Pilih tautan ini untuk membuka konsol Amazon EC2, difilter untuk menampilkan instance yang dikelola grup node login cluster.

Dari tampilan ini, Anda dapat memilih instance dan memilih Connect. Contoh cluster mendukung SSH masuk dan AWS Systems Manager koneksi di browser web. Untuk informasi selengkapnya, lihat [Connect ke klaster AWS PCS Anda](#).

Setelah Anda terhubung ke instance login, Anda dapat mengikuti tutorial di [Jelajahi lingkungan cluster di AWS PCS](#).

Bersihkan cluster AWS PCS di CloudFormation

Jika Anda biasa CloudFormation membuat cluster AWS PCS Anda, Anda dapat membuka [CloudFormation konsol](#) dan menghapus tumpukan untuk menghapus cluster dan semua sumber daya terkait.

Important

Untuk cluster sampel, jika Anda membuat grup node komputasi tambahan atau antrian di klaster Anda (di luar login dan compute-1 grup yang dibuat CloudFormation templat sampel), Anda harus menggunakan [konsol AWS PCS](#) atau AWS CLI menghapus sumber daya tersebut sebelum menghapus tumpukan. CloudFormation Untuk informasi selengkapnya, lihat [Menghapus cluster di AWS PCS](#).

Bagian dari CloudFormation template untuk AWS PCS

CloudFormation Template memiliki 1 atau lebih bagian yang masing-masing melayani tujuan tertentu. CloudFormation mendefinisikan format standar, sintaks, dan bahasa dalam template. Untuk informasi selengkapnya, lihat [Bekerja dengan CloudFormation templat](#) di Panduan AWS CloudFormation Pengguna.

CloudFormation template sangat dapat disesuaikan dan oleh karena itu formatnya dapat bervariasi. Untuk memahami bagian yang diperlukan dari CloudFormation template untuk membuat cluster AWS PCS, kami sarankan Anda memeriksa template sampel yang kami sediakan untuk membuat cluster sampel. Topik ini menjelaskan secara singkat bagian-bagian dari template sampel itu.

Important

Contoh kode dalam topik ini tidak lengkap. Kehadiran ellipsis ([. . .]) menunjukkan bahwa ada kode tambahan yang tidak ditampilkan. Untuk mengunduh CloudFormation template berformat YAML lengkap, lihat. [CloudFormation template untuk membuat contoh cluster AWS PCS](#)

Daftar Isi

- [Header](#)
- [Metadata](#)

- [Parameter](#)
- [Pemetaan](#)
- [Sumber Daya](#)
- [Output](#)

Header

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: AWS Parallel Computing Service "getting started" cluster
```

`AWSTemplateFormatVersion` mengidentifikasi versi format template yang sesuai dengan template. Untuk informasi selengkapnya, lihat [sintaks versi format CloudFormation templat](#) di Panduan AWS CloudFormation Pengguna.

`Transform` menentukan makro yang CloudFormation menggunakan untuk memproses template. Untuk informasi selengkapnya, lihat [bagian Transform CloudFormation template](#) di Panduan AWS CloudFormation Pengguna. `AWS::Serverless-2016-10-31` Transformasi memungkinkan CloudFormation untuk memproses template yang ditulis dalam sintaks AWS Serverless Application Model (AWS SAM). Untuk informasi selengkapnya, lihat [AWS::Serverless mengubah](#) dalam Panduan AWS CloudFormation Pengguna.

Metadata

```
### Stack metadata
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: PCS Cluster configuration
        Parameters:
          - SlurmVersion
          - ManagedAccounting
          - AccountingPolicyEnforcement
      - Label:
          default: PCS ComputeNodeGroups configuration
        Parameters:
          - NodeArchitecture
          - KeyName
```

```

- ClientIpCidr
- Label:
  default: HPC Recipes configuration
Parameters:
  - HpcRecipesS3Bucket
  - HpcRecipesBranch

```

metadataBagian CloudFormation template memberikan informasi tentang template itu sendiri. Template sampel membuat cluster komputasi kinerja tinggi (HPC) lengkap yang menggunakan AWS PCS. Bagian metadata dari template sampel mendeklarasikan parameter yang mengontrol cara CloudFormation meluncurkan (ketentuan) tumpukan yang sesuai. Ada parameter yang mengontrol pilihan arsitektur (`NodeArchitecture`), versi Slurm (`SlurmVersion`), dan kontrol akses (`KeyName` dan `ClientIpCidr`).

Parameter

ParametersBagian ini mendefinisikan parameter kustom untuk template. CloudFormation menggunakan definisi parameter ini untuk membangun dan memvalidasi formulir yang berinteraksi dengan Anda ketika Anda meluncurkan tumpukan dari template ini.

```

Parameters:

NodeArchitecture:
  Type: String
  Default: x86
  AllowedValues:
    - x86
    - Graviton
  Description: Processor architecture for the login and compute node instances

SlurmVersion:
  Type: String
  Default: 25.05
  Description: Version of Slurm to use
  AllowedValues:
    - 24.11
    - 25.05

ManagedAccounting:
  Type: String
  Default: 'disabled'
  AllowedValues:

```

- 'enabled'
- 'disabled'

Description: Monitor cluster usage, manage access control, and enforce resource limits with Slurm accounting. Requires Slurm 24.11 or newer.

AccountingPolicyEnforcement:

Description: Specify which Slurm accounting policies to enforce

Type: String

Default: none

AllowedValues:

- none
- 'associations,limits,safe'

KeyName:

Description: SSH keypair to log in to the head node

Type: AWS::EC2::KeyPair::KeyName

AllowedPattern: ".+" # Required

ClientIpCidr:

Description: IP(s) allowed to access the login node over SSH. We recommend that you restrict it with your own IP/subnet (x.x.x.x/32 for your own ip or x.x.x.x/24 for range. Replace x.x.x.x with your own PUBLIC IP. You can get your public IP using tools such as <https://ifconfig.co/>)

Default: 127.0.0.1/32

Type: String

AllowedPattern: (\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3})/(\d{1,2})

ConstraintDescription: Value must be a valid IP or network range of the form x.x.x.x/x.

HpcRecipesS3Bucket:

Type: String

Default: aws-hpc-recipes

Description: HPC Recipes for AWS S3 bucket

AllowedValues:

- aws-hpc-recipes
- aws-hpc-recipes-dev

HpcRecipesBranch:

Type: String

Default: main

Description: HPC Recipes for AWS release branch

AllowedPattern: '^(?!.*\/\.git\$)(?!.*\/\.)(?!.*\\.\.)[a-zA-Z0-9-_\.\.]+\$'

Pemetaan

MappingsBagian ini mendefinisikan pasangan kunci-nilai yang menentukan nilai berdasarkan kondisi atau dependensi tertentu.

Mappings:

```
Architecture:
  AmiArchParameter:
    Graviton: arm64
    x86: x86_64
  LoginNodeInstances:
    Graviton: c7g.xlarge
    x86: c6i.xlarge
  ComputeNodeInstances:
    Graviton: c7g.xlarge
    x86: c6i.xlarge
```

Sumber Daya

ResourcesBagian ini mendeklarasikan AWS sumber daya untuk menyediakan dan mengkonfigurasi sebagai bagian dari tumpukan.

Resources:

[...]

Template menyediakan infrastruktur cluster sampel dalam lapisan. Dimulai dengan Networking untuk konfigurasi VPC. Penyimpanan disediakan oleh sistem ganda: EfsStorage untuk penyimpanan bersama dan FSxLStorage untuk penyimpanan berkinerja tinggi. Cluster inti didirikan melaluiPCSCluster.

```
Networking:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      ProvisionSubnetsC: "False"
    TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/net/hpc_large_scale/assets/main.yaml'
```

```

EfsStorage:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      SubnetIds: !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
      SubnetCount: 1
      VpcId: !GetAtt [ Networking, Outputs.VPC ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/storage/efs_simple/assets/main.yaml'

FSxLStorage:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      PerUnitStorageThroughput: 125
      SubnetId: !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
      VpcId: !GetAtt [ Networking, Outputs.VPC ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/storage/fsx_lustre/assets/persistent.yaml'

[...]

# Cluster
PCSCluster:
  Type: AWS::PCS::Cluster
  Properties:
    Name: !Sub '${AWS::StackName}'
    Size: SMALL
    Scheduler:
      Type: SLURM
      Version: !Ref SlurmVersion
    Networking:
      SubnetIds:
        - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
      SecurityGroupIds:
        - !GetAtt [ PCSSecurityGroup, Outputs.ClusterSecurityGroupId ]

```

Untuk sumber daya komputasi, template membuat dua grup node: PCSNodeGroupLogin untuk satu node login dan PCSNodeGroupCompute hingga empat node komputasi. Grup node ini didukung oleh PCSInstanceProfile untuk izin dan PCSLaunchTemplate misalnya konfigurasi.

```

# Compute Node groups
PCSIInstanceProfile:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      # We have to regionalize this in case CX use the template in more than one
      region. Otherwise,
      # the create action will fail since instance-role-${AWS::StackName} already
      exists!
      RoleName: !Sub '${AWS::StackName}-${AWS::Region}'
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/pcs-iip-minimal.yaml'

PCSLaunchTemplate:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      VpcDefaultSecurityGroupId: !GetAtt [ Networking, Outputs.SecurityGroup ]
      ClusterSecurityGroupId: !GetAtt [ PCSSecurityGroup,
Outputs.ClusterSecurityGroupId ]
      SshSecurityGroupId: !GetAtt [ PCSSecurityGroup,
Outputs.InboundSshSecurityGroupId ]
      EfsFileSystemSecurityGroupId: !GetAtt [ EfsStorage, Outputs.SecurityGroupId ]
      FSxLustreFileSystemSecurityGroupId: !GetAtt [ FSxLStorage,
Outputs.FSxLustreSecurityGroupId ]
      SshKeyName: !Ref KeyName
      EfsFileSystemId: !GetAtt [ EfsStorage, Outputs.EFSFileSystemId ]
      FSxLustreFileSystemId: !GetAtt [ FSxLStorage, Outputs.FSxLustreFileSystemId ]
      FSxLustreFileSystemMountName: !GetAtt [ FSxLStorage,
Outputs.FSxLustreMountName ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/cfn-pcs-lt-efs-fsx1.yaml'

# Compute Node groups - Login Nodes
PCSNODEGROUPLogin:
  Type: AWS::PCS::ComputeNodeGroup
  Properties:
    ClusterId: !GetAtt [PCSCluster, Id]
    Name: login
    ScalingConfiguration:
      MinInstanceCount: 1
      MaxInstanceCount: 1
    IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
    CustomLaunchTemplate:

```

```

    TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.LoginLaunchTemplateId ]
    Version: 1
  SubnetIds:
    - !GetAtt [ Networking, Outputs.DefaultPublicSubnet ]
  AmiId: !GetAtt [PcsSampleAmi, AmiId]
  InstanceConfigs:
    - InstanceType: !FindInMap [ Architecture, LoginNodeInstances, !Ref
NodeArchitecture ]

# Compute Node groups - Compute Nodes
PCSNodeGroupCompute:
  Type: AWS::PCS::ComputeNodeGroup
  Properties:
    ClusterId: !GetAtt [PCSCluster, Id]
    Name: compute-1
    ScalingConfiguration:
      MinInstanceCount: 0
      MaxInstanceCount: 4
    IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
    CustomLaunchTemplate:
      TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.ComputeLaunchTemplateId ]
      Version: 1
    SubnetIds:
      - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
    AmiId: !GetAtt [PcsSampleAmi, AmiId]
    InstanceConfigs:
      - InstanceType: !FindInMap [ Architecture, ComputeNodeInstances, !Ref
NodeArchitecture ]

```

Penjadwalan Job ditangani melalui PCSQueueCompute

```

PCSQueueCompute:
  Type: AWS::PCS::Queue
  Properties:
    ClusterId: !GetAtt [PCSCluster, Id]
    Name: demo
    ComputeNodeGroupConfigurations:
      - ComputeNodeGroupId: !GetAtt [PCSNodeGroupCompute, Id]

```

Pemilihan AMI terjadi secara otomatis melalui fungsi Pcs AMILookup Fn Lambda dan sumber daya terkait.

```

PcsAMILookupRole:
  Type: AWS::IAM::Role
  [...]

PcsAMILookupFn:
  Type: AWS::Lambda::Function
  Properties:
    Runtime: python3.12
    Handler: index.handler
    Role: !GetAtt PcsAMILookupRole.Arn
    Code:
      [...]
    Timeout: 30
    MemorySize: 128

# Example of using the custom resource to look up an AMI
PcsSampleAmi:
  Type: Custom::AMILookup
  Properties:
    ServiceToken: !GetAtt PcsAMILookupFn.Arn
    OperatingSystem: 'amzn2'
    Architecture: !FindInMap [ Architecture, AmiArchParameter, !Ref
NodeArchitecture ]
    SlurmVersion: !Ref SlurmVersion

```

Output

Template menghasilkan identifikasi dan manajemen cluster URLs melalui `ClusterId`, `PcsConsoleUrl`, dan `Ec2ConsoleUrl`.

```

Outputs:
  ClusterId:
    Description: The Id of the PCS cluster
    Value: !GetAtt [ PCSCluster, Id ]

  PcsConsoleUrl:
    Description: URL to access the cluster in the PCS console
    Value: !Sub
      - https://${ConsoleDomain}/pcs/home?region=${AWS::Region}#/clusters/${ClusterId}
      - { ConsoleDomain: !If [ GovCloud, 'console.amazonaws-us-gov.com', !If [ China,
'console.amazonaws.cn', !Sub '${AWS::Region}.console.aws.amazon.com'] ],

```








```





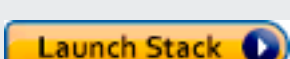
    ClusterId: !GetAtt [ PCSCluster, Id ]
  }
  Export:
    Name: !Sub ${AWS::StackName}-PcsConsoleUrl

  Ec2ConsoleUrl:
    Description: URL to access instance(s) in the login node group via Session Manager
    Value: !Sub
      - https://${ConsoleDomain}/ec2/home?region=
        ${AWS::Region}#Instances:instanceState=running;tag:aws:pcs:compute-node-group-id=
        ${NodeGroupLoginId}
      - { ConsoleDomain: !If [ GovCloud, 'console.amazonaws-us-gov.com', !If [ China,
        'console.amazonaws.cn', !Sub '${AWS::Region}.console.aws.amazon.com' ] ],
        NodeGroupLoginId: !GetAtt [ PCSNodeGroupLogin, Id ]
      }
  Export:
    Name: !Sub ${AWS::StackName}-Ec2ConsoleUrl

```

CloudFormation template untuk membuat contoh cluster AWS PCS

AWS Region nama	AWS Region	Lihat sumber	Luncurkan tumpukan
US East (Northern Virginia)	us-east-1	Unduh YAMAL	
AS Timur (Ohio)	us-east-2	Unduh YAMAL	
AS Barat (Oregon)	us-west-2	Unduh YAMAL	
Asia Pasifik (Singapura)	ap-southeast-1	Unduh YAMAL	
Asia Pasifik (Sydney)	ap-southeast-2	Unduh YAMAL	
Asia Pasifik (Tokyo)	ap-northeast-1	Unduh YAMAL	
Europa (Frankfurt)	eu-central-1	Unduh YAMAL	

AWS Region nama	AWS Region	Lihat sumber	Luncurkan tumpukan
Eropa (Irlandia)	eu-west-1	Unduh YAMAL	
Eropa (London)	eu-west-2	Unduh YAMAL	
Eropa (Stockholm)	eu-north-1	Unduh YAMAL	
AWS GovCloud (AS-Timur)	us-gov-east-1	Unduh YAMAL	
AWS GovCloud (AS-Barat)	us-gov-west-1	Unduh YAMAL	

AWS Cluster PCS

Cluster AWS PCS terdiri dari komponen-komponen berikut:

- Instans terkelola dari perangkat lunak penjadwal sistem HPC, seperti daemon kontrol Slurm (`slurmctld`).
- Komponen yang terintegrasi dengan penjadwal sistem HPC untuk menyediakan dan mengelola instans Amazon EC2.
- Komponen yang terintegrasi dengan penjadwal sistem HPC 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 PCS menyediakan antarmuka jaringan elastis di subnet VPC Amazon Anda untuk menyediakan konektivitas dari perangkat lunak penjadwal ke instans EC2 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 AWS PCS](#)
- [Memperbarui cluster di AWS PCS](#)
- [Menghapus cluster di AWS PCS](#)
- [Ukuran cluster dalam AWS PCS](#)
- [Bekerja dengan rahasia cluster di AWS PCS](#)

Membuat cluster di AWS PCS

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda membuat cluster di AWS Parallel Computing Service (AWS PCS). Jika ini adalah pertama kalinya Anda membuat cluster AWS PCS, kami sarankan Anda mengikuti [Memulai Layanan Komputasi AWS Paralel](#). Tutorial ini dapat membantu Anda membuat sistem HPC yang berfungsi tanpa memperluas ke semua opsi yang tersedia dan arsitektur sistem yang dimungkinkan.

Note

Setelah membuat cluster, Anda dapat memodifikasi banyak pengaturan konfigurasi tanpa membangun kembali infrastruktur Anda. Untuk informasi selengkapnya, lihat [Memperbarui cluster di AWS PCS](#).

Note

Anda dapat mengonfigurasi pengaturan Slurm khusus untuk menerapkan kebijakan penjadwalan lanjutan dan manajemen sumber daya. Untuk informasi selengkapnya, lihat [Mengkonfigurasi pengaturan Slurm khusus di PCS AWS](#).

Prasyarat

- VPC dan subnet yang ada yang memenuhi persyaratan. [AWS Jaringan PCS](#) Sebelum Anda menerapkan klaster untuk penggunaan produksi, kami sarankan Anda memiliki pemahaman menyeluruh tentang persyaratan VPC dan subnet. Untuk membuat VPC dan subnet, lihat [Membuat VPC untuk klaster PCS Anda AWS](#)
- [Prinsipal IAM](#) dengan izin untuk membuat dan mengelola sumber daya AWS PCS. Untuk informasi selengkapnya, lihat [Identity and Access Management untuk Layanan Komputasi AWS Paralel](#).

Buat cluster AWS PCS

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk membuat cluster.

Konsol Manajemen AWS

Untuk membuat klaster DB

1. Buka konsol AWS PCS 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 AWS Region dan Akun AWS tempat Anda membuat cluster.


- Scheduler — Pilih penjadwal dan versi. Untuk informasi selengkapnya, lihat [Versi slurm di PCS AWS](#).
- 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, lihat [Ukuran cluster dalam AWS PCS](#).

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

- Jenis jaringan — Pilih jenis alamat IP untuk cluster Anda. Cluster Anda dapat menggunakan salah satu IPv4 atau IPv6, tetapi tidak keduanya. VPC dan subnet harus menggunakan jenis alamat jaringan yang sama. Blok alamat IP yang Anda gunakan untuk setiap subnet harus memiliki setidaknya 1 alamat yang tersedia. AWS menyimpan beberapa alamat di setiap subnet. Untuk informasi selengkapnya, lihat [Blok CIDR Subnet](#) di Panduan Pengguna Amazon VPC.
- VPC — Pilih VPC yang sudah ada yang memenuhi persyaratan PCS. AWS Untuk informasi selengkapnya, lihat [AWS PCS VPC dan persyaratan subnet dan pertimbangan](#). Setelah Anda membuat cluster, Anda tidak dapat mengubah VPC-nya. Jika tidak VPCs terdaftar, Anda harus membuatnya terlebih dahulu.
- Subnet - Semua subnet yang tersedia di VPC yang dipilih terdaftar. Pilih subnet yang memenuhi persyaratan subnet AWS PCS. Untuk informasi selengkapnya, lihat [AWS PCS VPC dan persyaratan subnet dan pertimbangan](#). Kami menyarankan Anda memilih subnet pribadi untuk menghindari mengekspos endpoint scheduler Anda ke internet publik.
- Grup keamanan — Tentukan grup keamanan yang Anda ingin AWS PCS 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. Anda dapat memilih Cepat membuat grup keamanan agar AWS PCS membuatnya dengan konfigurasi yang diperlukan di VPC yang Anda pilih, atau pilih grup keamanan yang ada. Untuk informasi selengkapnya, lihat [Persyaratan dan pertimbangan kelompok keamanan](#).

4. (Opsional) Di bagian konfigurasi akuntansi Slurm, Anda dapat mengaktifkan akuntansi Slurm dan mengatur parameter akuntansi. Untuk informasi selengkapnya, lihat [Akuntansi slurm di PCS AWS](#).

5. (Opsional) Di bagian konfigurasi Slurm, Anda dapat menambahkan nama parameter dan pasangan nilai untuk mengonfigurasi pengaturan Slurm tambahan. Untuk daftar lengkap parameter yang didukung, lihat [Pengaturan Slurm khusus untuk AWS cluster PCS](#).
6. (Opsional) Di bawah Tag, tambahkan tag apa pun ke cluster AWS PCS 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 AWS Region per Akun AWS. AWS PCS mengembalikan 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 AWS Region Anda ingin membuat cluster Anda, seperti `us-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 AWS Region dan Akun AWS di mana Anda membuat cluster.
 - Ganti *25.05* dengan versi Slurm yang didukung.

 Note

AWS PCS saat ini mendukung Slurm 25.05 dan 24.11.

- 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, lihat [Ukuran cluster dalam AWS PCS](#).


- 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 Anda ingin AWS PCS kaitkan dengan antarmuka jaringan yang dibuatnya untuk cluster Anda. Grup keamanan harus berada di VPC yang 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](#).

```
aws pcs create-cluster --region region \  
  --cluster-name my-cluster \  
  --scheduler type=SLURM,version=25.05 \  
  --size SMALL \  
  --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

- untuk digunakan IPv6, tambahkan networkType=IPV6 ke --networking konfigurasi.

```
--networking networkType=IPV6,subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

- Secara opsional, Anda dapat menambahkan --slurm-configuration opsi untuk menyesuaikan perilaku Slurm dan menentukan opsi konfigurasi Slurm. Contoh berikut menetapkan waktu idle scale-down menjadi 60 menit (3600 detik), memungkinkan akuntansi Slurm, dan menentukan slurm.conf pengaturan sebagai nilai untuk slurmCustomSettings Untuk informasi selengkapnya, lihat [Akuntansi slurm di PCS AWS](#).

 Note

Akuntansi didukung untuk Slurm 24.11 atau yang lebih baru.

```
aws pcs create-cluster --region region \  
  --cluster-name my-cluster \  
  --scheduler type=SLURM,version=25.05 \  
  --size SMALL \  
  --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

```
--slurm-configuration  
scaleDownIdleTimeInSeconds=3600,accounting='{mode=STANDARD}',slurmCustomSettings='{p
```

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

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

Important

Hanya ada 1 cluster dalam satu Creating keadaan per AWS Region per Akun AWS. AWS PCS mengembalikan kesalahan jika sudah ada cluster dalam Creating keadaan ketika Anda mencoba membuat cluster.

Langkah selanjutnya yang disarankan untuk kluster Anda

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

Memperbarui cluster di AWS PCS

AWS PCS memungkinkan Anda memperbarui konfigurasi cluster setelah pembuatan melalui UpdateCluster API atau konsol. Anda dapat memodifikasi pengaturan kluster tanpa membangun kembali infrastruktur Anda, yang mengurangi overhead operasional dan meminimalkan interupsi.

Manfaat pembaruan kluster

Memperbarui kluster AWS PCS memungkinkan Anda menyesuaikan infrastruktur HPC dengan persyaratan baru tanpa gangguan layanan. Perubahan konfigurasi membutuhkan waktu beberapa menit, bukan jam atau lebih yang dibutuhkan untuk membangun kembali cluster. Kemampuan ini penting untuk lingkungan produksi yang memerlukan waktu henti minimal dan untuk tim yang perlu menyesuaikan pengaturan cluster saat pola beban kerja berubah.

Perubahan konfigurasi yang didukung

Anda dapat memodifikasi tiga kategori utama pengaturan:

- Konfigurasi akuntansi - Aktifkan atau nonaktifkan akuntansi terkelola dan konfigurasi pengaturan retensi.
- Perilaku penurunan skala - Sesuaikan `scaleDownIdleTime` parameter, yang mengontrol berapa lama instance dinamis tetap mengganggu sebelum AWS PCS secara otomatis menghentikannya.
- Pengaturan kustom slurm - Ubah pengaturan Slurm yang didukung yang berlaku di tingkat cluster, termasuk Prolog, Epilog, dan `SelectTypeParameters`

Batasan

Anda tidak dapat mengubah konfigurasi tertentu setelah pembuatan klaster. Ini termasuk:

- Konfigurasi grup keamanan
- Pemilihan subnet VPC
- Ukuran cluster
- Versi slurm
- Nama klaster

Pengaturan ini merupakan dasar arsitektur cluster dan memerlukan pembuatan cluster baru untuk memodifikasinya.

Prasyarat untuk pembaruan cluster

Sebelum memperbarui cluster, pastikan kondisi berikut terpenuhi:

- Cluster harus dalam `ACTIVE`, `UPDATE_FAILED`, atau `SUSPENDED` negara
- Semua sumber daya terkait (Antrian, Grup Node Komputasi) harus dalam status `ACTIVE`
- Anda harus memiliki izin IAM yang sesuai untuk operasi `UpdateCluster`
- Tidak ada operasi pembaruan lain yang dapat berlangsung

Perbarui proses dan dampak pekerjaan

Selama operasi pembaruan, node komputasi terus menjalankan pekerjaan yang ada bahkan ketika pengontrol cluster menjadi tidak dapat dijangkau sebentar. Namun, sistem tidak dapat menerima pengajuan pekerjaan baru atau membuat keputusan penjadwalan selama periode ini.

Anda dapat memantau pembaruan cluster melalui antarmuka konsol dan API. Cluster akan bertransisi melalui status berikut selama pembaruan:

- UPDATING- Pembaruan sedang berlangsung
- ACTIVE- Update selesai berhasil
- UPDATE_FAILED- Pembaruan mengalami kesalahan

Penagihan selama pembaruan

Biaya per jam standar untuk klaster AWS PCS Anda berlanjut selama operasi pembaruan. Saat Anda memperbarui klaster untuk menonaktifkan akuntansi, penagihan untuk fitur akuntansi akan berhenti segera setelah klaster memasuki UPDATING status. Saat mengaktifkan akuntansi, penagihan tidak dimulai sampai cluster berhasil menyelesaikan pembaruan dan kembali ke status. ACTIVE

Topik

- [Perbarui klaster AWS PCS](#)
- [Pertanyaan yang sering diajukan tentang memperbarui cluster di AWS PCS](#)
- [Memecahkan masalah pembaruan klaster AWS PCS](#)

Perbarui klaster AWS PCS

Gunakan langkah-langkah ini untuk mengubah pengaturan penjadwal, konfigurasi akuntansi, dan pengaturan kustom Slurm di cluster Anda. Untuk informasi selengkapnya, lihat [Pengaturan Slurm khusus untuk AWS cluster PCS](#).

Prasyarat

- Cluster harus dalam ACTIVE, UPDATE_FAILED, atau SUSPENDED negara
- Semua sumber daya terkait (Antrian, Grup Node Komputasi) harus dalam status ACTIVE
- Tidak ada operasi pembaruan lain yang dapat berlangsung

Prosedur

Konsol Manajemen AWS

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Pada panel navigasi, silakan pilih Klaster.
3. Pilih cluster yang akan diperbarui.
4. Pilih Edit.
5. Pada halaman Edit cluster, ubah pengaturan yang diinginkan:
 - Di bawah konfigurasi Scheduler, perbarui waktu idle Scale-down untuk mengontrol berapa lama instans dinamis tetap idle sebelum penghentian otomatis.
 - Ubah pengaturan parameter Prolog, Epilog, dan Select-type sesuai kebutuhan.
 - Mengaktifkan, menonaktifkan, atau mengonfigurasi waktu retensi untuk akuntansi terkelola.
 - Di bawah Pengaturan penjadwal tambahan, tambahkan, edit, atau hapus pengaturan kustom Slurm. Untuk informasi selengkapnya tentang parameter yang didukung, lihat [Pengaturan Slurm khusus untuk AWS cluster PCS](#).

Note

Bidang yang tidak dapat diedit ditampilkan sebagai hanya-baca dan menunjukkan nilainya saat ini.

6. Pilih Perbarui untuk mengirimkan perubahan.
7. Pantau status cluster, yang ditampilkan sebagai “Memperbarui” selama proses. Status berubah ketika pembaruan selesai dengan sukses.

AWS CLI

1. Buka terminal atau command prompt.
2. Verifikasi status cluster menggunakan perintah berikut:

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

3. Kirim permintaan pembaruan menggunakan salah satu contoh berikut:

- Untuk mengaktifkan akuntansi terkelola:

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration 'accounting={mode=STANDARD}'
```

- Untuk memperbarui pengaturan Slurm Prolog:

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/  
prolog.sh"}]'
```

- Untuk memperbarui waktu idle scale-down:

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration 'scaleDownIdleTimeInSeconds=300'
```

4. Pantau kemajuan pembaruan dengan memeriksa status klaster:

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

Setelah permintaan pembaruan berhasil, perintah mengembalikan objek Cluster dengan semua perubahan. Status klaster berubah dari UPDATING ACTIVE saat selesai.

Pertanyaan yang sering diajukan tentang memperbarui cluster di AWS PCS

Dapatkan jawaban atas pertanyaan umum tentang memperbarui konfigurasi cluster di AWS PCS.

Pengaturan apa yang bisa saya modifikasi?

Anda dapat memodifikasi konfigurasi akuntansi (mengaktifkan/menonaktifkan akuntansi terkelola), perilaku penskalaan (Parameter `scaleDownIdle` waktu), dan setelan kustom Slurm yang didukung yang berlaku di tingkat cluster. Anda tidak dapat mengubah grup keamanan, subnet VPC, ukuran cluster, versi Slurm, atau nama cluster.

Bisakah saya mengantri beberapa pembaruan?

Tidak. Anda harus menunggu cluster kembali ke ACTIVE status sebelum mengirimkan pembaruan lain. Semua sumber daya terkait (Antrian, Grup Node Komputasi) juga harus dalam status. ACTIVE

Bisakah saya membatalkan operasi pembaruan cluster?

Tidak, Anda tidak dapat membatalkan operasi pembaruan klaster yang sedang berlangsung.

Bisakah saya mengirimkan pekerjaan saat klaster saya diperbarui?

Kami menyarankan Anda menghindari mengirimkan pekerjaan selama pembaruan klaster. Pengontrol Slurm mungkin tidak tersedia selama proses pembaruan.

Apakah pekerjaan saya akan terus berjalan selama pembaruan klaster?

Ya, menjalankan pekerjaan terus dijalankan pada node komputasi bahkan ketika pengontrol cluster menjadi tidak dapat dijangkau sebentar selama proses pembaruan. Namun, status pekerjaan mungkin tidak diperbarui sampai pengontrol tersedia lagi.

Bagaimana penagihan terpengaruh selama pembaruan?

Biaya per jam standar berlanjut selama operasi pembaruan. Saat menonaktifkan akuntansi, penagihan berhenti ketika cluster memasuki status. UPDATING Saat mengaktifkan akuntansi, penagihan dimulai ketika cluster berhasil kembali ke ACTIVE status.

Memecahkan masalah pembaruan klaster AWS PCS

Topik ini membantu Anda mengidentifikasi dan menyelesaikan masalah umum yang dapat terjadi saat memperbarui konfigurasi klaster.

Pembaruan gagal dengan kesalahan konfigurasi akuntansi

Penyebab umum

Cluster memasuki UPDATE_FAILED status dan pesan kesalahan menunjukkan masalah konfigurasi akuntansi. Ini biasanya terjadi ketika konfigurasi akuntansi tidak kompatibel dengan versi Slurm saat ini atau berisi pengaturan yang tidak valid.

Resolusi

Tinjau pengaturan akuntansi Anda untuk kompatibilitas dengan versi Slurm klaster Anda dan kirimkan permintaan pembaruan yang dikoreksi dengan parameter konfigurasi yang valid.

Pembaruan gagal dengan kesalahan pengaturan khusus

Penyebab umum

Cluster memasuki UPDATE_FAILED status dan pesan kesalahan menunjukkan masalah pengaturan kustom Slurm. Ini terjadi ketika Anda memberikan nilai parameter Slurm yang tidak valid atau kombinasi parameter yang tidak didukung.

Resolusi

Validasi pengaturan kustom Slurm Anda terhadap parameter yang didukung dan kirimkan permintaan pembaruan yang dikoreksi dengan nilai dan kombinasi parameter yang valid.

Tidak dapat mengirimkan permintaan pembaruan

Penyebab umum

Tombol pembaruan dinonaktifkan di konsol atau API mengembalikan kesalahan 400 tingkat. Ini terjadi ketika klaster tidak dalam keadaan yang sesuai, sumber daya terkait tidak aktif, atau ada kegagalan validasi dalam konfigurasi Anda.

Resolusi

Tunggu klaster dan semua sumber daya terkait mencapai ACTIVE status, lalu tinjau konfigurasi Anda untuk kesalahan validasi sebelum mengirimkan kembali permintaan pembaruan.

Kesalahan validasi

Penyebab umum

Perintah segera kembali dengan kesalahan HTTP 400 tingkat dan pesan deskriptif. Ini terjadi karena status cluster, status sumber daya, atau parameter konfigurasi yang tidak valid.

Resolusi

Atasi kesalahan validasi spesifik yang disebutkan dalam respons dan coba lagi operasi pembaruan.

Menghapus cluster di AWS PCS

Topik ini memberikan ikhtisar tentang cara menghapus klaster AWS PCS.

Pertimbangan saat menghapus cluster PCS AWS

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

Hapus cluster

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk menghapus cluster.

Konsol Manajemen AWS

Untuk menghapus klaster

1. Buka [konsol AWS PCS](#).
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 AWS Region 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
```

Ukuran cluster dalam AWS PCS

AWS PCS menyediakan 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

Important

Anda tidak dapat mengubah ukuran cluster setelah membuat cluster. Jika Anda perlu mengubah ukuran, Anda harus membuat cluster baru.

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 menciptakan rahasia cluster yang diperlukan untuk terhubung ke job scheduler di cluster. Anda juga membuat grup node komputasi AWS PCS, yang menentukan kumpulan instance yang akan diluncurkan sebagai respons terhadap peristiwa penskalaan. AWS PCS mengonfigurasi 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 PCS menyimpan rahasia cluster sebagai [rahasia terkelola](#) dengan awalan pcs! di AWS Secrets Manager. Biaya rahasia sudah termasuk dalam biaya untuk menggunakan AWS PCS. Anda dapat memutar rahasia cluster AWS Secrets Manager untuk menjaga kepatuhan keamanan dan memulihkan potensi kompromi keamanan.

Topik

- [Gunakan AWS Secrets Manager untuk menemukan rahasia cluster](#)
- [Gunakan AWS PCS untuk menemukan rahasia cluster](#)
- [Dapatkan rahasia cluster Slurm](#)
- [Memutar rahasia cluster di AWS PCS](#)

Gunakan AWS Secrets Manager untuk menemukan rahasia cluster

Konsol Manajemen AWS

1. Arahkan ke [konsol Secrets Manager](#).
2. Pilih Rahasia, lalu cari pcs! awalan.

Note

Rahasia cluster AWS PCS memiliki nama dalam bentuk di pcs!slurm-secret-*cluster-id* mana *cluster-id* adalah ID cluster AWS PCS.

AWS CLI

Setiap rahasia cluster AWS PCS 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 AWS Region untuk membuat cluster Anda, seperti `us-east-1`.
- Ganti *cluster-id* dengan ID cluster AWS PCS untuk menemukan rahasia cluster.

```
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 ARN untuk rahasia cluster AWS PCS. Masukkan perintah berikut, buat substitusi berikut:

- Ganti *region* dengan AWS Region untuk membuat cluster Anda, seperti `us-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": "get-started",  
    "id": "pcs_123456abcd",  
    "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",  
    "status": "ACTIVE",  
    "createdAt": "2024-12-17T21:03:52+00:00",  
    "modifiedAt": "2024-12-17T21:03:52+00:00",  
    "scheduler": {  
      "type": "SLURM",
```

```

    "version": "25.05"
  },
  "size": "SMALL",
  "slurmConfiguration": {
    "authKey": {
      "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!
slurm-secret-pcs_123456abcd-a12ABC",
      "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
    }
  },
  "networking": {
    "subnetIds": [
      "subnet-0123456789abcdef0"
    ],
    "securityGroupIds": [
      "sg-0123456789abcdef0"
    ]
  },
  "endpoints": [
    {
      "type": "SLURMCTLD",
      "privateIpAddress": "10.3.149.220",
      "port": "6817"
    }
  ]
}

```

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 AWS Region untuk membuat cluster Anda, seperti *us-east-1*.
- Ganti *secret-arn* dengan cluster secretArn from AWS PCS.

```

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 kepala sekolah IAM untuk mendapatkan rahasia cluster Slurm. Kepala IAM harus memiliki izin untuk membaca rahasianya. Untuk informasi selengkapnya, lihat [Istilah dan konsep peran](#) di Panduan AWS Identity and Access Management Pengguna.

Contoh kebijakan IAM 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"  
    }  
  ]  
}
```

Memutar rahasia cluster di AWS PCS

Gunakan Rotasi AWS Secrets Manager Terkelola untuk memutar rahasia cluster di AWS PCS. Rotasi rahasia reguler adalah praktik terbaik keamanan untuk mempertahankan postur keamanan yang kuat di lingkungan HPC. Kemampuan ini memungkinkan Anda untuk memenuhi standar kepatuhan industri termasuk HIPAA dan FedRAMP, yang mengamankan rotasi kredensi reguler.

Rahasia cluster melayani tujuan ganda: mengautentikasi node komputasi yang bergabung dengan cluster dan sebagai kunci JWT untuk otentikasi Slurm REST API. Ketika diputar, kedua aspek terpengaruh secara bersamaan.

Cara kerja rotasi rahasia cluster

Bersiaplah secara manual untuk menjaga stabilitas cluster selama rotasi rahasia:

1. Persiapan - Skala semua grup node komputasi ke kapasitas 0 dan pastikan tidak ada pekerjaan yang berjalan
2. Rotasi - Memulai rotasi melalui konsol Secrets Manager atau API
3. Monitoring - Melacak kemajuan melalui CloudTrail acara
4. Pemulihan — Skala grup node komputasi kembali ke kapasitas yang diinginkan

Selama rotasi, kluster Anda tetap dalam ACTIVE status dan penagihan berlanjut secara normal. Prosesnya biasanya memakan waktu beberapa menit.

Persyaratan dan pembatasan

Sebelum memutar rahasia cluster, lengkapi persyaratan ini:

- Cluster harus dalam ACTIVE atau UPDATE_FAILED negara
- Peran IAM harus memiliki izin `secretsmanager:RotateSecret`
- Semua grup node komputasi harus diskalakan ke kapasitas 0
- Hentikan semua pekerjaan sebelum rotasi

Pembatasan:

- Persiapan manual diperlukan untuk setiap rotasi
- Token JWT yang ada menjadi tidak valid dan memerlukan penerbitan kembali
- Node login BYO memerlukan pembaruan rahasia manual setelah rotasi

Topik

- [Putar rahasia cluster di AWS PCS](#)
- [Pertanyaan yang sering diajukan tentang rotasi rahasia cluster di AWS PCS](#)
- [Memecahkan masalah rotasi rahasia cluster di PCS AWS](#)

Putar rahasia cluster di AWS PCS

Putar rahasia klaster Anda untuk mematuhi persyaratan keamanan dan mengatasi potensi kompromi. Proses ini membutuhkan penempatan cluster Anda ke mode pemeliharaan.

Prasyarat

- Peran IAM dengan izin `secretsmanager:RotateSecret`
- Cluster di ACTIVE atau UPDATE_FAILED negara bagian

Prosedur

1. Beri tahu pengguna cluster tentang jendela pemeliharaan yang akan datang.
2. Masukkan cluster ke mode pemeliharaan dengan menskalakan semua grup node komputasi ke kapasitas 0.
 - a. Gunakan `UpdateComputeNodeGroup` API untuk menyetel keduanya `minInstanceCount` dan `maxInstanceCount` ke 0 untuk semua grup node komputasi.
 - b. Tunggu sampai semua node berhenti.
 - c. Opsional: Kuras antrian penjadwal dengan perintah Slurm sebelum Anda menghentikan kapasitas untuk penanganan pekerjaan yang anggun.
3. Memulai rotasi melalui Secrets Manager.
 - Metode konsol:
 - Arahkan ke Secrets Manager, pilih rahasia klaster Anda, dan pilih Rotate secret.
 - Metode API:
 - Gunakan Secrets Manager `rotate-secret` API.
4. Pantau kemajuan rotasi.
 - a. Lacak kemajuan melalui CloudTrail acara.
 - b. Periksa `lastRotatedDate` melalui konsol Secrets Manager atau `secretsmanager:describeSecret` API.
 - c. Tunggu `RotationSucceeded` atau `RotationFailed` CloudTrail acara.
5. Setelah rotasi berhasil, kembalikan kapasitas cluster.

- a. Gunakan UpdateComputeNodeGroup API untuk mengatur ulang grup node ke min/max kapasitas yang diinginkan.
- b. Untuk node login AWS yang dikelola PC: Tidak diperlukan tindakan tambahan.
- c. Untuk node login BYO:
 - i. Connect ke node login.
 - ii. Perbarui `/etc/slurm/slurm.key` dengan rahasia baru dari Secrets Manager.
 - iii. Mulai ulang Slurm Auth dan Cred Kiosk Daemon (karung).

Pertanyaan yang sering diajukan tentang rotasi rahasia cluster di AWS PCS

Temukan jawaban atas pertanyaan umum tentang rotasi rahasia cluster di AWS PCS.

Apa itu rahasia cluster?

Rahasia cluster adalah kredensi aman yang memungkinkan komunikasi aman antara pengontrol Slurm dan node komputasi AWS PCS. Ini juga berfungsi sebagai kunci JSON Web Token (JWT) untuk otentikasi Slurm REST API.

Apa perbedaan antara rahasia cluster dan kunci JWT?

Di AWS PCS, rahasia cluster dan kunci JWT adalah sumber daya yang sama yang melayani tujuan yang berbeda. Rahasia cluster mengotentikasi komunikasi internal Slurm, sedangkan kunci JWT menandatangani token untuk otentikasi REST API. Ketika diputar, kedua aspek terpengaruh secara bersamaan.

Berapa lama waktu rotasi?

Proses rotasi biasanya memakan waktu beberapa menit. Cluster Anda tetap dalam status AKTIF dan penagihan berlanjut secara normal selama rotasi.

Bisakah saya menjadwalkan rotasi otomatis?

Anda dapat mengaktifkan rotasi terjadwal di Secrets Manager. Namun, rilis awal memerlukan persiapan manual (penskalaan kelompok node ke 0) sebelum setiap rotasi.

Apakah token JWT saya yang ada masih berfungsi setelah rotasi?

Tidak, token JWT yang ada menjadi tidak valid setelah rotasi. Keluarkan token baru untuk klien REST API.

Di mana saya dapat menemukan rahasia cluster saya?

Anda dapat menemukan rahasia cluster Anda di konsol Secrets Manager atau melalui konsol AWS PCS. Untuk instruksi terperinci, lihat [Gunakan AWS Secrets Manager untuk menemukan rahasia cluster](#) dan [Gunakan AWS PCS untuk menemukan rahasia cluster](#).

Mengapa rotasi memerlukan penskalaan grup node ke 0?

Rotasi tidak memerlukan instance yang berjalan untuk memastikan stabilitas cluster selama proses pembaruan rahasia. Ini mencegah konflik otentikasi antara rahasia lama dan baru.

Persyaratan kepatuhan apa yang didukung fitur ini?

Fitur ini memungkinkan AWS PCS untuk memenuhi standar kepatuhan industri termasuk HIPAA dan FedRAMP, yang mengamankan rotasi kredensial reguler sebagai bagian dari kontrol keamanan mereka.

Memecahkan masalah rotasi rahasia cluster di PCS AWS

Rotasi rahasia cluster gagal jika lingkungan tidak disiapkan dengan benar. Penyebab paling umum adalah instance aktif di cluster Anda. Untuk mencegah kegagalan:

1. Atur semua grup node ke kapasitas 0.
2. Tunggu node berhenti.
3. Verifikasi klaster Anda tidak dalam status berikut:
`CREATE_FAILED`, `DELETE_FAILED`, `RESUMING`, `SUSPENDING`, atau `SUSPENDED`.

Jika rotasi gagal:

- Sebuah `RotationFailed` CloudTrail peristiwa muncul
- Rahasia cluster tetap tidak berubah
- Periksa `RotationFailed` acara CloudTrail untuk detailnya
- Selesaikan semua langkah persiapan untuk rotasi yang sukses

AWS PCS menghitung grup node

Grup node komputasi AWS PCS adalah kumpulan node logis (EC2 instance Amazon). Node ini dapat digunakan untuk menjalankan pekerjaan komputasi, serta untuk menyediakan akses interaktif berbasis shell ke sistem HPC. 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. Ketika 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 komputasi AWS PCS](#)
- [Menghapus grup node komputasi di PCS AWS](#)
- [Dapatkan detail grup node komputasi di AWS PCS](#)
- [Menemukan instance grup node komputasi di PCS AWS](#)

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 di [Memulai Layanan Komputasi AWS Paralel](#). Tutorial ini dapat membantu Anda membuat sistem HPC yang berfungsi tanpa memperluas ke semua opsi yang tersedia dan arsitektur sistem yang dimungkinkan.

Note

Anda dapat mengonfigurasi pengaturan Slurm khusus pada grup node komputasi untuk mengontrol pemanfaatan sumber daya dan perilaku tingkat simpul. Untuk informasi selengkapnya, lihat [Mengkonfigurasi pengaturan Slurm khusus di PCS AWS](#).

⚠ Important

AWS PCS saat ini membutuhkan kernel dengan IPv4 dukungan untuk komunikasi node lokal, bahkan ketika Anda menggunakan AWS PCS dalam jaringan IPv6 -only. Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Prasyarat

- Kuota layanan yang memadai untuk meluncurkan jumlah instans EC2 yang diinginkan di Anda. AWS Region Anda dapat menggunakan [Konsol Manajemen AWS](#) untuk memeriksa dan meminta kenaikan kuota layanan Anda.
- VPC dan subnet yang sudah ada yang memenuhi persyaratan jaringan AWS PCS. Kami menyarankan agar Anda benar-benar memahami persyaratan ini sebelum Anda menerapkan kluster untuk penggunaan produksi. Untuk informasi selengkapnya, lihat [AWS PCS VPC dan persyaratan subnet dan pertimbangan](#). Anda juga dapat menggunakan CloudFormation template untuk membuat VPC dan subnet. AWS menyediakan resep HPC untuk CloudFormation template. Untuk informasi lebih lanjut, lihat [aws-hpc-recipes](#) di GitHub.
- Profil instans IAM dengan izin untuk memanggil tindakan AWS PCS `RegisterComputeNodeGroupInstance` API dan akses ke AWS sumber daya lain yang diperlukan untuk instance grup node Anda. Untuk informasi selengkapnya, lihat [Profil instans IAM untuk Layanan Komputasi AWS Paralel](#).
- Template peluncuran untuk instance grup node Anda. Untuk informasi selengkapnya, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).
- Untuk membuat grup node komputasi yang menggunakan instans Spot Amazon EC2, Anda harus memiliki peran terkait layanan Spot `AWSServiceRoleForEC2` di situs Anda. Akun AWS Untuk informasi selengkapnya, lihat [Peran Amazon EC2 Spot untuk PCS AWS](#).

Buat grup node komputasi di AWS PCS


Anda dapat membuat grup node komputasi menggunakan Konsol Manajemen AWS atau. AWS CLI

Konsol Manajemen AWS

Untuk membuat grup node komputasi menggunakan konsol

1. Buka [konsol AWS PCS](#).

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. Template peluncuran EC2 - Pilih template peluncuran khusus 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 peluncuran Amazon EC2 dengan PCS AWS](#) untuk mempelajari cara membuatnya.

 Important

AWS PCS membuat 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. Versi template peluncuran EC2 - Anda harus memilih versi template peluncuran kustom Anda. Jika Anda mengubah versi nanti, Anda harus memperbarui grup node komputasi untuk mendeteksi perubahan dalam template peluncuran. Untuk informasi selengkapnya, lihat [Memperbarui grup node komputasi AWS PCS](#).
- c. ID AMI — jika template peluncuran Anda tidak menyertakan ID AMI, atau jika Anda ingin mengganti nilai dalam template peluncuran, berikan ID AMI di sini. Perhatikan bahwa AMI yang digunakan untuk grup node harus kompatibel dengan AWS PCS. Anda juga dapat memilih sampel AMI yang disediakan oleh AWS. Untuk informasi lebih lanjut tentang topik ini, lihat [Gambar Mesin Amazon \(AMIs\) untuk AWS PCS](#).
- d. Profil instans IAM - Pilih profil instance untuk grup simpul. Profil instans memberikan izin instans untuk mengakses AWS sumber daya dan layanan dengan aman. Jika Anda belum menyiapkannya, Anda dapat memilih Buat profil dasar agar AWS PCS membuatnya untuk Anda dengan kebijakan minimum, atau lihat [Profil instans IAM untuk Layanan Komputasi AWS Paralel](#).

- e. Subnet — Pilih satu atau beberapa subnet di VPC tempat cluster AWS PCS Anda digunakan. Jika Anda memilih beberapa subnet, komunikasi EFA 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 peluncuran EC2.
 - f. Instance — Pilih satu atau beberapa jenis instance untuk memenuhi permintaan penskalaan dalam grup node. Semua tipe instance harus memiliki arsitektur prosesor yang sama (x86_64 atau arm64) dan jumlah v. CPUs. Jika instance memiliki GPUs, 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 PCS tidak mendukung grup node komputasi dengan campuran instance statis dan dinamis.
5. (Opsional) Di bawah Pengaturan tambahan, tentukan yang berikut ini:
- a. Opsi pembelian — pilih Instans Sesuai Permintaan, Instans Spot, atau Blok Kapasitas yang ada. Juga pilih On-Demand jika Anda berencana untuk menggunakan On-Demand Capacity Reservation (ODCR). Untuk informasi selengkapnya, lihat [Menggunakan ODCRs dengan AWS PCS](#). Pilih Blok Kapasitas untuk menggunakan Blok Kapasitas Amazon EC2 yang ada untuk reservasi ML. Untuk informasi selengkapnya, lihat [Menggunakan Blok Kapasitas Amazon EC2 untuk ML dengan PCS AWS](#).
 - 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, Anda dapat menambahkan nama parameter dan pasangan nilai untuk mengonfigurasi pengaturan Slurm tambahan. Untuk daftar lengkap parameter yang didukung, lihat [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#).
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 menyediakan grup node. Ini dapat memakan waktu beberapa menit.

Direkomendasikan langkah selanjutnya

- Tambahkan grup node Anda ke antrian di 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 AWS Region untuk membuat cluster Anda, seperti `us-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 VPC cluster Anda.
5. Ganti *lt-ExampleID1* dengan ID untuk template peluncuran kustom Anda. Jika Anda belum menyiapkannya, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#) untuk mempelajari cara membuatnya.

Important

AWS PCS membuat 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.

6. Ganti *launch-template-version* dengan versi template peluncuran tertentu. AWS PCS mengaitkan grup node Anda dengan versi spesifik dari template peluncuran.

7. Ganti `arn:InstanceProfile` dengan ARN profil instans IAM Anda. Jika Anda belum mempersiapkannya, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#) bimbingan.
8. Ganti `min-instances` dan `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 PCS tidak 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 (x86_64 atau arm64) dan jumlah v. CPUs. Jika instance memiliki GPUs, 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
```

Example— Membuat grup node komputasi dengan pengaturan Slurm kustom

```
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 \
  --slurm-configuration \
  'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu, nvme"}]'
```

Untuk informasi selengkapnya, lihat [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#).

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 ke AMI, atau jika Anda ingin mengganti nilai tersebut. Perhatikan bahwa AMI yang digunakan untuk grup node harus kompatibel dengan AWS PCS. Anda juga dapat memilih sampel AMI yang disediakan oleh AWS. Untuk informasi lebih lanjut tentang topik ini, lihat [Gambar Mesin Amazon \(AMIs\) untuk AWS PCS](#).
- Gunakan `--purchase-option` untuk memilih cara AWS PCS membeli instans EC2 untuk grup node komputasi Anda. On-Demand adalah default.
 - ONDEMAND— Gunakan Instans Sesuai Permintaan. Pilih juga opsi ini jika Anda berencana untuk menggunakan Reservasi Kapasitas Sesuai Permintaan (ODCR). Untuk informasi selengkapnya, lihat [Menggunakan ODCRs dengan AWS PCS](#).
 - SPOT— Gunakan Instans Spot. Jika Anda memilih instans Spot, Anda juga dapat menggunakan `--allocation-strategy` untuk menentukan bagaimana 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](#).
 - CAPACITY_BLOCK— Gunakan Blok Kapasitas Amazon EC2 yang ada untuk reservasi ML. Untuk informasi selengkapnya, lihat [Menggunakan Blok Kapasitas Amazon EC2 untuk ML dengan PCS AWS](#).
- 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 sewenang-wenang. 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 di AWS PCS dalam Slurm konfigurasi Anda. Untuk informasi lebih lanjut, lihat [RealMemory](#) di 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 komputasi AWS PCS

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda memperbarui grup node komputasi AWS PCS. Untuk informasi tentang pengaturan kustom Slurm, lihat. [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#)

Opsi untuk memperbarui grup node komputasi AWS PCS

Memperbarui grup node komputasi AWS PCS memungkinkan Anda mengubah properti instans yang diluncurkan oleh AWS PCS, serta aturan tentang cara instance tersebut diluncurkan. Misalnya, Anda dapat mengganti AMI untuk instance grup 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 dan opsi pembelian yang disukai.

Pengaturan grup node berikut tidak dapat diubah setelah pembuatan:

- Nama
- Contoh

Pertimbangan saat memperbarui grup node komputasi AWS PCS

Grup node komputasi menentukan instans EC2 yang digunakan untuk memproses pekerjaan, menyediakan akses shell interaktif, dan tugas lainnya. Mereka sering dikaitkan dengan satu atau

lebih antrian AWS PCS. 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 instans IAM yang digunakan oleh grup node komputasi, ini akan segera berlaku pada instance running dan future.
- Untuk mengubah AMI yang digunakan oleh instance grup node komputasi, perbarui grup node komputasi (atau template peluncurannya) untuk menggunakan AMI baru dan tunggu AWS PCS mengganti instance.
- AWS PCS menggantikan 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 manggurr.
- Jika Anda mengurangi jumlah maksimum instance yang diizinkan dalam grup node komputasi, AWS PCS menghapus node dari Slurm untuk memenuhi maksimum baru. AWS PCS mengakhiri instance yang sedang berjalan terkait dengan node Slurm yang dihapus. Pekerjaan yang berjalan pada node yang dihapus gagal dan kembali ke antrian mereka.
- AWS PCS membuat template peluncuran terkelola untuk setiap grup node komputasi. Mereka diberi namapcs-*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 terkait layanan AWSServiceRoleForEC2Spot di akun Anda. Untuk informasi selengkapnya, lihat [Peran Amazon EC2 Spot untuk PCS AWS](#).

Untuk memperbarui grup node komputasi AWS PCS

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

Konsol Manajemen AWS

Untuk memperbarui grup node komputasi

1. Buka konsol AWS PCS 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.

Untuk informasi selengkapnya tentang pengaturan kustom Slurm, lihat. [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#)

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 Region tempat Anda ingin membuat klaster.
 - 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
```

Example— Memperbarui grup node komputasi dengan pengaturan Slurm khusus

```
aws pcs update-compute-node-group --region region-code \
  --cluster-identifier my-cluster \
  --compute-node-group-identifier my-node-group \
  --slurm-configuration \
  'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu, nvme"}]'
```

Untuk informasi selengkapnya, lihat [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#).

- Perbarui parameter grup node apa pun kecuali untuk `--instance-configs`. Misalnya, untuk menyetel ID AMI baru, pass `--amiId my-custom-ami-id my-custom-ami-id` where diganti dengan 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 PCS AWS

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

Pertimbangan saat menghapus grup node komputasi

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

- Setiap instans EC2 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 antrian AWS PCS](#).

Hapus grup node komputasi

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk menghapus grup node komputasi.

Konsol Manajemen AWS

Untuk menghapus grup node komputasi

1. Buka [konsol AWS PCS](#).
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 AWS Region 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`.

Dapatkan detail grup node komputasi di AWS PCS

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk mendapatkan detail tentang grup node komputasi, seperti ID grup node komputasi, Nama Sumber Daya Amazon (ARN), dan ID Amazon Machine Image (AMI). Detail ini seringkali merupakan nilai yang diperlukan untuk tindakan dan konfigurasi AWS PCS API.

Konsol Manajemen AWS

Untuk mendapatkan detail grup node komputasi

1. Buka [konsol AWS PCS](#).
2. Pilih cluster.
3. Pilih Compute node groups.
4. Pilih grup node komputasi dari panel daftar.

AWS CLI

Untuk mendapatkan detail grup node komputasi

1. Gunakan tindakan [ListClusters](#) API untuk menemukan nama atau ID klaster Anda.

```
aws pcs list-clusters
```

Contoh keluaran:

```
{
  "clusters": [
    {
      "name": "get-started-cfn",
      "id": "pcs_abc1234567",
      "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567",
      "createdAt": "2025-04-01T20:11:22+00:00",
      "modifiedAt": "2025-04-01T20:11:22+00:00",
      "status": "ACTIVE"
    }
  ]
}
```

2. Gunakan tindakan [ListComputeNodeGroups](#) API untuk mencantumkan grup node komputasi dalam klaster.

```
aws pcs list-compute-node-groups --cluster-identifier cluster-name-or-id
```

Contoh panggilan:

```
aws pcs list-compute-node-groups --cluster-identifier get-started-cfn
```

Contoh keluaran:

```
{
  "computeNodeGroups": [
    {
      "name": "compute-1",
      "id": "pcs_abc123abc1",
      "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/computenodegroup/pcs_abc123abc1",
    }
  ]
}
```

```

        "clusterId": "pcs_abc1234567",
        "createdAt": "2025-04-01T20:19:25+00:00",
        "modifiedAt": "2025-04-01T20:19:25+00:00",
        "status": "ACTIVE"
    },
    {
        "name": "login",
        "id": "pcs_abc456abc7",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
        computenodegroup/pcs_abc456abc7",
        "clusterId": "pcs_abc1234567",
        "createdAt": "2025-04-01T20:19:31+00:00",
        "modifiedAt": "2025-04-01T20:19:31+00:00",
        "status": "ACTIVE"
    }
]
}

```

- Gunakan tindakan [GetComputeNodeGroup](#) API untuk mendapatkan detail tambahan untuk grup node komputasi.

```
aws pcs get-compute-node-group --cluster-identifier cluster-name-or-id --
compute-node-group-identifier compute-node-group-name-or-id
```

Contoh panggilan:

```
aws pcs get-compute-node-group --cluster-identifier get-started-cfn --compute-
node-group-identifier compute-1
```

Contoh keluaran:

```

{
  "computeNodeGroup": {
    "name": "compute-1",
    "id": "pcs_abc123abc1",
    "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
    computenodegroup/pcs_abc123abc1",
    "clusterId": "pcs_abc1234567",
    "createdAt": "2025-04-01T20:19:25+00:00",
    "modifiedAt": "2025-04-01T20:19:25+00:00",
    "status": "ACTIVE",
    "amiId": "ami-0123456789abcdef0",

```

```
"subnetIds": [
  "subnet-abc012345789abc12"
],
"purchaseOption": "ONDEMAND",
"customLaunchTemplate": {
  "id": "lt-012345abcdef01234",
  "version": "1"
},
"iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-profile/
AWSPCS-get-started-cfn-us-east-1",
"scalingConfiguration": {
  "minInstanceCount": 0,
  "maxInstanceCount": 4
},
"instanceConfigs": [
  {
    "instanceType": "c6i.xlarge"
  }
]
}
```

Menemukan instance grup node komputasi di PCS AWS

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

Konsol Manajemen AWS

Untuk menemukan instance grup node komputasi Anda

1. Buka [konsol AWS PCS](#).
2. Pilih cluster.
3. Pilih Compute Node Groups.
4. Temukan ID untuk grup node login yang Anda buat.
5. Arahkan ke [konsol EC2](#) 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 AWS Region cluster Anda. Contoh: us-east-1
- Ganti *node-group-id* dengan ID (bukan nama) grup node komputasi Anda. Untuk menemukan ID dari grup node komputasi, lihat [Dapatkan detail grup node komputasi di AWS PCS](#).
- Ganti *running* dengan status instance lain seperti *pending* atau *terminated* untuk menemukan instans EC2 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 [DescribeInstances](#) di Referensi Amazon Elastic Compute Cloud API.

Menggunakan template peluncuran Amazon EC2 dengan PCS AWS

Di Amazon EC2, template peluncuran dapat menyimpan serangkaian preferensi sehingga Anda tidak perlu menentukannya satu per satu saat meluncurkan instance. AWS PCS menggabungkan template peluncuran sebagai cara yang fleksibel untuk mengkonfigurasi grup node komputasi. Saat Anda membuat grup node, Anda menyediakan template peluncuran. AWS PCS membuat template peluncuran turunan darinya yang mencakup transformasi untuk membantu memastikannya bekerja dengan layanan.

Memahami apa pilihan dan pertimbangan saat menulis template peluncuran kustom dapat membantu Anda menulis satu untuk digunakan dengan AWS PCS. Untuk informasi selengkapnya tentang template peluncuran, lihat [Meluncurkan Instance dari Meluncurkan instance dari template peluncuran](#) di Panduan Pengguna Amazon EC2.

Topik

- [Ikhtisar template peluncuran di AWS PCS](#)
- [Buat template peluncuran dasar](#)
- [Bekerja dengan data pengguna Amazon EC2 untuk PCS AWS](#)
- [Reservasi Kapasitas di AWS PCS](#)
- [Parameter template peluncuran yang berguna](#)

Ikhtisar template peluncuran di AWS PCS

Ada [lebih dari 30 parameter yang tersedia](#) yang dapat Anda sertakan dalam template peluncuran EC2, 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:

- Atribut tipe type/Specify instans instance (InstanceRequirements) — AWS PCS tidak mendukung pemilihan instans berbasis atribut.
- Jenis instans (InstanceType) - Tentukan jenis instance saat Anda membuat grup simpul.

- `Advanced details/IAM instance profile (IamInstanceProfile)` - Anda memberikan ini ketika Anda membuat atau memperbarui grup node.
- `Advanced details/Disable API termination (DisableApiTermination)` — AWS PCS harus mengontrol siklus hidup instance grup node yang diluncurkan.
- `Advanced details/Disable API stop (DisableApiStop)` — AWS PCS harus mengontrol siklus hidup instance grup node yang diluncurkan.
- `Advanced details/Stop` — Perilaku hibernate (`HibernationOptions`) — AWS PCS tidak mendukung hibernasi instance.
- `Advanced details/Elastic GPU (ElasticGpuSpecifications)` - Amazon Elastic Graphics mencapai akhir masa pakai pada 8 Januari 2024.
- `Advanced details/Elastic Inference (ElasticInferenceAccelerators)` - Amazon Elastic Inference tidak lagi tersedia untuk pelanggan baru.
- `Advanced details/Specify CPU options/Threads per core (ThreadsPerCore)` — AWS PCS menetapkan 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 pengguna Amazon EC2 untuk PCS AWS](#).
- `Aplikasi dan Gambar OS (ImageId)` - Anda dapat memasukkan ini. Namun, jika Anda menentukan ID AMI saat Anda membuat atau memperbarui grup node, itu akan mengganti nilai dalam template peluncuran. AMI yang Anda berikan harus kompatibel dengan AWS PCS. Untuk informasi lebih lanjut, lihat "[Gambar Mesin Amazon \(AMIs\) untuk AWS PCS](#)".
- `Jaringan settings/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 [Grup keamanan di AWS PCS](#).
- `Konfigurasi settings/Advanced jaringan jaringan (NetworkInterfaces)` - Jika Anda menggunakan instans EC2 dengan satu kartu jaringan, 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 PCS. AWS Namun, Anda dapat mereferensikan grup reservasi kapasitas, di mana grup tersebut berisi satu atau lebih reservasi kapasitas. Lihat informasi yang lebih lengkap di [Reservasi Kapasitas di AWS PCS](#).

Buat template peluncuran dasar

Anda dapat membuat template peluncuran menggunakan Konsol Manajemen AWS atau AWS CLI.

Konsol Manajemen AWS

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 key pair SSH 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 VPC cluster PCS Anda. Minimal, pilih:
 - Grup keamanan yang memungkinkan komunikasi dengan cluster AWS PCS
 - Grup keamanan yang memungkinkan komunikasi antar EC2 instans yang diluncurkan oleh AWS PCS
 - (Opsional) Grup keamanan yang memungkinkan akses SSH 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 formulir1t-0123456789abcdef01.

Direkomendasikan langkah selanjutnya

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

AWS CLI

Untuk membuat templat peluncuran

Buat template peluncuran Anda dengan perintah berikut.

- Sebelum menjalankan perintah, buat penggantian berikut:
 - a. Ganti *region-code* dengan AWS Region tempat Anda bekerja dengan AWS PCS
 - b. Ganti *my-launch-template-name* dengan nama untuk template Anda. Itu harus unik untuk Akun AWS dan AWS Region Anda gunakan.
 - c. Ganti *my-ssh-key-name* dengan nama kunci SSH pilihan Anda.
 - d. Ganti *sg-ExampleID1* dan *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 keamanan IDs. Semua grup keamanan yang Anda sertakan dalam template peluncuran harus dari AWS VPC cluster PCS Anda.

```
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 komputasi AWS PCS.

Bekerja dengan data pengguna Amazon EC2 untuk PCS AWS

Anda dapat menyediakan data pengguna EC2 di template peluncuran yang `cloud-init` berjalan saat instans diluncurkan. Blok data pengguna dengan tipe konten `cloud-config` dijalankan sebelum instance mendaftar dengan AWS PCS API, 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 template peluncuran harus dalam format [arsip multi-bagian MIME](#). Ini karena data pengguna Anda digabungkan dengan data pengguna AWS PCS lain yang diperlukan untuk mengkonfigurasi node di grup node Anda. Anda dapat menggabungkan beberapa blok data pengguna menjadi satu file multi-bagian MIME.

File multi-bagian MIME terdiri dari komponen berikut:

- Jenis konten dan deklarasi batas bagian: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- Deklarasi versi MIME: `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 multi-bagian MIME: `--==BOUNDARY==--` Anda harus menjaga garis sebelum batas penutupan kosong.

Note

Jika Anda menambahkan data pengguna ke template peluncuran di konsol Amazon EC2, 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 JSON ini.

```
{
  "LaunchTemplateName": "base64-user-data",
  "LaunchTemplateData": {
    "UserData":
"ewogICAgIkhhdW5jaFR1bXBsYXR1TmFtZSI6ICJpbmNyZWZzZS1jb250YWluZXItZm9sdW..."
  }
}
```

Contoh

- [Contoh: Instal perangkat lunak dari repositori paket](#)
- [Contoh: Jalankan skrip dari bucket S3](#)
- [Contoh: Mengatur variabel lingkungan global](#)
- [Menggunakan sistem file jaringan dengan AWS PCS](#)
- [Contoh: Gunakan sistem file EFS 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 pengguna Amazon EC2 untuk PCS AWS](#).

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 lvm.

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
- goyang

--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 pengguna Amazon EC2 untuk PCS AWS](#).

Skrip data pengguna berikut 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 dengan detail Anda sendiri:

- *amzn-s3-demo-bucket*— Nama bucket S3 yang dapat dibaca akun Anda.

- *object-key*— Kunci objek S3 dari skrip yang akan diimpor. Ini termasuk nama skrip dan lokasinya dalam struktur folder bucket. Misalnya, `scripts/script.sh`. Untuk informasi selengkapnya, lihat [Mengatur objek di konsol Amazon S3 menggunakan folder](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- *shell*— Shell Linux yang digunakan untuk menjalankan skrip, seperti `bash`.

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

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

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/object-key /tmp/script.sh
- /usr/bin/shell /tmp/script.sh

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

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

JSON

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

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 pengguna Amazon EC2 untuk PCS AWS](#).

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 file EFS sebagai direktori home bersama untuk AWS PCS

Berikan skrip ini sebagai nilai "userData" dalam template peluncuran Anda. Untuk informasi selengkapnya, lihat [Bekerja dengan data pengguna Amazon EC2 untuk PCS AWS](#).

Contoh ini memperluas contoh EFS mount in [Menggunakan sistem file jaringan dengan AWS PCS](#) untuk mengimplementasikan direktori home bersama. Isi `/home` dicadangkan sebelum sistem file EFS 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`— Path pada instance di mana Anda ingin me-mount sistem file EFS.
- `filesystem-id`— ID sistem file untuk sistem file EFS.

```
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===--
```

Contoh: Mengaktifkan SSH tanpa kata sandi

Anda dapat membangun contoh direktori home bersama untuk mengimplementasikan koneksi SSH antara instance cluster menggunakan kunci SSH. 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 koneksi SSH 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 Amazon untuk ML untuk memastikan bahwa Anda memiliki kapasitas komputasi yang diperlukan yang tersedia saat Anda membutuhkannya.

Reservasi Kapasitas Sesuai Permintaan (ODCRs) memungkinkan Anda memesan kapasitas komputasi untuk EC2 instans Amazon di Availability Zone tertentu untuk durasi berapa pun. Anda dapat membuat dan membatalkan reservasi kapan saja, tanpa komitmen jangka panjang atau pembayaran di muka. ODCRs ideal ketika Anda membutuhkan reservasi kapasitas fleksibel yang dapat Anda modifikasi saat kebutuhan Anda berubah. Untuk informasi selengkapnya, lihat [Reservasi Kapasitas Sesuai Permintaan](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Amazon EC2 Capacity Blocks for ML memungkinkan Anda memesan instans komputasi akselerasi berbasis GPU untuk penggunaan di masa mendatang, hingga 8 minggu sebelumnya. Anda dapat memesan blok 1-64 instance untuk jangka waktu dari 1 hari hingga 6 bulan. Blok Kapasitas ideal untuk beban kerja pembelajaran mesin yang memerlukan akses terjamin ke kapasitas GPU pada waktu tertentu. Untuk informasi selengkapnya, lihat [Blok Kapasitas untuk ML](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Topik

- [Menggunakan ODCRs dengan AWS PCS](#)
- [Menggunakan Blok Kapasitas Amazon EC2 untuk ML dengan PCS AWS](#)

Menggunakan ODCRs dengan AWS PCS

Anda dapat memilih bagaimana AWS PCS mengkonsumsi instans cadangan Anda. Jika Anda membuat ODCR terbuka, setiap instans yang cocok yang diluncurkan oleh AWS PCS atau proses lain di akun Anda dihitung terhadap reservasi. Dengan ODCR yang ditargetkan, 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 komputasi AWS PCS untuk menggunakan ODCR yang ditargetkan dengan menambahkannya ke template peluncuran. Berikut adalah langkah-langkah untuk melakukannya:

1. Buat Reservasi Kapasitas Sesuai Permintaan (ODCR) yang ditargetkan menggunakan [Amazon EC2 Buat](#) Panduan Pengguna Reservasi Kapasitas.
2. Kaitkan ODCR dengan template peluncuran. Ada dua cara untuk melakukannya:
 - a. Asosiasi ODCR langsung: Referensikan ID ODCR langsung di template peluncuran. Pendekatan ini memberikan kontrol kapasitas yang ketat dan tidak mendukung penimbunan ulang instance (Jika grup node komputasi meminta lebih banyak instance daripada yang tersedia di ODCR, tidak ada instance tambahan yang akan diluncurkan).
 - b. Asosiasi grup Reservasi Kapasitas: Tambahkan ODCR ke grup Reservasi Kapasitas dan referensi grup dalam templat peluncuran. Pendekatan ini mendukung penimbunan ulang instans, memungkinkan AWS PCS meluncurkan instance On-Demand tambahan jika kapasitas reservasi terlampaui.
3. Buat atau perbarui grup node komputasi AWS PCS untuk menggunakan template peluncuran. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS PCS Compute Node Groups](#).
 - Atur grup node komputasi ke `ONDEMAND`. `purchaseOption`

Contoh: Cadangan dan gunakan instance `hpc6a.48xlarge` dengan ODCR yang ditargetkan

Perintah contoh ini membuat ODCR yang ditargetkan 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 ODCR baru. [ID ODCR dapat diambil dari ARN "arn:aws:ec2:us-east-2:123456789012:capacity-reservation/ODCR-ID"](#) atau dengan menggunakan [Amazon EC2. DescribeCapacityReservations](#)

Asosiasi ODCR langsung: Tambahkan ID ODCR ke template peluncuran. Berikut adalah contoh template peluncuran yang mereferensikan ID ODCR.

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationTarget": {
      "CapacityReservationId": "cr-1234567890abcdef1"
    }
  }
}
```

Asosiasi grup Reservasi Kapasitas: Buat grup Reservasi Kapasitas dan tambahkan grup ke templat peluncuran. Perintah berikut membuat grup Reservasi Kapasitas bernama `EXAMPLE-CR-GROUP`.

```
aws resource-groups create-group \
  --name EXAMPLE-CR-GROUP \
  --configuration \
    '{"Type": "AWS::EC2::CapacityReservationPool"}' \
    '{"Type": "AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

Perintah berikut menambahkan ODCR ke grup Reservasi Kapasitas.

```
aws resource-groups group-resources --group EXAMPLE-CR-GROUP \
  --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/
cr-1234567890abcdef1
```

Dengan ODCR dibuat dan ditambahkan ke grup Reservasi Kapasitas, sekarang dapat dihubungkan ke grup node komputasi AWS PCS 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 komputasi AWS PCS untuk menggunakan instance `hpc6a.48xlarge` dan gunakan templat peluncuran yang mereferensikan 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 ke ukuran instans yang Anda inginkan.

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

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

Menggunakan Blok Kapasitas Amazon EC2 untuk ML dengan PCS AWS

Blok Kapasitas Amazon EC2 untuk ML adalah opsi pembelian Amazon EC2 yang memungkinkan Anda membayar di muka untuk memesan instans komputasi akselerasi berbasis GPU dalam rentang tanggal dan waktu tertentu untuk mendukung beban kerja berdurasi pendek. Instans yang berjalan di dalam Blok Kapasitas secara otomatis ditempatkan berdekatan di dalam Amazon UltraClusters EC2, untuk jaringan latensi rendah, skala petabit, dan non-pemblokiran. Untuk informasi selengkapnya, lihat [Blok Kapasitas untuk ML](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Anda dapat menggunakan template peluncuran agar AWS PCS menggunakan Blok Kapasitas saat meluncurkan instance untuk grup node komputasi.

Note

AWS PCS memperkenalkan dukungan untuk Blok Kapasitas sejak Slurm versi 24.05.

Batasan

- AWS PCS hanya mendukung Blok Kapasitas dengan keluarga instans P5en, P5e, P5, dan P4d.
- Anda hanya dapat mengaitkan grup node komputasi dengan 1 Blok Kapasitas pada satu waktu.
- Anda tidak dapat mengaitkan grup node komputasi dengan grup reservasi kapasitas yang menggabungkan beberapa Blok Kapasitas.
- Blok Kapasitas harus dalam `active` keadaan `scheduled` atau untuk digunakan dengan AWS PCS. Anda tidak dapat menggunakan Blok Kapasitas di negara bagian lain, seperti `payment-failed`. Untuk informasi selengkapnya, lihat [Melihat Blok Kapasitas](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Kedaluwarsa Blok Kapasitas

Blok Kapasitas terbatas pada tanggal dan rentang waktu tertentu. Saat Blok Kapasitas kedaluwarsa:

- Grup node komputasi yang terkait dengan Blok Kapasitas itu terus ada dan tetap terkait dengan antrian yang sama.
- Semua instance dalam grup node komputasi dihentikan dan pekerjaan aktif mungkin gagal, berdasarkan pengaturan Slurm Anda.
- AWS PCS tidak dapat meluncurkan instance baru di grup node komputasi.
- Semua pekerjaan antrian atau yang baru dikirimkan tetap dalam status tertunda hingga grup node komputasi lain dilampirkan ke antrian atau Anda memperbarui grup node komputasi untuk menggunakan templat peluncuran baru yang menentukan Blok Kapasitas baru.

Konfigurasi grup node komputasi AWS PCS untuk menggunakan Blok Kapasitas

Untuk mengaitkan Blok Kapasitas dengan grup node komputasi

1. Buat template EC2 peluncuran Amazon untuk AWS PCS yang menentukan Blok Kapasitas Anda. Untuk informasi selengkapnya tentang membuat template peluncuran untuk AWS PCS, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).

Template peluncuran Anda harus mencakup:

- Nilai `MarketType InstanceMarketOptions` harus diatur ke `capacity-block`.
 - A `CapacityReservationSpecification` dengan valid `CapacityReservationId`
 - Valid `InstanceType` yang cocok dengan jenis instans Blok Kapasitas yang Anda beli.
2. Buat grup node komputasi yang menggunakan template peluncuran. Untuk informasi selengkapnya, lihat [Membuat grup node komputasi di AWS PCS](#). Anda juga dapat memperbarui grup node komputasi yang ada untuk menggunakan template peluncuran. Untuk informasi selengkapnya, lihat [Memperbarui grup node komputasi AWS PCS](#).

Saat Anda membuat atau memperbarui grup node komputasi:

- Identitas IAM yang Anda gunakan untuk membuat atau memperbarui grup node komputasi harus memiliki izin berikut:

```
ec2:DescribeCapacityReservations
```

Untuk informasi selengkapnya, lihat [Izin minimum untuk AWS PCS](#).

- Blok Kapasitas harus dalam `active` keadaan `scheduled` atau.
- Atur grup node komputasi ke `CAPACITY_BLOCK.purchaseOption`
- Kelompok node komputasi tidak boleh melebihi ukuran Blok Kapasitas. `maxInstanceCount`
- Zona ketersediaan grup node komputasi harus cocok dengan 1 zona ketersediaan subnet grup node komputasi.

Important

Anda tidak dapat mengubah tipe instance dari grup node komputasi saat Anda memperbaruinya. Anda hanya dapat menggunakan Blok Kapasitas dengan tipe instance yang sama dengan grup node komputasi. Jika Anda ingin menggunakan Blok Kapasitas dengan jenis instans yang berbeda, Anda harus membuat grup node komputasi baru.

Pertanyaan yang sering diajukan tentang penggunaan Blok Kapasitas dengan AWS PCS

Saya baru saja membayar Blok Kapasitas dan segera mencoba menggunakannya dengan AWS PCS tetapi pembuatan grup node komputasi gagal. Apa yang terjadi?

Blok Kapasitas Anda mungkin tidak dalam `active` keadaan `scheduled` atau. Coba lagi setelah Blok Kapasitas `scheduled` atau `active`.

Saya menggunakan Blok Kapasitas di AWS PCS dan saya membeli ekstensi sebelum kedaluwarsa. Bagaimana cara saya terus menggunakannya di AWS PCS?

Anda tidak perlu melakukan apa pun untuk terus menggunakan Blok Kapasitas di AWS PCS. Tanggal akhir Blok Kapasitas Anda diperbarui setelah pembayaran ekstensi Anda berhasil. Selama Blok Kapasitas Anda tidak kedaluwarsa, grup node komputasi terus beroperasi. Jika pembayaran ekstensi gagal, Blok Kapasitas tetap ada `active` dan grup node komputasi beroperasi hingga Blok Kapasitas berakhir pada tanggal akhir aslinya.

Apa yang terjadi pada pekerjaan saya yang sedang antri dan berjalan jika Blok Kapasitas saya kedaluwarsa?

Pekerjaan antrian yang tidak dimulai sebelum Blok Kapasitas kedaluwarsa tetap tertunda hingga Anda melampirkan grup node komputasi lain ke antrian atau Anda memperbarui grup node

komputasi dengan Blok Kapasitas baru. Anda masih dapat mengirimkan pekerjaan ke antrian. Pengaturan Slurm Anda memengaruhi pekerjaan aktif. Secara default, pekerjaan aktif secara otomatis diantrian ulang, tetapi mungkin memiliki kesalahan atau gagal.

Blok Kapasitas Saya kedaluwarsa. Haruskah aku melakukan sesuatu?

Anda tidak perlu melakukan apa pun. Anda dapat memeriksa konsol Amazon EC2 untuk status reservasi kapasitas EC2 Anda. Ketika Blok Kapasitas kedaluwarsa, grup node komputasi yang terkait dengan Blok Kapasitas itu terus ada dan menangani antrian yang sama. Grup node komputasi tidak memiliki instance apa pun untuk menjalankan pekerjaan. Anda dapat menghapus grup node komputasi atau memisahkannya dari antrian untuk mencegah pengguna mengirimkan pekerjaan yang tidak akan berjalan.

Saya ingin menggunakan Blok Kapasitas baru dengan grup node komputasi AWS PCS saya. Apa yang harus saya lakukan?

Kami menyarankan Anda membuat grup node komputasi baru untuk menggunakan Blok Kapasitas baru. Untuk informasi selengkapnya, lihat [Konfigurasi grup node komputasi AWS PCS untuk menggunakan Blok Kapasitas](#).

Bagaimana cara berbagi 1 Blok Kapasitas di seluruh cluster dan layanan?

Anda dapat membagi Blok Kapasitas di beberapa cluster dan layanan. Misalnya, untuk membagi Blok Kapasitas dengan 64 p5.48xlarge instance dengan 20 node pada PCS-Cluster-1, 16 node pada PCS-Cluster-2, dan node yang tersisa untuk layanan lain, atur keduanya dan 20 untuk PCS-Cluster-1 `minInstanceCount` dan `maxInstanceCount` 16 untuk PCS-Cluster-2.

Dapatkah saya menggunakan lebih dari 1 Blok Kapasitas atau kapasitas gabungan dengan 1 grup node komputasi?

Tidak. Hanya 1 Blok Kapasitas yang dapat dikaitkan dengan grup node komputasi tunggal. AWS PCS tidak mendukung grup reservasi kapasitas yang menggabungkan beberapa Blok Kapasitas.

Bagaimana saya tahu kapan Blok Kapasitas saya mulai atau kedaluwarsa?

Terlepas dari AWS PCS, Amazon EC2 mengirimkan Capacity Block Reservation Delivered acara melalui EventBridge saat reservasi Blok Kapasitas dimulai dan Capacity Block Reservation Expiration Warning acara 40 menit sebelum reservasi Blok Kapasitas berakhir. Untuk informasi selengkapnya, lihat [Memantau Blok Kapasitas menggunakan EventBridge](#) Panduan Pengguna Amazon Elastic Compute Cloud.

Bagaimana Slurm melacak status Blok Kapasitas saya?

Anda dapat menjalankan `sinfo` untuk memahami bagaimana AWS PCS menggunakan Blok Kapasitas. Dalam contoh output berikut, antrian dikaitkan dengan grup node komputasi yang menjalankan 4 instance active dari Blok Kapasitas. Node berada dalam keadaan `idle` Slurm (tersedia untuk digunakan dan belum dialokasikan untuk pekerjaan apa pun).

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
fanout up infinite 4 idle node-fanout-[1-4]
```

Jika node berada dalam `maint` status, Anda dapat menjalankan `scontrol show res` untuk melihat detail tentang reservasi Slurm yang mengontrol status ini. Dalam contoh output berikut, Blok Kapasitas adalah `scheduled` dengan tanggal mulai `future`.

```
$ scontrol show res

ReservationName=node-fanout-scheduled StartTime=2025-10-14T13:09:17
EndTime=2025-10-14T13:11:17 Duration=00:02:00
  Nodes=node-fanout-[1-4] NodeCnt=4 CoreCnt=16 Features=(null) PartitionName=(null)
Flags=MAINT,SPEC_NODES
  TRES=cpu=16

  Users=root Groups=(null) Accounts=(null) Licenses=(null) State=ACTIVE
BurstBuffer=(null)
  MaxStartDelay=(null)

  Comment=node-fanout Scheduled
```

Bagaimana saya bisa tahu apakah kesalahan yang saya dapatkan saat meluncurkan kapasitas adalah karena Blok Kapasitas saya dibagikan?

Periksa Reservasi Kapasitas di konsol Amazon EC2 untuk mengetahui berapa banyak instans dari Blok Kapasitas yang disediakan secara aktif. Periksa tag setiap instance untuk menemukan layanan atau cluster mana yang menggunakannya. Misalnya, semua instance untuk AWS PCS memiliki tag AWS PCS seperti `aws:pcs:cluster-id = pcs_l0mizqyk5o | aws:pcs:compute-node-group-id = pcs_ic7onkfmfqk` yang menunjukkan cluster dan grup node komputasi mana yang dimiliki instance. Anda kemudian dapat memeriksa apakah Blok Kapasitas berada pada kapasitas maksimum.

Anda gunakan `scontrol show nodes` untuk memeriksa apakah node Blok Kapasitas di cluster AWS PCS memicu `ReservationCapacityExceeded`:

```
[root@ip-172-16-10-54 ~]# scontrol show nodes test-node-8-gamma-cb-2
NodeName=test-8-gamma-cb-2 CoresPerSocket=1
CPUAlloc=0 CPUEfctv=8 CPUTot=8 CPULoad=0.00
AvailableFeatures=test-8-gamma-cb,gpu
ActiveFeatures=test-8-gamma-cb,gpu
Gres=gpu:H100:1
NodeAddr=test-8-gamma-cb-2 NodeHostName=test-8-gamma-cb-2
RealMemory=249036 AllocMem=0 FreeMem=N/A Sockets=8 Boards=1
State=IDLE+CLOUD+POWERING_DOWN ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A
MCS_label=N/A
Partitions=my-q
BootTime=None SlurmdStartTime=None
LastBusyTime=Unknown ResumeAfterTime=None
CfgTRES=cpu=8,mem=249036M,billing=8
AllocTRES=
CurrentWatts=0 AveWatts=0
Reason=Failed to launch backing instance (Error Code:
ReservationCapacityExceeded) [root@2025-08-28T15:15:33]
```

Ketika beberapa grup node komputasi dilampirkan ke antrian yang sama, bagaimana saya bisa memaksa pekerjaan untuk berjalan pada instance yang didukung Blok Kapasitas?

Anda dapat menggunakan fitur dan kendala Slurm untuk mengunci pekerjaan ke set node tertentu. Kami menyarankan Anda untuk tidak menyetel bobot Slurm untuk setiap grup node komputasi karena itu hanya berfungsi dengan node yang tidak dalam status. `maint`

Parameter template peluncuran yang berguna

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

Aktifkan CloudWatch pemantauan terperinci

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

Konsol Manajemen AWS

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 Instans Versi 2 (IMDS v2)

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

Konsol Manajemen AWS

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 Antrian PCS

Antrian AWS PCS adalah abstraksi ringan atas implementasi asli penjadwal dari antrian kerja. Dalam kasus Slurm, antrian AWS PCS 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. Cluster AWS PCS dapat memiliki beberapa antrian pekerjaan. Misalnya, Anda dapat membuat antrian yang menggunakan Instans EC2 Sesuai Permintaan Amazon untuk pekerjaan prioritas tinggi dan antrian lain yang menggunakan Instans EC2 Spot Amazon untuk pekerjaan dengan prioritas rendah.

Topik

- [Membuat antrian di AWS PCS](#)
- [Memperbarui antrian AWS PCS](#)
- [Menghapus antrian di PCS AWS](#)

Membuat antrian di AWS PCS

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

Note

Anda dapat mengonfigurasi pengaturan Slurm kustom pada antrian untuk menerapkan kebijakan penjadwalan khusus partisi dan manajemen sumber daya. Untuk informasi selengkapnya, lihat [Mengkonfigurasi pengaturan Slurm khusus di PCS AWS](#).

Prasyarat

- Cluster AWS PCS - antrian hanya dapat dibuat terkait dengan cluster AWS PCS tertentu.
- Satu atau lebih grup node komputasi AWS PCS - antrian harus dikaitkan dengan setidaknya satu grup node komputasi AWS PCS.

Untuk membuat antrian di AWS PCS

Anda dapat membuat antrian menggunakan Konsol Manajemen AWS atau. AWS CLI

Konsol Manajemen AWS

Untuk membuat antrian menggunakan konsol

1. Buka [konsol AWS PCS](#).
2. Pilih cluster untuk 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. Menghitung grup node — Pilih 1 atau lebih grup node komputasi untuk melayani antrian ini. Grup node komputasi dapat dikaitkan dengan lebih dari 1 antrian.
4. (Opsional) Di Pengaturan penjadwal tambahan bagian, Anda dapat menambahkan nama parameter dan pasangan nilai untuk mengonfigurasi pengaturan Slurm tambahan. Untuk daftar lengkap parameter yang didukung, lihat [Pengaturan Slurm khusus untuk AWS antrian PCS](#).
5. (Opsional) Di bawah Tag, tambahkan tag apa pun ke antrian AWS PCS Anda
6. Pilih Buat antrian. Bidang Status akan menampilkan Membuat sementara AWS PCS membuat antrian. Pembuatan antrian dapat memakan waktu beberapa menit.

Direkomendasikan langkah selanjutnya

- Kirim pekerjaan ke antrian baru Anda.


AWS CLI

Untuk membuat antrian menggunakan AWS CLI

Gunakan perintah berikut untuk membuat antrian Anda. Lakukan penggantian berikut:

1. Ganti *region-code* dengan AWS Region cluster. Misalnya, us-east-1.

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 ID cluster Anda.
4. Ganti *compute-node-group-id* dengan ID grup node komputasi untuk melayani antrian. Misalnya, pcs_abcdef12345.

 Note

Saat Anda membuat antrian, Anda harus memberikan ID grup node komputasi dan bukan namanya.

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

Example— Membuat antrian dengan pengaturan Slurm kustom

```
aws pcs create-queue --region region-code \  
  --queue-name my-queue \  
  --cluster-identifier my-cluster \  
  --compute-node-group-configurations \  
  computeNodeGroupId=compute-node-group-id \  
  --slurm-configuration \  
  'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Untuk informasi selengkapnya, lihat [Pengaturan Slurm khusus untuk AWS antrian PCS](#).

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 antrian AWS PCS

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda memperbarui antrian AWS PCS. Untuk informasi tentang pengaturan kustom Slurm, lihat [Pengaturan Slurm khusus untuk AWS antrian PCS](#)

Pertimbangan saat memperbarui antrian AWS PCS

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

Untuk memperbarui antrian AWS PCS

Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk memperbarui antrian.

Konsol Manajemen AWS

Untuk memperbarui antrian

1. Buka konsol AWS PCS 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.
 - Pengaturan penjadwal tambahan - Menambahkan, memodifikasi, atau menghapus pengaturan Slurm kustom untuk antrian. Untuk informasi selengkapnya, lihat [Pengaturan Slurm khusus untuk AWS antrian PCS](#).
 - 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 AWS Region yang Anda inginkan untuk membuat cluster Anda.
 - b. Ganti *my-queue* dengan nama atau computeNodeId 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
computeNodeId=computeNodeGroupExampleID1,computeNodeId=computeNodeGroupExampleID2
```

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

Example— Memperbarui antrian dengan pengaturan Slurm kustom

```
aws pcs update-queue --region region-code \
  --queue-identifier my-queue \
  --cluster-identifier my-cluster \
  --slurm-configuration \
```

```
'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Untuk informasi selengkapnya, lihat [Pengaturan Slurm khusus untuk AWS antrian PCS](#).

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 PCS AWS

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 stop/cancel secara manual menggunakan perintah asli penjadwal (seperti `scancel` untuk Slurm).

Hapus antrian


Anda dapat menggunakan Konsol Manajemen AWS atau AWS CLI untuk menghapus antrian.

Konsol Manajemen AWS

Untuk menghapus antrian

1. Buka [konsol AWS PCS](#).
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 AWS Region 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 Node login PCS

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

Topik

- [Menggunakan grup node komputasi AWS PCS untuk menyediakan node login](#)
- [Menggunakan instance mandiri sebagai node login AWS PCS](#)
- [Menghubungkan node login mandiri ke beberapa cluster di PCS AWS](#)

Menggunakan grup node komputasi AWS PCS untuk menyediakan node login

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

Membuat grup node komputasi AWS PCS 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 instans EC2 dalam grup node komputasi.
- Pilih opsi pembelian sesuai permintaan untuk menghindari instans Anda direklamsi.
- Pilih nama informatif untuk grup node komputasi, seperti login.
- Jika Anda ingin instance node login dapat diakses di luar VPC Anda, pertimbangkan untuk menggunakan subnet publik.
- Jika Anda bermaksud mengizinkan akses SSH, template peluncuran akan membutuhkan grup keamanan yang mengekspos port SSH ke alamat IP pilihan Anda.
- Profil instans IAM seharusnya hanya memiliki izin AWS yang Anda inginkan untuk dimiliki pengguna akhir Anda. Lihat [Profil instans IAM untuk Layanan Komputasi AWS Paralel](#) untuk detail.

- Pertimbangkan untuk mengizinkan AWS Systems Manager Session Manager mengelola instans login Anda.
- Pertimbangkan untuk membatasi akses ke kredensial AWS instans hanya untuk pengguna administratif
- Pilih jenis instance yang lebih murah daripada grup node komputasi biasa, karena node login akan berjalan terus menerus.
- Gunakan AMI yang sama (atau turunan) seperti untuk grup node komputasi Anda yang lain untuk membantu memastikan semua instance memiliki perangkat lunak yang sama diinstal. Untuk informasi selengkapnya tentang penyesuaian AMIs, lihat [Gambar Mesin Amazon \(AMIs\) untuk AWS PCS](#)
- Konfigurasi sistem file jaringan yang sama (Amazon EFS, Amazon FSx for Lustre, dll.) Mount pada node login Anda seperti pada instance komputasi Anda. Untuk informasi selengkapnya, lihat [Menggunakan sistem file jaringan dengan AWS PCS](#).

Akses node login Anda

Setelah grup node komputasi baru Anda mencapai status AKTIF, Anda dapat menemukan instans EC2 yang telah dibuatnya dan masuk ke dalamnya. Untuk informasi selengkapnya, lihat [Menemukan instance grup node komputasi di PCS AWS](#).

Memperbarui grup node komputasi AWS PCS 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 komputasi AWS PCS](#).

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. Instans EC2 baru yang diluncurkan di grup node komputasi menggunakan templat peluncuran yang diperbarui. Untuk informasi selengkapnya, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).

Menghapus grup node komputasi AWS PCS 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. Lihat informasi yang lebih lengkap di [Menghapus grup node komputasi di PCS AWS](#).

Menggunakan instance mandiri sebagai node login AWS PCS

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

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

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

Daftar Isi

- [Langkah 1 - Ambil alamat dan rahasia untuk cluster AWS PCS target](#)
- [Langkah 2 - Luncurkan instans EC2](#)
- [Langkah 3 - Instal Slurm pada instance](#)
- [Langkah 4 - Ambil dan simpan rahasia cluster](#)
- [Langkah 5 - Konfigurasi koneksi ke cluster AWS PCS](#)
- [Langkah 6 - \(Opsional\) Uji koneksi](#)

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

Mengambil rincian tentang target AWS PCS cluster menggunakan AWS CLI dengan perintah yang berikut. Sebelum menjalankan perintah, buat penggantian berikut:

- Ganti *region-code* dengan AWS Region 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": "get-started",
    "id": "pcs_123456abcd",
    "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
    "status": "ACTIVE",
    "createdAt": "2024-12-17T21:03:52+00:00",
    "modifiedAt": "2024-12-17T21:03:52+00:00",
    "scheduler": {
      "type": "SLURM",
      "version": "25.05"
    },
    "size": "SMALL",
    "slurmConfiguration": {
      "authKey": {
        "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!slurm-secret-pcs_123456abcd-a12ABC",
        "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
      }
    },
    "networking": {
      "subnetIds": [
        "subnet-0123456789abcdef0"
      ],
      "securityGroupIds": [
        "sg-0123456789abcdef0"
      ]
    },
    "endpoints": [
      {
        "type": "SLURMCTLD",
        "privateIpAddress": "10.3.149.220",
        "port": "6817"
      }
    ]
  }
}
```

```
}  
}
```

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 instans EC2

Cara meluncurkan instans EC2

1. Buka [konsol Amazon EC2](#).
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, seperti `PCS-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 key pair SSH yang akan digunakan untuk instance.
7. Di bagian Pengaturan jaringan:
 - Pilih Edit.
 - i. Pilih VPC cluster AWS PCS Anda.
 - ii. Untuk Firewall (grup keamanan), pilih Pilih grup keamanan yang ada.
 - A. Pilih grup keamanan yang mengizinkan lalu lintas antara instance dan pengontrol Slurm cluster AWS PCS target. Untuk informasi selengkapnya, lihat [Persyaratan dan pertimbangan kelompok keamanan](#).
 - B. (Opsional) Pilih grup keamanan yang memungkinkan akses SSH masuk ke instans Anda.
8. Di bagian Penyimpanan, konfigurasi 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 peran IAM 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 Region tempat cluster target berjalan, dan ganti *secret-arn* dengan nilai untuk `secretArn` diambil di [Langkah 1](#).

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

⚠ Warning

Dalam lingkungan multiuser, 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 ke IMDS hanya untuk pengguna root atau administratif. 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
```

ℹ Note

Kunci Slurm harus dimiliki oleh pengguna dan grup tempat sackd layanan berjalan.

Langkah 5 - Konfigurasi koneksi ke cluster AWS PCS

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

ℹ Note

Jika Anda menggunakan Slurm 25.05 atau yang lebih baru, Anda dapat menggunakan skrip untuk mengatur node login Anda untuk terhubung ke beberapa cluster sebagai gantinya. Untuk informasi selengkapnya, lihat [Menghubungkan node login mandiri ke beberapa cluster di PCS AWS](#).

1. Siapkan file lingkungan untuk sackd layanan dengan perintah berikut. Sebelum menjalankan perintah, ganti *ip-address* dan *port* dengan nilai yang diambil dari titik akhir di [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-25.05/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 2024-12-17 16:34:55 UTC; 8s ago
 Main PID: 9985 (sackd)
   CGroup: /system.slice/sackd.service
           ##9985 /opt/aws/pcs/scheduler/slurm-25.05/sbin/sackd --systemd --conf-
server=10.3.149.220:6817

Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Starting Slurm auth and cred
kiosk daemon...
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Started Slurm auth and cred
kiosk daemon.
Dec 17 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-25.05/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-25.05/bin/srun --nodes=1 -p all --pty bash -i
```

Menghubungkan node login mandiri ke beberapa cluster di PCS AWS

`pcs-multi-cluster-login-configure.sh` Skrip menyediakan cara otomatis untuk mengkonfigurasi beberapa sackd daemon Slurm pada satu node login mandiri. Hal ini memungkinkan node login untuk berkomunikasi dengan beberapa cluster. Skrip mengotomatiskan operasi berikut:

- Menggunakan tindakan AWS PCS API untuk mendapatkan informasi cluster
- Meminta kunci otentikasi Slurm yang dikodekan base64
- Membuat file Slurm JWKS dengan kunci otentikasi cluster
- Mengkonfigurasi sackd layanan dengan titik akhir cluster dan port
- Membuat file systemd layanan untuk daemon khusus cluster sackd
- Menghasilkan skrip aktivasi untuk pengaturan lingkungan cluster
- Mengaktifkan dan memulai sackd layanan

Note

Skrip ini membutuhkan Slurm versi 25.05 atau yang lebih baru.

Slurm harus sudah diinstal pada instance (setara dengan [langkah 3](#) dalam proses manual). Instance harus dapat mencapai titik akhir cluster target. Skrip melakukan operasi setara [langkah 4](#) dan [langkah 5](#) dalam proses konfigurasi manual. Secara otomatis mendapatkan informasi cluster, mengkonfigurasi sackd layanan, membuat file systemd layanan yang diperlukan, dan membuat skrip aktivasi yang dapat digunakan pengguna untuk mengkonfigurasi lingkungan shell mereka untuk interaksi cluster.

Topik

- [Prasyarat untuk skrip konfigurasi node login AWS multi-cluster PCS](#)
- [AWS Kode skrip konfigurasi node login multi-cluster PCS](#)
- [Menggunakan skrip konfigurasi node login multi-cluster AWS PCS](#)

Prasyarat untuk skrip konfigurasi node login AWS multi-cluster PCS

Persyaratan sistem

- Linux OS dengan systemd dukungan
- Hak akses root untuk konfigurasi sistem

Perintah dan paket yang diperlukan

- `bash`— Penerjemah Shell (versi 4.0+)
- `curl`— Untuk pengambilan AWS metadata IMDS v2
- `jq`— Prosesor JSON untuk mengurai respons API AWS
- `aws`— AWS CLI v2 untuk menjalankan tindakan AWS PCS API dan untuk akses Secrets Manager
- `systemctl`— manajemen `systemd` layanan
- `find`— Utilitas pencarian sistem file
- `grep`— Pencocokan pola teks
- `sed`— Editor aliran untuk manipulasi teks
- `sort`— Utilitas penyortiran teks
- `tail`— Menampilkan baris terakhir dari sebuah file
- `mkdir`— Pembuatan direktori
- `chmod`— Mengubah izin file
- `chown`— Mengubah kepemilikan file
- `ldconfig`— Konfigurasi tautan dinamis

AWS persyaratan

- Cluster AWS PCS yang menjalankan Slurm versi 25.05 atau yang lebih baru
- AWS kredensial yang dikonfigurasi (melalui peran IAM, file kredensial, atau variabel lingkungan)
- Izin untuk:
 - `pcs:GetCluster`
 - `secretsmanager:GetSecretValue` (jika Anda menggunakan rahasia alternatif)

Pengguna dan grup sistem

- `slurm` Pengguna dan grup harus ada di sistem

Instalasi slurm

- Slurm harus dipasang di lokasi yang sama dengan paket installer AWS PCS Slurm:

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

AWS Kode skrip konfigurasi node login multi-cluster PCS

Simpan kode sumber berikut ke file dengan nama berikut:

```
pcs-multi-cluster-login-configure.sh
```

Kode sumber skrip

```
#!/bin/bash
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# AWS PCS Multi-Cluster Standalone Login Node Configuration Script
#
# This script configures AWS Parallel Computing Service (PCS) multi-cluster stand alone
# login nodes
# by setting up the Slurm authentication and credential kiosk daemon (sackd)
# for connecting to remote PCS clusters.
#
# Prerequisites:
# - AWS CLI configured with appropriate permissions
# - Slurm version 25.05 or later
# - Root privileges for system configuration
# - Network connectivity to AWS PCS endpoints

set -eo pipefail

# Function to display usage
usage() {
    echo "Usage: $0 --cluster-identifier <cluster-identifier> [--endpoint-url
<endpoint-url>]"
    echo "    $0 -h|--help"
}

# Function to display help
help() {
    echo "AWS PCS Multi-Cluster Standalone Login Node Configuration Script"
    echo "=====
```

```

    echo
    echo "This script configures multi-cluster standalone login node for AWS Parallel
Computing Service (PCS)"
    echo "by setting up the Slurm authentication and credential kiosk daemon (sackd)."
    echo
    usage
    echo
    echo "Options:"
    echo "  --cluster-identifier <id>      AWS PCS cluster identifier (required)"
    echo "  --endpoint-url <url>           Custom PCS endpoint URL (optional)"
    echo "  -h, --help                       Show this help message"
    echo
    echo "Examples:"
    echo "  $0 --cluster-identifier my-pcs-cluster"
    echo
    echo "Note: This script requires root privileges and Slurm version 25.05 or later."
}

# Function to retrieve authentication key
get_auth_key() {
    if [ "$ALTERNATE_SECRET_RETRIEVAL" = "true" ]; then
        echo "Retrieving authentication key from AWS Secrets Manager..." >&2
        local auth_key_arn=$(echo "$CLUSTER_INFO" | jq -r
'.cluster.slurmConfiguration.authKey.secretArn')
        local auth_key_version=$(echo "$CLUSTER_INFO" | jq -r
'.cluster.slurmConfiguration.authKey.secretVersion')

        if [ "$auth_key_arn" = "null" ] || [ "$auth_key_version" = "null" ]; then
            echo "Error: Auth key information not found in cluster configuration" >&2
            exit 1
        fi

        if ! aws secretsmanager get-secret-value --secret-id "$auth_key_arn" --version-
id "$auth_key_version" --query SecretString --output text --region "$REGION" 2>/dev/
null; then
            echo "Error: Failed to retrieve auth key from Secrets Manager" >&2
            exit 1
        fi
    else
        echo "Please enter the base64-encoded Slurm authentication key:" >&2
        echo -n "Base64 of the Slurm secret key: " >&2
        local key
        read -rs key
        echo >&2
    fi
}

```

```

    echo "$key"
  fi
}

# Function to get next available SACKD port
get_next_sackd_port() {
  local exclude_file="$1"
  local port=6918
  local used_ports=()

  # Get all currently used SACKD ports into an array
  while IFS= read -r line; do
    used_ports+=("$line")
  done < <(find /etc/sysconfig -name "sackd-pcs-*" ! -path "$exclude_file" \
    -exec grep SACKD_PORT= '{}' ';' 2>/dev/null | \
    sed 's/.*SACKD_PORT=//' | sort -n)

  # Loop through used ports to find first available port
  for used_port in "${used_ports[@]"; do
    if [ "$port" -lt "$used_port" ]; then
      break
    elif [ "$port" -eq "$used_port" ]; then
      ((port++))
    fi
  done

  echo "$port"
}

# Function to configure cluster
configure_cluster() {
  mkdir -p /etc/slurm
  SLURM_JWKS_FILE="/etc/slurm/slurm-`${CLUSTER_NAME}`.jwks"
  echo '{"keys":
[{"alg":"HS256","kty":"oct","kid":"key-`${CLUSTER_ID}`","k":"`${BASE64_SLURM_KEY}`"}]}'
  | jq -c '.' > "${SLURM_JWKS_FILE}"

  chmod 0600 "$SLURM_JWKS_FILE"
  chown slurm:slurm "$SLURM_JWKS_FILE"

  SLURM_INSTALL_PATH="/opt/aws/pcs/scheduler/slurm-`${SLURM_VERSION}`"

  SACKD_RUNTIME_DIRECTORY="/run/slurm-`${CLUSTER_NAME}`"
  mkdir -p "${SACKD_RUNTIME_DIRECTORY}"

```

```

chown slurm:slurm "${SACKD_RUNTIME_DIRECTORY}"

mkdir -p /etc/sysconfig
SACKD_SERVICE_NAME="sackd-pcs-${CLUSTER_NAME}"
SACKD_SERVICE_ENV="/etc/sysconfig/${SACKD_SERVICE_NAME}"
SACKD_PORT=$(get_next_sackd_port "${SACKD_SERVICE_ENV}")
cat > "${SACKD_SERVICE_ENV}" << EOF
SACKD_OPTIONS='--conf-server=$ENDPOINTS'
SLURM_SACK_JWKS='$SLURM_JWKS_FILE'
RUNTIME_DIRECTORY='$SACKD_RUNTIME_DIRECTORY'
SACKD_PORT=$SACKD_PORT
EOF

SACKD_SERVICE_PATH="/etc/systemd/system/${SACKD_SERVICE_NAME}.service"

cat << EOF > "${SACKD_SERVICE_PATH}"
[Unit]
Description=Slurm auth and cred kiosk daemon
After=network-online.target remote-fs.target
Wants=network-online.target
ConditionPathExists=${SACKD_SERVICE_ENV}

[Service]
Type=notify
EnvironmentFile=${SACKD_SERVICE_ENV}
User=slurm
Group=slurm
RuntimeDirectory=slurm-${CLUSTER_NAME}
RuntimeDirectoryMode=0755
ExecStart=${SLURM_INSTALL_PATH}/sbin/sackd --systemd \${SACKD_OPTIONS}
ExecReload=/bin/kill -HUP \${MAINPID}
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF

chown root:root "${SACKD_SERVICE_PATH}"
chmod 0644 "${SACKD_SERVICE_PATH}"
systemctl daemon-reload && systemctl enable "${SACKD_SERVICE_NAME}"
systemctl restart "${SACKD_SERVICE_NAME}"

```

```

    ACTIVATE_SCRIPT="activate-pcs-`${CLUSTER_NAME}`"
    cat > "$ACTIVATE_SCRIPT" << EOF
# Activate script for Slurm cluster `${CLUSTER_NAME}`

# Add Slurm paths
export PATH="`${SLURM_INSTALL_PATH}`/bin:\$PATH"
export MANPATH="`${SLURM_INSTALL_PATH}`/share/man:\$MANPATH"
export LD_LIBRARY_PATH="`${SLURM_INSTALL_PATH}`/lib:\$LD_LIBRARY_PATH"
ldconfig

# Set Slurm configuration
export SLURM_CONF="/run/slurm-`${CLUSTER_NAME}`/conf/slurm.conf"
export PCS_CLUSTER_NAME="`${CLUSTER_NAME}`"
export PCS_CLUSTER_IDENTIFIER="`${CLUSTER_IDENTIFIER}`"
export PCS_CLUSTER_ID="`${CLUSTER_ID}`"

echo "Activated PCS cluster environment: `${CLUSTER_NAME}`"

# Deactivate function
function deactivate-pcs-`${CLUSTER_NAME}`() {
    export PATH="\$(echo "\$PATH" | sed -e "s|`${SLURM_INSTALL_PATH}`/bin:||g" -e "s|:
`${SLURM_INSTALL_PATH}`/bin||g" -e "s|^`${SLURM_INSTALL_PATH}`/bin\$||")"
    export MANPATH="\$(echo "\$MANPATH" | sed -e "s|`${SLURM_INSTALL_PATH}`/share/man:||
g" -e "s|:`${SLURM_INSTALL_PATH}`/share/man||g" -e "s|^`${SLURM_INSTALL_PATH}`/share/man\
\$||")"
    export LD_LIBRARY_PATH="\$(echo "\$LD_LIBRARY_PATH" | sed -e "s|
`${SLURM_INSTALL_PATH}`/lib:||g" -e "s|:`${SLURM_INSTALL_PATH}`/lib||g" -e "s|^
`${SLURM_INSTALL_PATH}`/lib\$||")"
    unset SLURM_CONF
    unset PCS_CLUSTER_NAME
    unset PCS_CLUSTER_IDENTIFIER
    unset PCS_CLUSTER_ID
    unset -f deactivate-pcs-`${CLUSTER_NAME}`
    ldconfig
    echo "Deactivated PCS cluster environment: `${CLUSTER_NAME}`"
}

export -f deactivate-pcs-`${CLUSTER_NAME}`

EOF
}

# Main function

```

```
main() {
    # Parse arguments
    CLUSTER_IDENTIFIER=""
    PCS_ENDPOINT_URL=""

    while [ "$1" != "" ]; do
        case $1 in
            --cluster-identifier)
                shift
                CLUSTER_IDENTIFIER="$1"
                ;;
            --endpoint-url)
                shift
                PCS_ENDPOINT_URL="--endpoint-url $1"
                ;;
            -h|--help)
                help
                exit 0
                ;;
            *)
                echo "Invalid argument: $1" >&2
                usage >&2
                exit 1
                ;;
        esac
        shift
    done

    # Validate required arguments
    if [ -z "$CLUSTER_IDENTIFIER" ]; then
        echo "Error: --cluster-identifier is required" >&2
        usage >&2
        exit 1
    fi

    # Validate running as root
    if [ "$EUID" -ne 0 ]; then
        echo "Error: This script must be run as root" >&2
        exit 1
    fi

    # Validate required commands are available
    for cmd in aws jq curl; do
        if ! command -v "$cmd" &> /dev/null; then
```

```
        echo "Error: Required command '$cmd' not found" >&2
        exit 1
    fi
done

# Get the region name from IMDS v2 with error handling (try IPv6 first, fallback to
IPv4)
echo "Retrieving AWS region from instance metadata..."
# Try IPv6 IMDS endpoint first (fd00:ec2::254) with fast timeout (1s connect, 2s
total)
# If IPv6 fails, fallback to IPv4 IMDS endpoint (169.254.169.254)
IMDS_ENDPOINT="http://[fd00:ec2::254]"
if ! TOKEN=$(curl -s -X PUT "${IMDS_ENDPOINT}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --connect-timeout 1 --max-time 2 2>/dev/null); then
    IMDS_ENDPOINT="http://169.254.169.254"
    if ! TOKEN=$(curl -s -X PUT "${IMDS_ENDPOINT}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --max-time 5); then
        echo "Error: Failed to retrieve IMDS token. Ensure this script is running
on an EC2 instance." >&2
        exit 1
    fi
fi

if ! REGION=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" "${IMDS_ENDPOINT}/
latest/dynamic/instance-identity/document" --max-time 5 | jq -r '.region'); then
    echo "Error: Failed to retrieve AWS region from instance metadata" >&2
    exit 1
fi

echo "Detected AWS region: $REGION"

# Retrieve cluster information from AWS PCS
echo "Retrieving cluster information for: $CLUSTER_IDENTIFIER"
# shellcheck disable=SC2086
if ! CLUSTER_INFO=$(aws pcs get-cluster --region "$REGION" --cluster-identifier
"$CLUSTER_IDENTIFIER" $PCS_ENDPOINT_URL 2>/dev/null); then
    echo "Error: Failed to retrieve cluster information. Check cluster identifier
and AWS permissions." >&2
    exit 1
fi

CLUSTER_ID=$(echo "$CLUSTER_INFO" | jq -r '.cluster.id')
CLUSTER_NAME=$(echo "$CLUSTER_INFO" | jq -r '.cluster.name')
SLURM_VERSION=$(echo "$CLUSTER_INFO" | jq -r '.cluster.scheduler.version')
```

```
SLURM_VERSION=${SLURM_VERSION#Slurm_}

# Check if Slurm version is >= 25.05
# shellcheck disable=SC2072
if [[ "$SLURM_VERSION" < "25.05" ]]; then
    echo "Error: This script requires Slurm version 25.05 or later. Found version:
$SLURM_VERSION" >&2
    exit 1
fi

ENDPOINTS=$(echo "$CLUSTER_INFO" | jq -r '.cluster.endpoints[] | select(.type
== "SLURMCTLD") | (if .privateIpAddress != "" then .privateIpAddress else "["
+ .ipv6Address + "]" end) + ":" + .port' | tr '\n' ',' | sed 's/,,$//')

# Get BASE64_SLURM_KEY
BASE64_SLURM_KEY=$(get_auth_key)

if [ -z "$BASE64_SLURM_KEY" ]; then
    echo "Error: base64 Slurm key cannot be empty" >&2
    exit 1
fi

configure_cluster

# Final configuration summary
echo "======"
echo "Configuration completed successfully!"
echo "======"
echo "Cluster Name: $CLUSTER_NAME"
echo "Cluster ID: $CLUSTER_ID"
echo "Slurm Version: $SLURM_VERSION"
echo "Service Name: $SACKD_SERVICE_NAME"
echo "SACKD Port: $SACKD_PORT"
echo
echo "To activate this cluster environment, run:"
echo "  source ./$ACTIVATE_SCRIPT"
echo
echo "To deactivate this cluster environment, run:"
echo "  deactivate-pcs-`${CLUSTER_NAME}`"
echo
echo "To check service status:"
echo "  systemctl status $SACKD_SERVICE_NAME"
echo
echo "To view service logs:"
```

```
    echo " journalctl -u $SACKD_SERVICE_NAME -f"
}

# Exit if being sourced for testing
[[ "${BASH_SOURCE[0]}" != "${0}" ]] && return

# Execute main function
main "$@"
```

Menggunakan skrip konfigurasi node login multi-cluster AWS PCS

Menjalankan skrip

Untuk menjalankan skrip konfigurasi

1. Simpan [konten skrip dalam](#) file bernama:

```
pcs-multi-cluster-login-configure.sh
```

2. Jadikan itu dapat dieksekusi:

```
chmod +x pcs-multi-cluster-login-configure.sh
```

3. Jalankan skrip .

```
./pcs-multi-cluster-login-configure.sh --cluster-identifier cluster-name
```

Lingkungan interaksi cluster

Setelah konfigurasi berhasil, skrip menghasilkan skrip aktivasi khusus cluster di direktori saat ini. Script memiliki nama `activate-pcs-cluster-name`. Skrip aktivasi mengkonfigurasi variabel lingkungan dan jalur yang diperlukan untuk berinteraksi dengan cluster target.

Untuk mengaktifkan lingkungan cluster

- Gunakan `source` perintah untuk menjalankan skrip aktivasi

```
source ./activate-pcs-cluster-name
```

Example

```
# Activate cluster environment for cluster 'my-cluster'
source ./activate-pcs-my-cluster

# Now you can use Slurm commands
sinfo
squeue
sbatch my-job.sh
```

Apa yang dilakukan skrip aktivasi

- Menetapkan variabel `SLURM_CONF` lingkungan untuk menunjuk ke konfigurasi cluster.
- Memperbarui `PATH` untuk menyertakan binari Slurm cluster.
- Mengkonfigurasi variabel lingkungan Slurm lain yang diperlukan (`MANPATH`,). `LD_LIBRARY_PATH`
- Menetapkan variabel identifikasi cluster AWS PCS.
- Memungkinkan interaksi tanpa batas dengan cluster AWS PCS target.

Untuk menonaktifkan lingkungan cluster

- Jalankan perintah penonaktifan.

```
deactivate-pcs-cluster-name
```

Example

```
# After activating a cluster
source ./activate-pcs-my-cluster

# Work with the cluster
sinfo

# Deactivate when done
deactivate-pcs-my-cluster
```

Apa yang dilakukan perintah nonaktifkan

- Mengembalikan variabel PATH lingkungan asli.
- Menghapus set variabel lingkungan Slurm khusus cluster.
- Mengembalikan lingkungan shell ke status pra-aktivasi.

Note

Aktivasi khusus sesi dan harus bersumber di sesi shell tempat Anda ingin berinteraksi dengan cluster.

AWS Jaringan PCS

Cluster AWS PCS Anda dibuat dalam VPC Amazon. 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 komputasi AWS PCS. Untuk informasi selengkapnya tentang template peluncuran, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).

Topik

- [AWS PCS VPC dan persyaratan subnet dan pertimbangan](#)
- [Membuat VPC untuk klaster PCS Anda AWS](#)
- [Grup keamanan di AWS PCS](#)
- [Beberapa antarmuka jaringan di AWS PCS](#)
- [Grup penempatan untuk instans EC2 di PCS AWS](#)
- [Menggunakan Elastic Fabric Adapter \(EFA\) dengan PCS AWS](#)

AWS PCS VPC dan persyaratan subnet dan pertimbangan

Saat Anda membuat cluster AWS PCS, Anda menentukan VPC subnet di VPC itu. Topik ini memberikan gambaran umum tentang persyaratan dan pertimbangan khusus AWS PCS untuk VPC dan subnet yang Anda gunakan dengan cluster Anda. Jika Anda tidak memiliki VPC untuk digunakan dengan AWS PCS, Anda dapat membuatnya menggunakan template AWS yang CloudFormation disediakan. Untuk informasi selengkapnya VPCs, lihat [Virtual private cloud \(VPC\)](#) di Panduan Pengguna Amazon VPC.

Persyaratan dan pertimbangan VPC

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

- VPC harus memiliki cukup banyak alamat IP yang tersedia untuk cluster, node apa pun, dan sumber daya cluster lain yang ingin Anda buat. Untuk informasi selengkapnya, lihat [Pengalaman IP untuk subnet Anda VPCs dan subnet](#) di Panduan Pengguna Amazon VPC.
- Jika klaster Anda menggunakan IPv6:

- Kaitkan blok IPv6 CIDR dengan VPC Anda. Untuk informasi selengkapnya, lihat [Membuat VPC](#) di Panduan Pengguna Amazon VPC.

⚠ Important

Meskipun Anda dapat mengonfigurasi VPC Anda dengan keduanya IPv4 dan IPv6, Anda hanya dapat memilih 1 jenis jaringan untuk cluster Anda.

- Aktifkan IPv6 alamat penetapan otomatis untuk subnet Anda.
- Untuk informasi lebih lanjut, lihat:
 - [IPv6 pada AWS](#)
 - [Memahami IPv6 pengalamatan di AWS dan merancang rencana pengalamatan yang dapat diskalakan](#)
- VPC harus memiliki nama host DNS dan dukungan resolusi DNS. Jika tidak, node tidak dapat mendaftarkan cluster pelanggan. Untuk informasi selengkapnya, lihat [Atribut DNS untuk VPC Anda](#) dalam Panduan Pengguna Amazon VPC.
- VPC mungkin memerlukan titik akhir VPC AWS PrivateLink untuk dapat menghubungi PCS API. AWS Untuk informasi selengkapnya, lihat [Menyambungkan VPC ke layanan yang digunakan AWS PrivateLink](#) di Panduan Pengguna Amazon VPC.

⚠ Important

AWS PCS tidak mendukung VPC dengan penyewaan instans khusus. VPC yang Anda gunakan untuk AWS PCS harus menggunakan penyewaan default instance. Anda dapat mengubah penyewaan instance untuk VPC yang ada. Untuk informasi selengkapnya, lihat [Mengubah penyewaan instans VPC](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Persyaratan dan pertimbangan subnet

Saat Anda membuat cluster Slurm, AWS PCS membuat [Elastic Network Interface \(ENI\)](#) di subnet yang Anda tentukan. Antarmuka jaringan ini memungkinkan komunikasi antara pengontrol penjadwal dan VPC pelanggan. Antarmuka jaringan juga memungkinkan Slurm untuk berkomunikasi dengan komponen yang digunakan di akun Anda. 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.
- Jika cluster Anda menggunakan IPv6, semua subnet di cluster Anda harus menggunakan IPv6.

Important

Grup node komputasi yang dikonfigurasi dengan sampel AWS PCS AMIs dan beberapa antarmuka jaringan tidak akan berfungsi saat ini jika subnet hanya dikonfigurasi untuk digunakan. IPv6 Gunakan subnet dual-stack (IPv4 and IPv6) atau IPv4 -only subnet sebagai gantinya. Untuk informasi selengkapnya, lihat [Menggunakan sampel Amazon Machine Images \(AMIs\) dengan 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 klaster AWS PCS, dan ke subnet lain di VPC yang sama.

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 klaster Anda menggunakan IPv4 dan Anda berencana untuk menyebarkan node ke subnet publik, subnet tersebut harus menetapkan alamat publik secara otomatis IPv4 .

Note

Contoh di subnet publik harus menggunakan grup keamanan dengan aturan masuk yang mengizinkan lalu lintas dari alamat IP publik. Kecuali Anda memiliki batasan alamat sumber tertentu, ini berarti alamat IPv4 sumber 0.0.0.0/0 atau alamat IPv6 sumber: ::/0.

- Jika subnet tempat Anda menyebarkan node adalah subnet pribadi dan tabel rutenya tidak menyertakan rute ke [perangkat terjemahan alamat jaringan \(NAT\)](#) (IPv4), tambahkan titik akhir VPC menggunakan VPC pelanggan. AWS PrivateLink Titik akhir VPC diperlukan untuk semua AWS layanan yang dihubungi node. Satu-satunya titik akhir yang diperlukan adalah AWS PCS mengizinkan node memanggil tindakan RegisterComputeNodeGroupInstance API. Untuk informasi selengkapnya, lihat [RegisterComputeNodeGroupInstance](#) di Referensi AWS PCS API.
- Status subnet publik atau pribadi tidak memengaruhi AWS PCS; titik akhir yang diperlukan harus dapat dijangkau.

Membuat VPC untuk kluster PCS Anda AWS

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

Gunakan Amazon VPC untuk meluncurkan sumber daya VPC 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 layanan VPC Amazon sebelum menerapkan cluster VPC produksi. Untuk informasi selengkapnya, lihat [Apa itu Amazon VPC?](#) dalam mode visual penulis. Panduan Pengguna Amazon VPC.

Cluster PCS, node, dan sumber daya pendukung (seperti sistem file dan layanan direktori) digunakan dalam VPC Amazon Anda. Jika Anda ingin menggunakan VPC Amazon yang ada dengan PCS, itu harus memenuhi persyaratan yang dijelaskan dalam [AWS PCS VPC dan persyaratan subnet dan pertimbangan](#). Topik ini menjelaskan cara membuat VPC yang memenuhi persyaratan PCS menggunakan templat yang disediakan AWS. CloudFormation 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 VPC Amazon untuk PC, Anda harus memiliki izin IAM yang diperlukan untuk membuat sumber daya Amazon VPC. Sumber daya ini adalah VPCs, subnet, grup keamanan, tabel dan rute rute, dan gateway internet dan NAT. Untuk informasi selengkapnya, lihat [Membuat VPC dengan subnet publik](#) di Panduan Pengguna Amazon VPC. Untuk meninjau daftar lengkap Amazon EC2, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon EC2](#) di Referensi Otorisasi Layanan.

Buat VPC Amazon

Buat VPC dengan menyalin dan menempelkan URL yang sesuai untuk AWS Region tempat Anda akan menggunakan PCS. Anda juga dapat mengunduh CloudFormation templat dan mengunggahnya sendiri ke [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 VPC Amazon untuk PCS

1. Buka template di [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, masukkan `CidrBlock10.3.0.0/16`
 - 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 Capabilities, centang kotak untuk Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM.

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

Note

Untuk melihat semua sumber daya yang dibuat CloudFormation template, buka [CloudFormation konsol](#). Pilih hpc-networking tumpukan dan kemudian pilih tab Sumber Daya.

Grup keamanan di AWS PCS

Grup keamanan di Amazon EC2 bertindak sebagai firewall virtual untuk mengontrol lalu lintas masuk dan keluar ke instans. Gunakan template peluncuran untuk grup node komputasi AWS PCS 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, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).

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

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

Contoh konfigurasi grup keamanan permisif

IPv4

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

IPv6

Jenis aturan	Protokol	Port	Sumber	Destinasi
Ke dalam	Semua	Semua	Diri Sendiri	
Ke luar	Semua	Semua		::/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 instans EC2
- 6818 - aktifkan koneksi keluar dari `slurmctld` hingga `slurmd` berjalan pada instans EC2

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

- 6817 - aktifkan koneksi keluar `slurmctld` dari instans EC2.

- 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 `srtn`
- Lalu lintas EFA antara instance grup node. Untuk informasi selengkapnya, lihat [Mempersiapkan grup keamanan berkemampuan EFA](#) di Panduan Pengguna untuk Instans Linux
- Lalu lintas antar simpul lain yang diperlukan oleh beban kerja Anda

Beberapa antarmuka jaringan di AWS PCS

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

Konfigurasi kartu jaringan tambahan untuk instance dalam grup node komputasi AWS PCS dengan menambahkan antarmuka jaringan ke templat peluncuran EC2-nya. 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 komputasi AWS PCS yang akan menggunakan template peluncuran.
- Perangkat jaringan utama, di mana komunikasi jaringan rutin seperti SSH dan lalu lintas HTTPS akan terjadi, ditetapkan dengan menetapkan `aDeviceIndex`. `0` Antarmuka jaringan lainnya memiliki `fileDeviceIndex`. `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 template peluncuran.
- Grup keamanan untuk setiap antarmuka jaringan diatur menggunakan `Groups`. Dalam contoh ini, grup keamanan SSH 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 instans EC2 di PCS AWS

Anda dapat menggunakan grup penempatan untuk memengaruhi penempatan instans EC2 agar sesuai dengan kebutuhan beban kerja yang berjalan padanya.

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 instans Amazon EC2 Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Kami menyarankan Anda menyertakan grup penempatan cluster saat Anda mengonfigurasi grup node komputasi AWS PCS 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 URL berikut 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 peluncuran EC2 untuk grup node komputasi AWS PCS.

Menggunakan Elastic Fabric Adapter (EFA) dengan PCS AWS

Elastic Fabric Adapter (EFA) adalah interkoneksi jaringan canggih berkinerja tinggi AWS yang dapat Anda pasang ke instans EC2 untuk mempercepat aplikasi High Performance Computing (HPC) dan machine learning. Mengaktifkan aplikasi Anda berjalan pada cluster AWS PCS dengan EFA melibatkan konfigurasi instance grup node komputasi AWS PCS untuk menggunakan EFA sebagai berikut.

Note

Instal EFA pada AMI yang AWS kompatibel dengan PC — AMI yang digunakan dalam AWS grup node komputasi PCS harus memiliki driver EFA yang diinstal dan dimuat. Untuk informasi tentang cara membuat AMI kustom dengan perangkat lunak EFA yang diinstal, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Daftar Isi

- [Identifikasi instans EC2 yang mendukung EFA](#)
- [Buat grup keamanan untuk mendukung komunikasi EFA](#)

- [\(Opsional\) Buat grup penempatan](#)
- [Membuat atau memperbarui template peluncuran EC2](#)
- [Membuat atau memperbarui grup node komputasi untuk EFA](#)
- [\(Opsional\) Uji EFA](#)
- [\(Opsional\) Gunakan CloudFormation templat untuk membuat templat peluncuran berkemampuan EFA](#)

Identifikasi instans EC2 yang mendukung EFA

Untuk menggunakan EFA, semua jenis instans yang diizinkan untuk grup komputasi AWS PCS harus mendukung EFA, dan harus memiliki jumlah v yang sama CPUs (dan GPUs jika sesuai). Untuk daftar instans yang mendukung EFA, lihat [Adaptor Kain Elastis untuk beban kerja HPC dan ML di Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud](#). Anda juga dapat menggunakan AWS CLI untuk melihat daftar jenis instance yang mendukung EFA. Ganti *region-code* dengan AWS Region tempat Anda menggunakan AWS PCS, seperti `us-east-1`.

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

Note

Tentukan berapa banyak antarmuka jaringan yang tersedia — Beberapa instans EC2 memiliki beberapa kartu jaringan. Ini memungkinkan mereka untuk memiliki banyak EFAs. Untuk informasi selengkapnya, lihat [Beberapa antarmuka jaringan di AWS PCS](#).

Buat grup keamanan untuk mendukung komunikasi EFA

AWS CLI

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

- *region-code*— Tentukan AWS Region tempat Anda menggunakan AWS PCS, seperti `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 keamanan EFA 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 mendukung EFA. Unduh template dari URL berikut, 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 `fa-sg`.
 - Di bawah VPC, pilih VPC tempat Anda akan menggunakan PCS. AWS

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

(Opsional) Buat grup penempatan

Kami menyarankan Anda meluncurkan semua instance yang menggunakan EFA dalam grup penempatan cluster untuk meminimalkan jarak fisik di antara mereka. Buat grup penempatan untuk setiap grup node komputasi tempat Anda berencana menggunakan EFA. Lihat [Grup penempatan untuk instans EC2 di PCS AWS](#) untuk membuat grup penempatan untuk grup node komputasi Anda.

Membuat atau memperbarui template peluncuran EC2

Antarmuka jaringan EFA diatur dalam template peluncuran EC2 untuk grup node komputasi AWS PCS. Jika ada beberapa kartu jaringan, beberapa EFAs dapat dikonfigurasi. Grup keamanan EFA 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 antarmuka EFA. Setiap EFA membutuhkan grup keamanan yang memungkinkan lalu lintas EFA (`sg-EfaSecGroupId`). Kelompok keamanan lain, terutama yang menangani lalu lintas reguler seperti SSH atau HTTPS, hanya perlu dilampirkan ke antarmuka jaringan utama (ditunjuk oleh `a DeviceIndex0`). Luncurkan templat tempat antarmuka jaringan ditentukan tidak mendukung pengaturan grup keamanan menggunakan SecurityGroupIds parameter—Anda harus menetapkan nilai untuk setiap antarmuka jaringan yang Anda Groups konfigurasi.

```
{
  "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 untuk EFA

Grup node komputasi AWS PCS Anda harus berisi instance yang memiliki jumlah vCPUs, arsitektur prosesor, dan dukungan EFA yang sama. Konfigurasi grup node komputasi untuk menggunakan AMI dengan perangkat lunak EFA yang diinstal di dalamnya, dan untuk menggunakan templat peluncuran yang mengonfigurasi antarmuka jaringan berkemampuan EFA.

(Opsional) Uji EFA

Anda dapat mendemonstrasikan komunikasi yang mendukung EFA antara dua node dalam grup node komputasi dengan menjalankan `fi_pingpong` program, yang termasuk dalam instalasi perangkat lunak EFA. Jika tes ini berhasil, kemungkinan 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=25.05.4
  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=25.05.4
  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 (atau SSM). Perhatikan bahwa ini adalah alamat IP internal, jadi Anda mungkin perlu terhubung dari salah satu node login Anda jika Anda menggunakan SSH. Perlu diketahui juga bahwa instance perlu dikonfigurasi dengan kunci SSH 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 berkemampuan EFA

Karena ada beberapa dependensi untuk menyiapkan EFA, CloudFormation template telah disediakan yang dapat Anda gunakan untuk mengkonfigurasi grup node komputasi. Ini mendukung instance 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 URL berikut, lalu unggah ke CloudFormation konsol di AWS Region tempat Anda menggunakan AWS PCS.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/pcs-1t-efa.yaml
```

Dengan templat terbuka di 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 komputasi AWS PCS Anda, seperti ***NODEGROUPNAME***-efa-1t
- Di bawah Parameter
 - Di bawah NumberOfNetworkCards, pilih jumlah kartu jaringan dalam instance yang akan ada di grup node Anda.
 - Di bawah VpcId, pilih VPC tempat cluster AWS PCS Anda digunakan.
 - Di bawah NodeGroupSubnetId, pilih subnet di VPC klaster Anda tempat instans berkemampuan EFA 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 cluster dan ke AWS PCS API. Banyak pelanggan memilih grup keamanan default dari VPC cluster mereka.
 - Di bawah SshSecurityGroupId, berikan ID untuk grup keamanan yang Anda gunakan untuk mengizinkan akses SSH masuk ke node di cluster Anda.
 - Untuk SshKeyName, pilih keypair SSH untuk akses ke node di cluster Anda.
 - Untuk LaunchTemplateName, masukkan nama deskriptif untuk template peluncuran seperti ***NODEGROUPNAME***-efa-1t. Nama harus unik untuk Anda Akun AWS di AWS Region mana Anda akan menggunakan AWS PCS.
- Di bawah Kemampuan
 - Centang kotak untuk saya akui yang AWS CloudFormation mungkin membuat sumber daya IAM.

Pantau status CloudFormation tumpukan. Ketika mencapai CREATE_COMPLETE template peluncuran siap untuk digunakan. Gunakan dengan grup node komputasi AWS PCS, seperti dijelaskan di atas dalam [Membuat atau memperbarui grup node komputasi untuk EFA](#).

Menggunakan sistem file jaringan dengan AWS PCS

Anda dapat melampirkan sistem file 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 sistem file yang disediakan oleh AWS layanan, termasuk Amazon Elastic File System \(Amazon EFS\), Amazon FSx untuk Lustre, Amazon untuk NetApp ONTAP, Amazon FSx untuk OpenZFS, dan Amazon File Cache.](#) Anda juga dapat menggunakan sistem file yang dikelola sendiri, seperti server NFS.

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 sistem file jaringan bersama dan instance grup node komputasi.
- Aturan grup keamanan untuk sistem file jaringan 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 kluster PCS Anda mungkin mengalami kesalahan izin.
- Pemasangan sistem file dilakukan menggunakan templat EC2 peluncuran. Kesalahan atau batas waktu dalam memasang sistem file jaringan dapat mencegah instance tersedia untuk menjalankan pekerjaan. Ini, pada gilirannya, dapat menyebabkan biaya yang tidak terduga. Untuk informasi selengkapnya tentang men-debug templat peluncuran, lihat [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).

Contoh pemasangan jaringan

Anda dapat membuat sistem file menggunakan Amazon EFS, Amazon FSx untuk Lustre, Amazon untuk NetApp ONTAP, Amazon FSx untuk 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 file Amazon EFS. Pastikan ia memiliki target mount di setiap Availability Zone tempat Anda akan meluncurkan instance grup node komputasi PCS. Pastikan juga setiap target pemasangan dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node komputasi PCS. 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 `ccloud-config` mekanisme untuk memasang sistem file Amazon EFS. 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 file EFS

```
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 tempat Anda akan menggunakan PCS. AWS Untuk meminimalkan transfer antar zona, terapkan di subnet di Availability Zone yang sama di mana Anda akan meluncurkan sebagian besar instance grup node komputasi PCS Anda. Pastikan sistem file dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node komputasi PCS. 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 AWS Region mana sistem file FSx for Lustre digunakan (harus sama dengan sistem PCS Anda AWS)
- (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 NetApp ONTAP

Pengaturan sistem file

Buat Amazon FSx untuk sistem file NetApp ONTAP di VPC tempat Anda akan AWS menggunakan PCS. Untuk meminimalkan transfer antar zona, terapkan di subnet di Availability Zone yang sama di mana Anda akan meluncurkan sebagian besar instance grup node komputasi AWS PCS Anda. Pastikan sistem file dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node komputasi AWS PCS. Untuk informasi selengkapnya tentang grup keamanan, lihat [Kontrol Akses Sistem File dengan Amazon VPC](#) di Panduan Pengguna FSx ONTAP.

Luncurkan template

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

- *mount-point-directory*— Jalur pada instance di mana Anda ingin memasang volume FSx ONTAP Anda
- *svm-id*— ID SVM untuk sistem file FSx ONTAP
- *filesystem-id*— ID sistem file FSx untuk sistem file ONTAP
- *region-code*— Di AWS Region mana sistem file FSx untuk ONTAP digunakan (harus sama dengan sistem AWS PCS Anda)
- *volume-name*— Nama volume FSx untuk ONTAP

```
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 svm-id.filesystem-id.fsx.region-code.amazonaws.com:/volume-name /mount-point-directory

--==MYBOUNDARY==
```

Amazon FSx untuk OpenZFS

Pengaturan sistem file

Buat sistem file FSx untuk OpenZFS di VPC di mana Anda akan menggunakan PCS. AWS Untuk meminimalkan transfer antar zona, terapkan di subnet di Availability Zone yang sama di mana Anda

akan meluncurkan sebagian besar instance grup node komputasi AWS PCS Anda. Pastikan sistem file dikaitkan dengan grup keamanan yang memungkinkan akses masuk dan keluar dari instance grup node komputasi AWS PCS. Untuk informasi selengkapnya tentang grup keamanan, lihat [Mengelola akses sistem file dengan Amazon VPC di Panduan Pengguna FSx](#) untuk OpenZFS.

Luncurkan template

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

- *mount-point-directory*— Jalur pada instance di mana Anda ingin memasang untuk berbagi OpenZFS FSx
- *filesystem-id*— ID sistem file untuk sistem file FSx untuk OpenZFS
- *region-code*— Di AWS Region mana sistem file FSx untuk OpenZFS digunakan (harus sama dengan sistem PCS Anda AWS)

```
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,rsync,rsync=1048576,wsync=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 Anda akan menggunakan AWS PCS. Untuk meminimalkan transfer antar zona, pilih subnet di Availability Zone yang sama di mana Anda akan meluncurkan sebagian besar instance grup node komputasi PCS Anda. Pastikan File Cache dikaitkan dengan grup keamanan yang memungkinkan lalu lintas masuk dan keluar pada port 988 antara instance PCS Anda dan Cache File. 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 Domain Name System (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 PCS bekerja 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 AMI yang disediakan oleh dan dikelola oleh AWS. Jika Anda menggunakan AWS PCS dalam produksi, kami sarankan Anda membuat sendiri AMIs. Topik ini mencakup cara menemukan dan menggunakan sampel AMIs, serta cara membuat dan menggunakan kustomisasi Anda sendiri AMIs.

Topik

- [Menggunakan sampel Amazon Machine Images \(AMIs\) dengan AWS PCS](#)
- [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#)
- [Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS](#)
- [Catatan rilis untuk sampel AWS PCS AMIs](#)

Menggunakan sampel Amazon Machine Images (AMIs) dengan AWS PCS

AWS menyediakan [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.

Important

Grup node komputasi yang dikonfigurasi dengan sampel AWS PCS AMIs dan beberapa antarmuka jaringan tidak akan berfungsi saat ini jika subnet hanya dikonfigurasi untuk digunakan. IPv6 Gunakan subnet dual-stack (IPv4 and IPv6) atau IPv4 -only subnet sebagai gantinya.

Temukan sampel AWS PCS saat ini AMIs

Konsol Manajemen AWS

Contoh AWS PCS AMIs memiliki konvensi penamaan berikut:

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

Nilai yang diterima

- *OS* – amzn2
- *architecture* – x86_64 atau arm64
- *scheduler* – slurm
- *scheduler-major-version* – 25.05

Untuk menemukan sampel AWS PCS AMIs

1. Buka [EC2 konsol Amazon](#).
2. Navigasi ke AMIs.
3. Pilih Gambar publik.
4. Di Temukan AMI berdasarkan atribut atau tag, cari AMI menggunakan nama templat.

Contoh

- Contoh AMI untuk Slurm 25.05 pada instance Arm64

```
aws-pcs-sample_ami-amzn2-arm64-slurm-25.05
```

- Contoh AMI untuk Slurm 25.05 pada instans x86

```
aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05
```

Note

Jika ada beberapa AMIs, gunakan AMI dengan cap waktu terbaru.

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

AWS CLI

Anda dapat menemukan sampel AWS PCS terbaru AMI dengan perintah yang mengikuti. Ganti *region-code* dengan AWS Region tempat Anda menggunakan AWS PCS, seperti `us-east-1`.

- x86_64

```
aws ec2 describe-images --region region-code --owners amazon \
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05*' \
          'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

- Arm64

```
aws ec2 describe-images --region region-code --owners amazon \
--filters 'Name=name,Values=aws-pcs-sample_ami-amzn2-arm64-slurm-25.05*' \
          'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

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

Pelajari lebih lanjut tentang sampel AWS PCS AMIs

Untuk melihat konten, detail konfigurasi untuk rilis sampel AWS PCS saat ini dan sebelumnya AMIs, lihat [Catatan rilis untuk sampel AWS PCS AMIs](#).

Bangun sendiri yang AMIs kompatibel dengan AWS PCS

Untuk mempelajari cara membuat sendiri AMIs yang bekerja dengan AWS PCS, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Gambar Mesin Amazon Kustom (AMIs) untuk AWS PCS

AWS PCS dirancang 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 agen AWS PCS dan versi Slurm yang kompatibel diinstal dan dikonfigurasi dengan benar. Anda harus menggunakan AWS penginstal yang disediakan untuk menginstal perangkat lunak AWS PCS pada AMI kustom Anda. Kami menyarankan Anda menggunakan installer

AWS yang disediakan untuk menginstal Slurm pada AMI kustom Anda tetapi Anda dapat menginstal Slurm sendiri jika Anda mau (tidak disarankan).

Note

Jika Anda ingin mencoba AWS PCS tanpa membuat AMI khusus, Anda dapat menggunakan sampel AMI yang disediakan oleh AWS. Untuk informasi selengkapnya, lihat [Menggunakan sampel Amazon Machine Images \(AMIs\) dengan AWS PCS](#).

Important

AWS PCS saat ini membutuhkan kernel dengan IPv4 dukungan untuk komunikasi node lokal, bahkan ketika Anda menggunakan AWS PCS dalam jaringan IPv6 -only.

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

Topik

- [Langkah 1 - Luncurkan instance sementara](#)
- [Langkah 2 - Instal agen AWS PCS](#)
- [Langkah 3 - Instal Slurm](#)
- [Langkah 4 - \(Opsional\) Instal driver tambahan, perpustakaan, dan perangkat lunak aplikasi](#)
- [Langkah 5 - Buat AMI yang kompatibel dengan AWS PCS](#)
- [Langkah 6 - Gunakan AMI kustom dengan grup node komputasi AWS PCS](#)
- [Langkah 7 - Hentikan instance sementara](#)

Langkah 1 - Luncurkan instance sementara

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

Untuk meluncurkan instans sementara

1. Buka [konsol Amazon EC2](#).

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, seperti PCS-AMI-instance. Nama ditetapkan ke instans sebagai tanda sumber daya (Name=PCS-AMI-instance).
4. Di bagian Application and OS Images, 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 akses SSH masuk ke instans Anda.
8. Di bagian Penyimpanan, konfigurasi 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 agen AWS PCS

Instal agen yang mengonfigurasi instance yang diluncurkan oleh AWS PCS untuk digunakan dengan Slurm. Untuk informasi lebih lanjut tentang agen AWS PCS, lihat [AWS Versi agen PCS](#).

Untuk menginstal agen AWS PCS

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, Amazon Linux 2023, RHEL 9, RHEL 8, Rocky Linux 9, dan Rocky Linux 8

```
sudo yum update -y
```

- Ubuntu 22.04 dan Ubuntu 24.04


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

3. Boot ulang dan terhubung kembali ke instans Anda.

4. Unduh file instalasi agen AWS PCS. File instalasi dikemas ke dalam file tarball () `.tar.gz` terkompresi. Untuk mengunduh versi stabil terbaru, gunakan perintah berikut. Ganti *region* dengan AWS Region tempat Anda meluncurkan instance sementara Anda, seperti `us-east-1`.

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz -o aws-pcs-agent-v1.3.2-1.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 future dari perangkat lunak agen AWS PCS.

5. (Opsional) Verifikasi keaslian dan integritas tarball perangkat lunak AWS PCS. 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 kunci GPG publik untuk AWS PCS dan impor ke keyring Anda. Ganti *region* dengan AWS Region 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.region.amazonaws.com/aws-pcs-public-key.pub && \  
gpg --import aws-pcs-public-key.pub
```

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

```
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 agen AWS PCS jika sidik jari tidak cocok. Hubungi [AWS Support](#).

- c. Unduh file tanda tangan dan verifikasi tanda tangan file tarball perangkat lunak AWS PCS. Ganti *region* dengan AWS Region tempat Anda meluncurkan instance sementara Anda, seperti `us-east-1`.

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz.sig && \  
gpg --verify ./aws-pcs-agent-v1.3.2-1.tar.gz.sig
```

Output harus serupa dengan yang berikut ini:

```
gpg: assuming signed data in './aws-pcs-agent-v1.3.2-1.tar.gz'  
gpg: Signature made Thu 06 Nov 2025 11:10:36 AM CET using RSA key ID ECC0AE5C  
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"  
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: B7E1 8788 3517 6A74 C3D5 EAF5 6088 136D ECC0 AE5C
```

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 perangkat lunak AWS PCS 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-v1.3.2-1.tar.gz && \  
cd aws-pcs-agent
```

7. Instal perangkat lunak AWS PCS.

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

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

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

Output harus serupa dengan yang berikut ini:

```
AGENT_INSTALL_DATE='Fri Dec 13 12:28:43 UTC 2024'  
AGENT_VERSION='1.3.2'  
AGENT_RELEASE='1'
```

Langkah 3 - Instal Slurm

Instal versi Slurm yang kompatibel dengan AWS PCS. Untuk informasi selengkapnya, lihat [Versi slurm di PCS AWS](#).

Note


Jika Anda memiliki AMI dengan versi perangkat lunak Slurm sebelumnya yang diinstal di dalamnya, Anda harus melakukan langkah-langkah berikut untuk menginstal versi baru Slurm. Agen AWS PCS memungkinkan versi binari Slurm yang benar saat runtime, sesuai dengan versi Slurm yang dikonfigurasi pada waktu pembuatan cluster.

Untuk menginstal Slurm

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

```
curl https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz \  
-o aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz
```

Anda juga bisa mendapatkan versi terbaru dengan mengganti nomor versi latest dengan perintah sebelumnya (misalnya: `aws-pcs-slurm-25.05-installer-latest.tar.gz`). Untuk daftar lengkap versi yang tersedia dengan checksum, lihat. [Versi slurm di PCS AWS](#)

 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 kunci GPG publik untuk AWS PCS dan impor ke keyring Anda. Ganti *region* dengan AWS Region 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.region.amazonaws.com/aws-pcs-  
public-key.pub && \  
gpg --import aws-pcs-public-key.pub
```

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

```
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 AWS Region tempat Anda meluncurkan instance sementara Anda, seperti `east-1`.

```
wget https://aws-pcs-repo-region.s3.region.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz.sig && \
  gpg --verify ./aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz.sig
```

Output harus serupa dengan yang berikut ini:

```
gpg: assuming signed data in './aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz'
gpg: Signature made Fri 24 Oct 2025 05:05:11 PM UTC using RSA key ID ECC0AE5C
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"
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: B7E1 8788 3517 6A74 C3D5 EAF5 6088 136D ECC0 AE5C
```

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](#).

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

```
tar -xf aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz && \
  cd aws-pcs-slurm-25.05-installer
```

- 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
```

- Periksa file versi penjadwal untuk mengonfirmasi penginstalan.

```
cat /opt/aws/pcs/scheduler/slurm-25.05/version
```

Output harus serupa dengan yang berikut ini:

```
SLURM_INSTALL_DATE='Mon Nov 3 14:23:38 UTC 2025'
```

```
SLURM_VERSION='25.05.4'  
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 AMI khusus untuk AWS PCS sebelumnya, kami sarankan Anda terlebih dahulu membuat dan menguji AMI hanya dengan perangkat lunak AWS PCS dan Slurm yang diinstal, kemudian secara bertahap menambahkan perangkat lunak dan konfigurasi Anda sendiri setelah Anda mengkonfirmasi keberhasilan awal.

Contoh

- Perangkat lunak Elastic Fabric Adapter (EFA). Untuk informasi selengkapnya, lihat [Memulai EFA dan MPI untuk beban kerja HPC di Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud](#).
- Klien Amazon Elastic File System (Amazon EFS). Untuk informasi selengkapnya, lihat [Menginstal klien Amazon EFS 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 FSx for 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](#).
- NVIDIA Driver, CUDA, dan DCGM, untuk menggunakan tipe instans p* atau g*.

Langkah 5 - Buat AMI yang 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 komputasi AWS PCS.

⚠ Important

AWS PCS saat ini membutuhkan kernel dengan IPv4 dukungan untuk komunikasi node lokal, bahkan ketika Anda menggunakan AWS PCS dalam jaringan IPv6 -only.

Untuk membuat AMI dari instans sementara Anda

1. Buka [konsol Amazon EC2](#).
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 tujuan AMI.
 - c. Pilih Buat gambar.
5. Di panel navigasi, pilih AMIs.
6. Temukan AMI yang Anda buat dalam daftar. Tunggu statusnya berubah dari Pending ke Available, lalu gunakan dengan grup node komputasi AWS PCS.

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

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

⚠ Important

AWS PCS saat ini membutuhkan kernel dengan IPv4 dukungan untuk komunikasi node lokal, bahkan ketika Anda menggunakan AWS PCS dalam jaringan IPv6 -only.

New compute node group

Untuk menggunakan AMI kustom

1. Buka [konsol AWS PCS](#).
2. Pada panel navigasi, silakan pilih Klaster.

3. Pilih cluster tempat Anda akan menggunakan AMI kustom, lalu pilih Compute node groups.
4. Buat grup node komputasi baru. Untuk informasi selengkapnya, lihat [Membuat grup node komputasi di AWS PCS](#). Di bawah ID AMI, cari nama atau ID AMI kustom yang ingin Anda gunakan. Selesai mengkonfigurasi grup node komputasi, lalu pilih Buat grup node komputasi.
5. (Opsional) Konfirmasikan AMI mendukung peluncuran instans. 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 konsol Amazon EC2 hingga muncul instance yang ditandai dengan ID grup node komputasi baru. Untuk informasi lebih lanjut tentang ini, lihat [Menemukan instance grup node komputasi di PCS AWS](#).
 - b. Saat Anda melihat peluncuran instance dan menyelesaikan proses bootstrap, konfirmasikan itu menggunakan AMI yang diharapkan. Untuk melakukan ini, pilih instance, lalu periksa ID AMI di bawah Detail. Ini harus cocok dengan AMI yang Anda konfigurasi dalam pengaturan grup node komputasi.
 - c. (Opsional) Perbarui konfigurasi penskalaan grup node komputasi ke nilai pilihan Anda.

Existing compute node group

Untuk menggunakan AMI kustom

1. Buka [konsol AWS PCS](#).
2. Pada panel navigasi, silakan pilih Klaster.
3. Pilih cluster tempat Anda akan menggunakan AMI kustom, lalu pilih Compute node groups.
4. Pilih grup simpul yang ingin Anda konfigurasi dan pilih Edit. Di bawah ID AMI, cari nama atau ID AMI kustom yang ingin Anda gunakan. Selesai mengkonfigurasi grup node komputasi, lalu pilih Perbarui. Instance baru yang diluncurkan di grup node komputasi akan menggunakan ID AMI yang diperbarui. Instans yang ada akan terus menggunakan AMI lama sampai AWS PCS menggantikannya. Untuk informasi selengkapnya, lihat [Memperbarui grup node komputasi AWS PCS](#).
5. (Opsional) Konfirmasikan AMI mendukung peluncuran instans. 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 konsol Amazon EC2 hingga muncul instance yang ditandai dengan ID grup node komputasi baru. Untuk informasi lebih lanjut tentang ini, lihat [Menemukan instance grup node komputasi di PCS AWS](#)..
- b. Saat Anda melihat peluncuran instance dan menyelesaikan proses bootstrap, konfirmasi itu menggunakan AMI yang diharapkan. Untuk melakukan ini, pilih instance, lalu periksa ID AMI di bawah Detail. Ini harus cocok dengan AMI yang Anda konfigurasi 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 berfungsi sebagaimana dimaksud dengan AWS PCS, Anda dapat menghentikan instans sementara untuk menghentikan biaya untuk itu.

Untuk mengakhiri instans sementara

1. Buka [konsol Amazon EC2](#).
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 perangkat lunak AWS PCS 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 membangun kustom AMIs untuk digunakan dengan AWS PCS atau Anda dapat menggunakan metode Anda sendiri.

Daftar Isi

- [AWS Penginstal perangkat lunak agen PCS](#)
- [Pemasang slurm](#)
- [Sistem operasi yang didukung](#)

- [Tipe instans yang didukung](#)
- [Versi Slurm yang didukung](#)
- [Verifikasi penginstal menggunakan checksum](#)

AWS Penginstal perangkat lunak agen PCS

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

Untuk informasi lebih lanjut tentang perangkat lunak agen AWS PCS, lihat [AWS Versi agen PCS](#).

Pemasang slurm

Penginstal Slurm mengunduh, mengkompilasi, dan menginstal versi Slurm yang relevan dan dependensinya. Anda dapat menggunakan installer Slurm untuk membangun kustom AMIs untuk PCS. AWS Anda juga dapat menggunakan mekanisme Anda sendiri jika mereka konsisten dengan konfigurasi perangkat lunak yang disediakan oleh penginstal Slurm. Untuk informasi selengkapnya tentang dukungan AWS PCS untuk Slurm, lihat [Versi slurm di PCS AWS](#)

Perangkat lunak AWS yang disediakan menginstal yang berikut ini:

- [Slurm pada versi mayor dan pemeliharaan yang diminta \(saat ini versi 25.05.x\) - Lisensi GPL 2](#)
 - 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 dukungan PMIX dan JWT
 - Slurm dipasang di `/opt/aws/pcs/schedulers/slurm-25.05`
- [OpenPMix \(versi 4.2.6\) - Lisensi](#)
 - OpenPMix diinstal sebagai subdirektori `/opt/aws/pcs/scheduler/`
- [libjwt \(versi 1.17.0\) - Lisensi MPL-2.0](#)
 - libjwt diinstal sebagai subdirektori dari `/opt/aws/pcs/scheduler/`

Perangkat lunak AWS yang disediakan mengubah konfigurasi sistem sebagai berikut:

- systemdFile Slurm yang dibuat oleh build disalin `/etc/systemd/system/` dengan nama file. `slurmd-25.05.service`
- Jika tidak ada, pengguna Slurm dan grup (`slurm:slurm`) dibuat dengan UID/GID of. 401
- Folder `/etc/aws/pcs/scheduler/slurm-25.05/pluginstack.conf.d/` dibuat, untuk menyimpan [Perluas fungsionalitas Slurm pada AWS PCS dengan plugin SPANK](#) konfigurasi Anda.
- Di Amazon Linux 2 dan Rocky Linux 9 instalasi menambahkan repositori EPEL 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

Lihat [Sistem operasi yang didukung di AWS PCS](#).

Note

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

Tipe instans yang didukung

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

Versi Slurm yang didukung

Lihat [Versi slurm di PCS AWS](#).

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 [Dukungan](#).

Misalnya, perintah berikut menghasilkan SHA256 checksum untuk tarball Slurm 25.05.4-1.

```
$ sha256sum aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz
```

Contoh output:

```
3b0f93bce441d4f4f6935175f2c1e81cd961cb923adb416fa6689f5592047a7d aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz
```

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

AWS Agen PCS

Penginstal	Unduh URL	SHA256 checksum
AWS Agen PCS 1.3.2-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-</i></code>	<code>06b32a952a1c849e3442e35c28ac2e4d6962</code>

Penginstal	Unduh URL	SHA256 checksum
	<code>east-1 .amazonaws.com/ aws-pcs-agent/aws-pcs- agent-v1.3.2-1.tar.gz</code>	<code>b09286cad748f3c83d 561b52ec6f</code>
AWS Agen PCS 1.3.1-1	<code>https://aws-pcs-repo- us-east-1.s3.us-east-1. amazonaws.com/ aws-pcs-agent/aws-pcs- agent-v1.3.1-1.tar.gz</code>	<code>5b7f1eb7b3a86bd2d3 31b5cb0138d868dc94 52da34b480becd86af 892c7e8d19</code>
AWS Agen PCS 1.3.0-1	<code>https://aws-pcs-repo- us-east-1.s3.us-east-1. amazonaws.com/ aws-pcs-agent/aws-pcs- agent-v1.3.0-1.tar.gz</code>	<code>eadc9b65c3db248bdd e2a6c41814dfb1b972 39f24ad55e03d8526d d9ab4a8d16</code>
AWS Agen PCS 1.2.2-1	<code>https://aws-pcs-repo- us-east-1.s3.us-east-1. amazonaws.com/ aws-pcs-agent/aws-pcs- agent-v1.2.2-1.tar.gz</code>	<code>fd7b6ea5442db75d72 3fc4971781ce6ae511 baa21b87c4286fc1df 8127b282b8</code>
AWS Agen PCS 1.2.1-1	<code>https://aws-pcs-repo- us-east-1.s3.us-east-1. amazonaws.com/ aws-pcs-agent/aws-pcs- agent-v1.2.1-1.tar.gz</code>	<code>2b784643ca01ccca1b aa64fbfb34bb41efe8 bdca69470998b74ce3 962bc271d4</code>

Penginstal	Unduh URL	SHA256 checksum
AWS Agen PCS 1.2.0-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.0-1.tar.gz</code>	<code>470db8c4fc9e50277b6317f98584b6b547e73523043e34f018eeca e767846805</code>
AWS Agen PCS 1.1.1-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz</code>	<code>bef078bf60a6d8ecde2e6c49cd34d088703f02550279e3bf483d57 a235334dc6</code>
AWS Agen PCS 1.1.0-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.0-1.tar.gz</code>	<code>594c32194c71bccc5d66e5213213ae38dd2c6d2f9a950bb01accea 0bbab0873a</code>
AWS Agen PCS 1.0.1-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.1-1.tar.gz</code>	<code>04e22264019837e3f42d8346daf5886eaaced21571742eb505ea8 911786bcb2</code>
AWS Agen PCS 1.0.0-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz</code>	<code>d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042 144b134ce0</code>

Pemasang slurm

Penginstal	Unduh URL	SHA256 checksum
Slurm 25.05.4-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz</code>	3b0f93bce441d4f4f6 935175f2c1e81cd961 cb923adb416fa6689f 5592047a7d
Slurm 25.05.3-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.3-1.tar.gz</code>	851bb5815b6700ceb3 0cc4a3fda204ca8ce3 62c14528c339908983 255a936cf0
Slurm 24.11.6-2	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.6-2.tar.gz</code>	f17cd78e0bc6b9c818 b794d9d2685cceabdc 73f4fbb12f7566ae5b 86a5abc32b
Slurm 24.11.6-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.6-1.tar.gz</code>	225de9fc18206f5f65 f412effe1fd457614a c97ee9822b3ff804a4 52b0fae522
Slurm 24.11.5-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz</code>	593efe4d66bef2f3e4 6d5a382fb5a32f7a3c a2510bcf1b3c85739f 4f951810d5

Penginstal	Unduh URL	SHA256 checksum
Slurm 24.05.8-2	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-2.tar.gz</code>	<code>c494b0b55c319a4c2f3faf668c759d46c32c4c7aa94ae97d94128328fe95364b</code>
Slurm 24.05.8-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-1.tar.gz</code>	<code>210a43b376af082bbad640b2032655885790c5dab0e6489cc327c7310a375849</code>
Slurm 24.05.7-1	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.7-1.tar.gz</code>	<code>0b5ed7c81195de2628c78f37c79e63fc4ae99132ca6b019b53a0d68792ee82c5</code>
Slurm 24.05.5-2	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz</code>	<code>7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b</code>
Slurm 23.11.10-4 (usang)	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-4.tar.gz</code>	<code>bb2d8c919c69dba38d14358f49c7f0427564c5dd4af85a1c9eca2c57ceeae29a</code>

Penginstal	Unduh URL	SHA256 checksum
Slurm 23.11.10-3 (usang)	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-3.tar.gz</code>	488a10ee0fbd57ec0e0ff7ea708a9e3038fafdc025c6bb391c75c2e2a7852a00
Slurm 23.11.10-2 (usang)	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-2.tar.gz</code>	0bbe85423305c05987931168caf98da08a34c25f9eec0690e8e74de0b7bc8752
Slurm 23.11.10-1 (usang)	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-1.tar.gz</code>	27e8faa9980e92cdfd8cfdc71f937777f0934552ce61e33dac4ecf5a20321e44
Slurm 23.11.9-1 (usang)	<code>https://aws-pcs-repo-<i>us-east-1</i>.s3.<i>us-east-1</i>.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz</code>	1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8

Catatan rilis untuk sampel AWS PCS AMIs

AMIs untuk versi utama terbaru yang didukung dari penjadwal menerima pembaruan keamanan dan perbaikan bug kritis. Patch keamanan tambahan ini tidak termasuk dalam catatan rilis resmi.

⚠ Important

Contoh AMIs yang terkait dengan versi penjadwal lama tidak didukung dan tidak menerima pembaruan.

⚠ Important

AMIs Sampel untuk tujuan demonstrasi dan tidak direkomendasikan untuk beban kerja produksi.

Daftar Isi

- [AWS Sampel PCS AMIs untuk x86_64 \(Amazon Linux 2\)](#)
- [AWS Sampel PCS AMIs untuk Arm64 \(Amazon Linux 2\)](#)

AWS Sampel PCS AMIs untuk x86_64 (Amazon Linux 2)

Slurm 25,05

Nama AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05`

Instans EC2 yang didukung

- Semua instance dengan prosesor x86 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol Amazon EC2. Pilih Jenis Instance, lalu cari Architectures=x86_64.

Konten AMI

- Layanan AWS yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: x86_64
- Jenis volume EBS: gp2
- Pemasang EFA: 1.43.1

- GDRCopy: 2.5.1
- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 24.11

Note

AWS PCS mendukung akuntansi untuk Slurm 24.11 dan yang lebih baru. Untuk informasi selengkapnya, lihat [Akuntansi slurm di PCS AWS](#).

Nama AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11`

Instans EC2 yang didukung

- Semua instance dengan prosesor x86 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol [Amazon EC2](#). Pilih Jenis Instance, lalu cari `Architectures=x86_64`.

Konten AMI

- AWS Layanan yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: `x86_64`
- Jenis volume EBS: `gp2`
- Pemasang EFA: 1.33.0
- GDRCopy: 2.4
- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 24,05

Nama AMI

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.05`

Instans EC2 yang didukung

- Semua instance dengan prosesor x86 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol [Amazon EC2](#). Pilih Jenis Instance, lalu cari `Architectures=x86_64`.

Konten AMI

- AWS Layanan yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: `x86_64`
- Jenis volume EBS: `gp2`
- Pemasang EFA: `1.33.0`
- GDRCopy: `2.4`
- Pengemudi NVIDIA: `550.127.08`
- NVIDIA CUDA: `12.4.1_550.54.15`

Slurm 23.11

Nama AMI

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

Instans EC2 yang didukung

- Semua instance dengan prosesor x86 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol [Amazon EC2](#). Pilih Jenis Instance, lalu cari `Architectures=x86_64`.

Konten AMI

- AWS Layanan yang Didukung: AWS PCS

- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: x86_64
- Jenis volume EBS: gp2
- Pemasang EFA: 1.33.0
- GDRCopy: 2.4
- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

AWS Sampel PCS AMIs untuk Arm64 (Amazon Linux 2)

Slurm 25,05

Nama AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-25.05`

Instans EC2 yang didukung

- Semua instance dengan prosesor Arm 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol Amazon EC2. Pilih Jenis Instance, lalu cari Architectures=ARM64.

Konten AMI

- Layanan AWS yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: arm64
- Jenis volume EBS: gp2
- Pemasang EFA: 1.43.1
- GDRCopy: 2.5.1
- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 24.11

Note

AWS PCS mendukung akuntansi untuk Slurm 24.11 dan yang lebih baru. Untuk informasi selengkapnya, lihat [Akuntansi slurm di PCS AWS](#).

Nama AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.11`

Instans EC2 yang didukung

- Semua instance dengan prosesor Arm 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol [Amazon EC2](#). Pilih Jenis Instance, lalu cari `Architectures=arm64`.

Konten AMI

- AWS Layanan yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: arm64
- Jenis volume EBS: gp2
- Pemasang EFA: 1.33.0
- GDRCopy: 2.4
- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 24,05

Nama AMI

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.05`

Instans EC2 yang didukung

- Semua instance dengan prosesor Arm 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol [Amazon EC2](#). Pilih Jenis Instance, lalu cari `Architectures=arm64`.

Konten AMI

- AWS Layanan yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: arm64
- Jenis volume EBS: gp2
- Pemasang EFA: 1.33.0
- GDRCopy: 2.4
- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 23.11

Nama AMI

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

Instans EC2 yang didukung

- Semua instance dengan prosesor Arm 64-bit. Untuk menemukan instans yang kompatibel, navigasikan ke konsol [Amazon EC2](#). Pilih Jenis Instance, lalu cari `Architectures=arm64`.

Konten AMI

- AWS Layanan yang Didukung: AWS PCS
- Sistem Operasi: Amazon Linux 2
- Arsitektur Komputasi: arm64
- Jenis volume EBS: gp2
- Pemasang EFA: 1.33.0
- GDRCopy: 2.4

- Pengemudi NVIDIA: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Sistem operasi yang didukung di AWS PCS

AWS PCS menggunakan Amazon Machine Image (AMI) yang dikonfigurasi untuk grup node komputasi untuk meluncurkan EC2 instance dalam grup node komputasi tersebut. AMI menentukan sistem operasi yang digunakan EC2 instance. Anda tidak dapat mengubah sistem operasi dalam sampel AWS PCS AMIs. Anda harus membuat AMI khusus jika Anda ingin menggunakan sistem operasi yang berbeda. Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon \(AMIs\) untuk AWS PCS](#).

Sistem operasi yang didukung

- Amazon Linux 2

Ini adalah sistem operasi dalam sampel AWS PCS AMIs.

Important

Sampel AMIs adalah untuk tujuan demonstrasi dan tidak direkomendasikan untuk beban kerja produksi. Anda harus membuat dan menggunakan AMI khusus untuk beban kerja produksi, bahkan jika Anda berniat menggunakan Amazon Linux 2.

- Amazon Linux 2023
- RedHat Perusahaan Linux 9 (RHEL 9)

Biaya sesuai permintaan untuk RHEL jenis instans apa pun lebih tinggi daripada sistem operasi lain yang didukung. Untuk informasi selengkapnya tentang harga, lihat Harga [Sesuai Permintaan dan Bagaimana Red Hat Enterprise Linux di Amazon Elastic Compute Cloud ditawarkan dan diberi harga?](#) .

- RedHat Perusahaan Linux 8 (RHEL 8)
- Berbatu Linux 9

Anda dapat menggunakan [Rocky Linux 9 resmi AMIs](#) sebagai basis untuk AMI khusus. Build AMI kustom Anda mungkin gagal jika AMI dasar tidak memiliki kernel terbaru.

Untuk meng-upgrade kernel

1. [Luncurkan instance menggunakan id AMI rocky9 dari sini: https://rockylinux.org/cloud-images/](https://rockylinux.org/cloud-images/)
2. ssh ke dalam instance dan jalankan perintah berikut:

```
sudo yum -y update
```

3. Buat gambar dari instance. Anda menentukan gambar ini sebagai ParentImage untuk AMI kustom Anda.

- Berbatu Linux 8
- Ubuntu 22.04

Ubuntu 22.04 membutuhkan kunci yang lebih aman untuk SSH dan tidak mendukung kunci RSA secara default. Kami menyarankan Anda membuat dan menggunakan ED25519 kunci sebagai gantinya.

- Ubuntu 24.04

AWS Versi agen PCS

Perangkat lunak agen AWS PCS mengonfigurasi instans EC2 yang diluncurkan AWS PCS untuk digunakan dengan Slurm. Anda menyertakan agen dalam Amazon Machine Images (AMI) yang Anda tentukan saat membuat grup node komputasi untuk kluster Anda. Instans EC2 yang diluncurkan di grup node komputasi tersebut menggunakan AMI yang ditentukan dan perangkat lunak agen AWS PCS yang disertakan. Agen AWS PCS memungkinkan instans EC2 untuk mendaftarkan dirinya sebagai bagian dari cluster. Untuk menggunakan perangkat lunak agen AWS PCS terbaru, Anda harus memperbarui kustom Anda AMIs. Untuk informasi selengkapnya, lihat [Langkah 2 - Instal agen AWS PCS](#) di [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

AWS Versi agen PCS	Tanggal rilis	Catatan rilis
v1.3.2-1	10 Maret 2026	<ul style="list-style-type: none"> Memperbaiki masalah di mana node komputasi yang menjalankan RHEL 8.10 atau Rocky Linux 8.10 gagal melakukan bootstrap karena backport curl SiGv4 yang salah di sistem operasi tersebut.
v1.3.1-1	November 7, 2025	<ul style="list-style-type: none"> Peningkatan penonaktifan hyperthreading dengan menggunakan parameter sysfs `smt/control` bila tersedia. Memperbaiki kondisi balapan potensial saat CPU terkunci saat boot saat Agen PCS mencoba menonaktifkan hyperthreading. Memperbaiki masalah yang menyebabkan InstanceId dan InstanceType bidang node komputasi

AWS Versi agen PCS	Tanggal rilis	Catatan rilis
		Slurm diisi dengan stempel waktu dan tanda hubung, masing-masing.
v1.3.0-1	November 3, 2025	<ul style="list-style-type: none">• Menambahkan dukungan untuk sistem operasi baru: Amazon Linux 2023, Ubuntu 24, RHEL 8, Rocky 8.
v1.2.2-1	Oktober 16, 2025	<ul style="list-style-type: none">• Kueri metadata instance yang diizinkan ke IPv6 titik akhir jika titik akhir tidak IPv4 tersedia.• Memperbaiki masalah yang mencegah hyperthreading dinonaktifkan jika kernel mengembalikan thread saudara sebagai rentang ID CPU.• Memperbaiki masalah yang menghasilkan pesan kegagalan palsu di log saat hyperthreading berhasil dinonaktifkan.

AWS Versi agen PCS	Tanggal rilis	Catatan rilis
v1.2.1-1	Juni 19, 2025	<ul style="list-style-type: none"> • Agen AWS PCS sekarang mencoba memulai slurmd hingga 30 menit jika pengontrol tidak tersedia. • Memperbaiki masalah yang menghasilkan konfigurasi slurmd yang salah jika respons RegisterComputeNodeGroupInstance berisi titik akhir SLURMDBD.
v1.2.0-1	7 Maret 2025	<ul style="list-style-type: none"> • Dukungan yang diaktifkan untuk IPv6 <code>inslurmd.conf</code>.
v1.1.1-1	Desember 13, 2024	<ul style="list-style-type: none"> • Memperbaiki masalah saat versi Slurm yang salah dilaporkan dalam panggilan ke <code>RegisterComputeNodeGroupInstance</code> • Memperbaiki masalah saat metadata instance tidak diambil dengan benar jika skrip kustom dieksekusi. <code>/opt/aws/pcs/etc/bootstrap_hooks/</code>
v1.1.0-1	Desember 6, 2024	<ul style="list-style-type: none"> • Aktifkan skrip kustom <code>/opt/aws/pcs/etc/bootstrap_hooks/</code> untuk dijalankan sebelum langkah bootstrap.

AWS Versi agen PCS	Tanggal rilis	Catatan rilis
v1.0.1-1	Oktober 22, 2024	<ul style="list-style-type: none">• Memperbaiki masalah saat perangkat NVIDIA tidak berfungsi saat <code>slurmd</code> dimulai pada instans berkemampuan GPU.
v1.0.0-1	Agustus 28, 2024	<ul style="list-style-type: none">• Pelepasan awal.

Penjadwal slurm di PCS AWS

Slurm adalah manajer beban kerja sumber terbuka yang dirancang untuk cluster Linux yang menyediakan penjadwalan pekerjaan, alokasi sumber daya, dan kemampuan pemantauan pekerjaan untuk beban kerja HPC. AWS PCS mendukung penjadwal Slurm untuk mengelola beban kerja cluster Anda.

Topik

- [Versi slurm di PCS AWS](#)
- [Akuntansi slurm di PCS AWS](#)
- [Slurm REST API di PCS AWS](#)
- [Mem-boot ulang node komputasi dengan Slurm di PCS AWS](#)
- [Mengkonfigurasi pengaturan Slurm khusus di PCS AWS](#)
- [Perluas fungsionalitas Slurm pada AWS PCS dengan plugin SPANK](#)
- [Gunakan Plugin Filter CLI Slurm untuk menyesuaikan pengiriman pekerjaan di PCS AWS](#)

Versi slurm di PCS AWS

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 PCS dirancang untuk memperbarui pengontrol Slurm secara otomatis dengan versi patch.

Ketika SchedMD mengakhiri [dukungan](#) untuk versi utama tertentu, AWS PCS menetapkan versi itu sebagai End of Life (EOL). Setelah EOL, tidak ada cluster baru yang dapat dibuat dengan versi itu, meskipun cluster yang ada dapat terus berjalan hingga 12 bulan tanpa dukungan yang dijamin. AWS PCS mengirimkan pemberitahuan terlebih dahulu jika versi utama Slurm dekat dengan EOL, 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 kluster Anda, untuk mengakses kemajuan dan peningkatan terbaru.

Versi Slurm yang didukung di PCS AWS

Tabel berikut menunjukkan versi Slurm yang didukung dan tanggal serta informasi penting untuk setiap versi.

Versi slurm	Tanggal rilis SchedMD	AWS Tanggal rilis PCS	AWS Tanggal PCS EOL	Versi agen AWS PCS minimum yang kompatibel	Sampel AWS PCS yang didukung AMIs
25.05	5/29/2025	10/16/2025	5/31/2027	1.0.0-1	<ul style="list-style-type: none"> aws-pcs-s ample_ami -amzn2-x86_64-slurm-25.05 aws-pcs-s ample_ami -amzn2-arm64-slurm-25.05
24.11	11/29/2024	5/14/2025	5/31/2026	1.0.0-1	<ul style="list-style-type: none"> aws-pcs-s ample_ami -amzn2-x86_64-slurm-24.11 aws-pcs-s ample_ami

Versi slurm	Tanggal rilis SchedMD	AWS Tanggal rilis PCS	AWS Tanggal PCS EOL	Versi agen AWS PCS minimum yang kompatibel	Sampel AWS PCS yang didukung AMIs
					-amzn2-arm64-slurm-24.11

Versi Slurm yang tidak didukung di PCS AWS

Tabel berikut menunjukkan versi Slurm yang tidak didukung di AWS PCS.

Versi slurm	Tanggal rilis SchedMD	AWS Tanggal rilis PCS	AWS Tanggal PCS EOL		
24.05	5/30/2024	12/18/2024	11/30/2025		
23.11	11/21/2023	8/28/2024	5/31/2025		

Catatan rilis untuk versi Slurm di PCS AWS

Topik ini menjelaskan perubahan penting untuk setiap versi Slurm yang saat ini didukung di AWS PCS. Kami sarankan Anda meninjau perubahan antara versi lama dan baru saat Anda mengupgrade kluster Anda.

Slurm 25,05

Perubahan diterapkan di AWS PCS

- Slurm SchedulerParameter `requeue_on_resume_failure` sekarang Diaktifkan secara default.
- “`stderr`” telah dihapus sebagai opsi untuk `LogTimeFormat`, karena dinonaktifkan di Slurm 25.05.
- AWS PCS mendukung konfigurasi `sackd` multi-cluster: node login dapat mengakses beberapa cluster.

Untuk informasi lebih lanjut tentang Slurm 25.05, lihat publikasi berikut:

- Pengumuman rilis SchedMD: <https://www.schedmd.com/slurm-version-25-05-0-is-now-available/>
- Catatan rilis SchedMD: https://github.com/SchedMD/slurm/blob/slurm-25-05-0-1/RELEASE_notes.md

Slurm 24,11

Perubahan diterapkan di AWS PCS

- AWS PCS mendukung akuntansi Slurm. Untuk informasi selengkapnya, lihat [Akuntansi slurm di PCS AWS](#).

Untuk informasi lebih lanjut tentang Slurm 24.11, lihat publikasi berikut:

- [Pengumuman rilis SchedMD](#)
- [Catatan rilis SchedMD](#)

Slurm 24,05

Perubahan diterapkan di AWS PCS

- Modul Slurm Step Manager baru sekarang diaktifkan secara default di AWS PCS. Modul ini memberikan manfaat yang signifikan dengan membongkar manajemen langkah dari pengontrol pusat ke node komputasi, secara substansional meningkatkan konkurensi sistem di lingkungan dengan penggunaan langkah berat. Untuk mendukung konfigurasi ini dan mengisolasi Prolog dan Epilog memproses eksekusi yang lebih baik, flag prolog baru (Contain,Alloc) diaktifkan.
- Komunikasi hierarkis dari pengontrol ke node komputasi diaktifkan untuk mengoptimalkan komunikasi intra-node Slurm, yang meningkatkan skalabilitas dan kinerja. Selain itu, konfigurasi routing sekarang menggunakan daftar node partisi untuk komunikasi dari controller, bukan algoritma routing default plugin, meningkatkan ketahanan sistem.
- Plugin hash baru HashPlugin=hash/sha3 menggantikan yang sebelumnya. hash/k12 plugin ini sekarang diaktifkan secara default di cluster AWS PCS.
- Log pengontrol slurm sekarang menyertakan kemampuan audit yang ditingkatkan untuk semua panggilan prosedur jarak jauh masuk (RPC) ke. slurmctld Log termasuk alamat sumber, pengguna yang diautentikasi, dan jenis RPC sebelum pemrosesan koneksi.

Untuk informasi lebih lanjut tentang Slurm 24.05, lihat publikasi berikut:

- [Pengumuman rilis SchedMD](#)
- [Catatan rilis SchedMD](#)

Slurm 23.11

Pengaturan slurm Anda dapat mengubah PCS AWS

- SuspendTimeDefaultnya ke. 60 Gunakan parameter `scaleDownIdleTimeInSeconds` konfigurasi AWS PCS untuk mengaturnya. Untuk informasi selengkapnya, lihat [scaleDownIdleTimeInSeconds](#) parameter tipe `ClusterSlurmConfiguration` data di Referensi AWS PCS API.
- MaxJobCountDan MaxArraySize didasarkan pada ukuran yang Anda pilih untuk cluster. Untuk informasi selengkapnya, lihat [size](#) parameter aksi `CreateCluster` API di Referensi AWS PCS API.
- Pengaturan `SelectTypeParameters` Slurm default ke. `CR_CPU` Anda dapat memberikannya sebagai nilai untuk `slurmCustomSettings` mengaturnya saat Anda membuat cluster. Untuk informasi selengkapnya, lihat [slurmCustomSettings](#) parameter aksi `CreateCluster` API dan [SlurmCustomSetting](#) Referensi AWS PCS API.
- Anda dapat mengatur `Prolog` dan `Epilog` pada tingkat cluster. Anda dapat memberikannya sebagai nilai untuk `slurmCustomSettings` mengaturnya saat Anda membuat cluster. Untuk informasi selengkapnya, lihat [CreateCluster](#) dan [SlurmCustomSetting](#) di Referensi AWS PCS API.
- Anda dapat mengatur `Weight` dan `RealMemory` pada tingkat grup node komputasi. Anda dapat memberikannya sebagai nilai `slurmCustomSettings` untuk mengaturnya saat Anda membuat grup node komputasi. Untuk informasi selengkapnya, lihat [CreateComputeNodeGroup](#) dan [SlurmCustomSetting](#) di Referensi AWS PCS API.

Pertanyaan yang sering diajukan tentang versi Slurm di PCS AWS

AWS PCS mempertahankan dukungan untuk beberapa versi Slurm. Ketika versi Slurm baru diperkenalkan, AWS PCS menyediakan dukungan teknis dan patch keamanan hingga versi tersebut mencapai akhir dukungan (EOS) dari SchedMD. AWS PCS mengacu pada tanggal EOS untuk versi Slurm sebagai akhir masa pakai (EOL) agar konsisten dengan terminologi AWS

Berapa lama AWS PCS mendukung versi Slurm?

AWS Dukungan PCS untuk versi Slurm selaras dengan siklus dukungan SchedMD untuk versi utama. AWS PCS mendukung versi saat ini dan 2 versi utama terbaru sebelumnya. Ketika SchedMD merilis versi utama baru, AWS PCS mengakhiri dukungan untuk versi tertua yang didukung. AWS PCS merilis versi utama baru Slurm sesegera mungkin tetapi mungkin ada penundaan antara rilis SchedMD dan ketersediaannya di PCS. AWS

Bagaimana cara cluster saya mendapatkan rilis versi patch Slurm baru?

Untuk mengatasi bug dan perbaikan keamanan, AWS PCS dirancang untuk secara otomatis menerapkan tambalan ke pengontrol cluster yang berjalan di akun milik layanan internal. Untuk menginstal patch pada instans EC2 di instans Anda Akun AWS, perbarui Amazon Machine Image (AMI) untuk grup node komputasi Anda dan perbarui grup node komputasi untuk menggunakan AMI yang diperbarui. Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Note

Pengontrol slurm tidak tersedia saat kami memperbaruinya. Menjalankan pekerjaan tidak terpengaruh. Pekerjaan yang dikirimkan sebelum controller cluster menjadi tidak tersedia ditahan sampai controller tersedia.

Bagaimana saya mendapat informasi tentang acara EOL versi Slurm yang akan datang?

Kami mengirimkan pesan email 6 bulan sebelum tanggal EOL. Kami mengirimi Anda pesan email setiap bulan sebelum EOL, dengan pesan email terakhir 1 minggu sebelum tanggal EOL. Setelah tanggal EOL, kami mengirim pesan email bulanan selama 12 bulan kepada pelanggan yang menjalankan cluster AWS PCS dengan versi EOL Slurm. Kami mungkin menangguhkan cluster dengan versi EOL Slurm jika kerentanan keamanan diidentifikasi untuk versi itu.

Bagaimana saya bisa menentukan apakah versi Slurm yang digunakan oleh cluster saya menjalankan versi EOL Slurm?

Kami mengirimi Anda pesan email untuk memberi tahu Anda bahwa Anda memiliki cluster yang berjalan dengan versi EOL Slurm. Kami memposting peringatan untuk Dasbor AWS Health peringatan yang berisi rincian cluster Anda dengan versi EOL Slurm. Anda juga dapat menggunakan konsol AWS PCS untuk mengidentifikasi cluster dengan versi EOL Slurm.

Apa yang harus saya lakukan jika versi Slurm saya dekat atau di luar EOL?

Buat cluster baru dengan versi Slurm yang didukung lebih baru dan perbarui versi Slurm di AMI grup node komputasi Anda. Versi Slurm di AMI Anda dan menjalankan instans EC2 tidak boleh lebih dari 2 versi di belakang versi Slurm cluster. Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Apa yang akan terjadi jika saya tidak beralih ke versi Slurm yang lebih baru pada tanggal EOL?

Anda tidak dapat membuat cluster baru dengan versi EOL Slurm. Cluster yang ada dapat beroperasi hingga 12 bulan tanpa AWS dukungan, dan tidak ada tindakan segera yang diperlukan untuk mempertahankan operasinya. Setelah tanggal EOL, dukungan, pembaruan keamanan, dan ketersediaan tidak dijamin. Kami mungkin menanggihkan cluster untuk alasan keamanan. Kami sangat menyarankan Anda menggunakan versi Slurm yang didukung untuk menjaga keamanan dan dukungan untuk klaster AWS PCS Anda.

Apa risiko mengoperasikan cluster dengan versi EOL Slurm?

Cluster dengan versi EOL Slurm menghadirkan risiko keamanan dan operasional yang signifikan. Tanpa pemantauan aktif SchedMD, kerentanan keamanan mungkin tetap tidak terdeteksi atau tidak ditangani. Jika kerentanan kritis ditemukan, kami mungkin segera menanggihkan klaster Anda.

Apa yang terjadi pada pekerjaan, komputasi klaster, penyimpanan, dan sumber daya jaringan saya ketika klaster saya ditanggihkan?

Semua sumber daya yang dikelola oleh AWS PCS dihentikan. Ini termasuk pengontrol Slurm, grup node komputasi, dan instans EC2. Setiap pekerjaan yang berjalan pada instance komputasi segera dihentikan, dan klaster memasuki status ditanggihkan. Sumber daya yang dikelola pelanggan, seperti sistem file eksternal, tetap utuh. Anda dapat menggunakan konsol AWS PCS dan tindakan API untuk mengakses konfigurasi cluster.

Bisakah saya memulai ulang cluster yang ditanggihkan untuk melanjutkan pekerjaannya yang tersisa?

Tidak, Anda tidak dapat memulai ulang cluster yang ditanggihkan. Anda dapat menggunakan konfigurasi cluster yang ditanggihkan untuk membuat klaster baru dengan versi Slurm yang didukung. Anda dapat menjalankan pekerjaan yang tersisa jika Anda menyimpannya di sistem file eksternal.

Dapatkah saya meminta perpanjangan di luar masa tenggang 12 bulan?

Tidak, Anda tidak dapat meminta ekstensi untuk menjalankan kluster Anda di luar masa tenggang 12 bulan. Kami menyediakan waktu yang diperpanjang untuk membantu Anda beralih ke versi Slurm yang didukung. Untuk menghindari gangguan pada operasi kluster Anda, kami sarankan Anda beralih sebelum versi Slurm Anda mencapai EOL.

Akuntansi slurm di PCS AWS

Anda dapat mengaktifkan akuntansi pada kluster AWS PCS baru Anda untuk memantau penggunaan kluster, menerapkan batas sumber daya, dan mengelola kontrol akses berbutir halus ke antrian tertentu atau grup node komputasi. AWS PCS membuat dan mengelola database akuntansi untuk cluster Anda, menghilangkan kebutuhan bagi Anda untuk membuat dan mengelola database akuntansi terpisah Anda sendiri. AWS PCS menggunakan fitur akuntansi di Slurm. Untuk informasi selengkapnya tentang fitur akuntansi di Slurm, lihat dokumentasi [Slurm di SchedMD](#).

Untuk menggunakan akuntansi, aktifkan saat Anda membuat cluster baru dan secara opsional mengatur parameter akuntansi. Setelah status cluster Anda `Active` dan memiliki grup node komputasi, Anda dapat terhubung ke shell Linux dari node login untuk melakukan fungsi akuntansi, seperti melihat data pekerjaan dengan perintah `Slurmsacct`.

Note

Akuntansi didukung untuk Slurm 24.11 atau yang lebih baru.

AWS PCS console

Pada halaman `Create cluster`, Anda harus memilih versi Slurm yang valid (versi 24.11 atau yang lebih baru). Di bawah pengaturan `Scheduler`, aktifkan `Akuntansi`.

AWS PCS API

Berikan `accounting` konfigurasi dalam panggilan Anda ke tindakan `CreateCluster` API. Dalam `accounting` objek, atur mode ke `STANDARD`. Untuk informasi selengkapnya, lihat [CreateCluster](#) dan [Akuntansi](#) di Referensi AWS PCS API.

Contoh berikut menggunakan AWS CLI untuk memanggil tindakan `CreateCluster` API. Substring nilai parameter `accounting='{mode=STANDARD}'` memungkinkan akuntansi.

```
aws pcs create-cluster --cluster-name cluster-name \
```

```
--scheduler type=SLURM,version=24.11 \  
--size SMALL \  
--networking subnetIds=cluster-subnet-  
id,securityGroupIds=cluster-security-group-id \  
--slurm-configuration  
scaleDownIdleTimeInSeconds=180,accounting='{mode=STANDARD}',slurmCustomSettings='[{parameter
```

Important

Anda mendapatkan biaya penagihan tambahan jika Anda mengaktifkan akuntansi. Untuk informasi selengkapnya, lihat [halaman harga AWS PCS](#).

Memodifikasi pengaturan akuntansi

Anda dapat mengaktifkan atau menonaktifkan akuntansi pada cluster yang ada tanpa membangun kembali infrastruktur Anda. Untuk informasi selengkapnya, lihat [Memperbarui cluster di AWS PCS](#).

Saat Anda menonaktifkan akuntansi, penagihan untuk fitur akuntansi berhenti segera setelah cluster memasuki UPDATING status. Saat Anda mengaktifkan akuntansi, penagihan dimulai ketika cluster berhasil kembali ke ACTIVE status.

Konsep kunci untuk akuntansi Slurm di PCS AWS

Konsep berikut khusus untuk AWS PCS dan mengontrol bagaimana AWS PCS mengimplementasikan akuntansi Slurm.

Database akuntansi

AWS PCS menyimpan data akuntansi Anda dalam database yang dibuat dalam Akun AWS yang AWS memiliki. Anda tidak memiliki akses ke `slurmdbd.conf`.

Waktu pembersihan default

Pengaturan AWS PCS ini menentukan periode retensi (dalam hari) untuk semua jenis catatan akuntansi (pekerjaan, acara, reservasi, langkah, penangguhan, transaksi, data penggunaan). Misalnya, jika nilainya 30, AWS PCS menyimpan catatan akuntansi selama 30 hari. Anda memberikan nilai ini saat membuat cluster. Jika Anda tidak memberikan nilai, AWS PCS menyimpan catatan akuntansi dalam database tanpa batas waktu.

AWS PCS console

Anda menentukan waktu pembersihan default sebagai bagian dari langkah-langkah untuk membuat cluster. Pada halaman Buat cluster, Anda harus memilih versi Slurm yang valid (versi 24.11 atau yang lebih baru) dan mengaktifkan akuntansi. Di bawah pengaturan Scheduler, berikan nilai integer untuk waktu pembersihan Default (hari).

AWS PCS API

Tentukan `defaultPurgeTimeInDays` sebagai bagian dari `accounting` informasi yang Anda berikan dalam panggilan Anda ke tindakan `CreateCluster` API. Untuk informasi selengkapnya, lihat [CreateCluster](#) dan [Akuntansi](#) di Referensi AWS PCS API.

Note

Saat Anda menggunakan AWS PCS API untuk membuat cluster, nilai default untuk `defaultPurgeTimeInDays` adalah `-1` dan `0` bukan nilai yang valid.

Penegakan kebijakan akuntansi

Pengaturan ini menentukan seberapa ketat Slurm memberlakukan aturan pengiriman pekerjaan, batasan sumber daya, dan kebijakan akuntansi untuk klaster Anda. Pengaturan ini sesuai dengan `AccountingStorageEnforce` parameter dalam `slurm.conf` file cluster Anda. Anda dapat memilih kombinasi opsi penegakan hukum. Jika Anda tidak memilih opsi apa pun, tidak ada kendala akuntansi yang diterapkan pada pekerjaan di cluster. AWS PCS mendukung opsi berikut:

- asosiasi — job-to-account pemetaan
- batas — kendala sumber daya
- QoS — kualitas persyaratan layanan
- mode aman - dijamin penyelesaian dalam batas
- nosteps - nonaktifkan akuntansi langkah
- nojobs - nonaktifkan akuntansi pekerjaan

Untuk informasi selengkapnya tentang opsi ini, lihat [dokumentasi Slurm di SchedMD](#).

AWS PCS console

Anda mengatur opsi sebagai bagian dari langkah-langkah untuk membuat cluster. Pada halaman Buat cluster, Anda harus memilih versi Slurm yang valid (versi 24.11 atau yang lebih baru) dan mengaktifkan akuntansi. Pilih opsi yang Anda inginkan dari daftar dropdown penegakan kebijakan akuntansi di bawah Pengaturan penjadwal.

AWS PCS API

Di Slurm, opsi ini diatur dalam file cluster. `slurm.conf` Anda tidak memiliki akses langsung ke untuk klaster AWS PCS Anda. `slurm.conf` Sebagai gantinya, Anda `SlurmCustomSettings` memberikan tindakan `CreateCluster` API saat membuat klaster. Untuk informasi selengkapnya, lihat [CreateCluster](#) di Referensi AWS PCS API.

Dapatkan konfigurasi akuntansi untuk cluster AWS PCS yang ada

Konfigurasi akuntansi Slurm disertakan dalam konfigurasi Slurm untuk cluster Anda.

AWS PCS console

1. Pilih Cluster dari panel navigasi.
2. Pilih nama cluster dari daftar.
3. Pada tab Konfigurasi, temukan konfigurasi akuntansi di bawah konfigurasi Slurm

AWS PCS API

Gunakan aksi `GetCluster` API untuk mendapatkan konfigurasi cluster. Anda dapat menemukan konfigurasi akuntansi di `slurmConfiguration`. Pengaturan untuk mode dan nilai `defaultPurgeTimeInDays` berada di bawah `accounting`. Opsi penegakan kebijakan akuntansi yang dipilih ada di bawah `slurmCustomSettings`. Untuk informasi selengkapnya, lihat [GetCluster](#) di Referensi AWS PCS API.

Slurm REST API di PCS AWS

AWS PCS menyediakan dukungan terkelola untuk REST API asli Slurm melalui `slurmrestd`, memberikan antarmuka HTTP untuk interaksi cluster terprogram. Anda dapat mengirimkan pekerjaan, memantau status klaster, dan mengelola sumber daya melalui permintaan HTTP standar tanpa memerlukan akses shell langsung ke klaster Anda.

Kasus penggunaan umum

Slurm REST API mendukung berbagai skenario integrasi:

- Integrasi Aplikasi Web: Membangun frontend kustom dan aplikasi web yang mengirimkan dan mengelola pekerjaan secara langsung.
- Integrasi Notebook Jupyter: Memungkinkan peneliti mengirimkan pekerjaan dari lingkungan notebook tanpa meninggalkan alur kerja pengembangan mereka.
- Integrasi Solusi Mitra: Hubungkan alat HPC pihak ketiga dan manajer alur kerja ke kluster AWS PCS Anda.
- Manajemen Cluster Terprogram: Mengotomatiskan alur kerja pengajuan pekerjaan, pemantauan, dan manajemen sumber daya.
- Alur Kerja Komputasi Penelitian: Mendukung lingkungan penelitian akademis dan perusahaan yang membutuhkan manajemen pekerjaan berbasis API.

Persyaratan dan pembatasan

Sebelum menggunakan Slurm REST API, tinjau detail berikut:

- Cluster Anda harus menggunakan Slurm versi 25.05 atau lebih tinggi.
- Titik akhir API hanya akan dapat diakses melalui alamat IP pribadi dalam VPC kluster Anda.
- Grup keamanan cluster Anda harus mengizinkan lalu lintas HTTP pada port 6820.
- Otentikasi membutuhkan token JWT dengan klaim identitas pengguna tertentu.

Keterbatasan saat ini meliputi:

- Token yang dihasilkan `scontrol` token oleh tidak didukung.
- `X-SLURM-USER-NAME` peniruan header tidak tersedia.
- Beberapa fungsi memerlukan akuntansi Slurm untuk diaktifkan.
- Tidak kompatibel dengan mekanisme plugin filter Slurm CLI.
- Koneksi ke titik akhir REST API tidak dienkripsi dengan TLS.

Topik

- [Mengaktifkan Slurm REST API di PCS AWS](#)

- [Mengautentikasi dengan Slurm REST API di PCS AWS](#)
- [Menggunakan Slurm REST API untuk manajemen pekerjaan di PCS AWS](#)
- [Slurm REST API pertanyaan yang sering diajukan di PCS AWS](#)

Mengaktifkan Slurm REST API di PCS AWS

Aktifkan Slurm REST API untuk mengakses antarmuka HTTP cluster Anda untuk manajemen dan pemantauan pekerjaan terprogram. Anda dapat mengaktifkan fitur ini selama pembuatan klaster atau memperbarui cluster yang ada yang memenuhi persyaratan.

Prasyarat

Sebelum mengaktifkan Slurm REST API, pastikan Anda memiliki:

- Versi cluster: Slurm versi 25.05 atau lebih tinggi.
- Grup keamanan: Aturan yang memungkinkan lalu lintas HTTP pada port 6820 dari sumber yang Anda inginkan.

Prosedur

Untuk mengaktifkan Slurm REST API di cluster baru

Konsol Manajemen AWS

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Pilih Buat klaster.
3. Di bawah detail Cluster, pilih Slurm versi 25.05 atau lebih tinggi.
4. Konfigurasi pengaturan cluster lainnya sesuai kebutuhan.
5. Di bagian konfigurasi Scheduler, atur REST API ke Enabled.
6. Konfigurasi grup keamanan klaster Anda untuk memungkinkan lalu lintas HTTP pada port 6820 dari sumber yang Anda inginkan.
7. Selesaikan proses pembuatan cluster.

AWS CLI

1. Tambahkan konfigurasi Slurm REST saat membuat cluster Anda.

```
aws pcs create-cluster --region region \  
  --cluster-name my-cluster \  
  --scheduler type=SLURM, version=25.05 \  
  --size SMALL \  
  --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1 \  
  --slurm-configuration slurmRest='{mode=STANDARD}'
```

2. Konfigurasi grup keamanan kluster Anda untuk memungkinkan lalu lintas HTTP pada port 6820 dari sumber yang Anda inginkan.

Untuk mengaktifkan Slurm REST API pada cluster yang ada

Konsol Manajemen AWS

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Pilih kluster Anda dari daftar.
3. Verifikasi bahwa kluster Anda menggunakan Slurm versi 25.05 atau lebih tinggi dalam detail kluster.
4. Pilih Edit cluster.
5. Di bagian konfigurasi Scheduler, atur REST API ke Enabled.
6. Pilih Perbarui kluster untuk menerapkan perubahan.
7. Konfigurasi grup keamanan kluster Anda untuk memungkinkan lalu lintas HTTP pada port 6820 dari sumber yang Anda inginkan.

AWS CLI

1. Perbarui cluster Anda dengan konfigurasi Slurm REST, seperti dalam contoh ini.

```
aws pcs update-cluster --cluster-identifier my-cluster \  
  --slurm-configuration 'slurmRest={mode=STANDARD}'
```

2. Konfigurasi grup keamanan kluster Anda untuk memungkinkan lalu lintas HTTP pada port 6820 dari sumber yang Anda inginkan.

Apa yang terjadi setelah mengaktifkan

Saat Anda mengaktifkan REST API, AWS PCS secara otomatis:

- Menghasilkan kunci penandatanganan JWT dan menyimpannya di AWS Secrets Manager.
- Mengekspos titik akhir API di `https://<clusterPrivateIpAddress>:6820` dalam VPC Anda.
- Memperbarui konfigurasi klaster Anda untuk menampilkan detail titik akhir REST API.

Anda sekarang dapat mengautentikasi dan menggunakan REST API untuk manajemen pekerjaan dan operasi klaster.

Mengautentikasi dengan Slurm REST API di PCS AWS

Slurm REST API di AWS PCS menggunakan otentikasi JSON Web Token (JWT) untuk memastikan akses aman ke sumber daya cluster Anda. AWS PCS menyediakan kunci penandatanganan terkelola yang disimpan di AWS Secrets Manager, yang Anda gunakan untuk menghasilkan token JWT yang berisi klaim identitas pengguna tertentu.

Prasyarat

Sebelum mengautentikasi dengan Slurm REST API, pastikan Anda memiliki:

- Konfigurasi cluster: Cluster AWS PCS dengan Slurm 25.05+ dan REST API diaktifkan.
- Izin AWS: Akses ke AWS Secrets Manager untuk kunci penandatanganan JWT.
- Informasi pengguna: Nama pengguna, ID pengguna POSIX, dan satu atau lebih grup POSIX IDs untuk akun cluster Anda.
- Akses jaringan: Konektivitas dalam VPC cluster Anda dengan grup keamanan yang memungkinkan port 6820.

Prosedur

Untuk mengambil alamat titik akhir Slurm REST API

Konsol Manajemen AWS

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Pilih klaster Anda dari daftar.
3. Dalam detail konfigurasi cluster, cari bagian Endpoints.
4. Perhatikan alamat IP pribadi dan port untuk Slurm REST API (slurmrestd).

5. Anda dapat melakukan panggilan API dengan mengirimkan permintaan HTTP yang diformat dengan benar ke alamat ini.

AWS CLI

1. Kueri status klaster Anda dengan `aws pcs get-cluster`. Cari SLURMRESTD titik akhir di `endpoints` bidang dalam respons. Inilah contohnya:

```
"endpoints": [  
  {  
    "type": "SLURMCTLD",  
    "privateIpAddress": "192.0.2.1",  
    "port": "6817"  
  },  
  {  
    "type": "SLURMRESTD",  
    "privateIpAddress": "192.0.2.1",  
    "port": "6820"  
  }  
]
```

2. Anda dapat melakukan panggilan API dengan mengirimkan permintaan HTTP yang diformat dengan benar `http://<privateIpAddress>:<port>/`

Untuk mengambil kunci penandatanganan JWT

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Pilih klaster Anda dari daftar.
3. Dalam detail konfigurasi cluster, cari bagian Authentication Scheduler.
4. Perhatikan ARN dan versi kunci JSON Web Token (JWT).
5. Gunakan tombol AWS CLI untuk mengambil kunci penandatanganan dari Secrets Manager:

```
aws secretsmanager get-secret-value --secret-id arn:aws:secretsmanager:region:account:secret:name --version-id version
```

Untuk menghasilkan token JWT

1. Buat JWT dengan klaim wajib berikut:

- `exp`— Waktu kedaluwarsa dalam hitungan detik sejak 1970 untuk JWT
 - `iat`— Waktu saat ini dalam hitungan detik sejak 1970
 - `sun`— Nama pengguna untuk otentikasi
 - `uid`— ID pengguna POSIX
 - `gid`— ID grup POSIX
 - `id`— Properti identitas POSIX tambahan
 - `gecos`— Kolom komentar pengguna, sering digunakan untuk menyimpan nama yang dapat dibaca manusia
 - `dir`— Direktori home pengguna
 - `shell`— Shell default pengguna
 - `gids`— Daftar grup POSIX tambahan IDs tempat pengguna berada
2. Tanda tangani JWT menggunakan kunci penandatanganan yang diambil dari Secrets Manager.
 3. Tetapkan waktu kedaluwarsa yang sesuai untuk token.

Note

Sebagai alternatif untuk `sun` klaim, Anda dapat memberikan salah satu dari yang berikut:

- `username`
- Nama bidang kustom yang Anda tentukan melalui `userclaimfield` di `AuthAltParameters Slurm custom settings`
- `nameBidang` dalam `id` klaim

Untuk mengautentikasi permintaan API

1. Sertakan token JWT dalam permintaan HTTP Anda menggunakan salah satu metode berikut:
 - Token pembawa - Tambahkan header `Authorization: Bearer <jwt>`
 - Slurm header - Tambahkan header `X-SLURM-USER-TOKEN: <jwt>`
2. Buat permintaan HTTP ke titik akhir REST API:

Berikut adalah contoh mengakses `/ping` API menggunakan `curl` dan header. `Authorized: Bearer`

```
curl -X GET -H "Authorization: Bearer <jwt>" \  
http://<privateIpAddress>:6820/slurm/v0.0.43/ping
```

Contoh generasi JWT

Ambil kunci penandatanganan JWT cluster AWS PCS dan simpan sebagai file lokal. Ganti nilai untuk `aws-region`, `secret-arn`, dan versi rahasia dengan nilai yang sesuai untuk klaster Anda.

```
#!/bin/bash  
SECRET_KEY=$(aws secretsmanager get-secret-value \  
--region aws-region \  
--secret-id secret-arn \  
--version-stage secret-version \  
--query 'SecretString' \  
--output text)  
echo "$SECRET_KEY" | base64 --decode > jwt.key
```

Contoh Python ini menggambarkan cara menggunakan kunci penandatanganan untuk menghasilkan token JWT:

```
#!/usr/bin/env python3  
  
import sys  
import os  
import pprint  
import json  
import time  
from datetime import datetime, timedelta, timezone  
from jwt import JWT  
from jwt.jwa import HS256  
from jwt.jwk import jwk_from_dict  
from jwt.utils import b64decode, b64encode  
if len(sys.argv) != 3:  
    sys.exit("Usage: gen_jwt.py [jwt_key_file] [expiration_time_seconds]")  
SIGNING_KEY = sys.argv[1]  
EXPIRATION_TIME = int(sys.argv[2])  
with open(SIGNING_KEY, "rb") as f:  
    priv_key = f.read()  
signing_key = jwk_from_dict({  
    'kty': 'oct',
```

```
'k': b64encode(priv_key)
})
message = {
    "exp": int(time.time() + EXPIRATION_TIME),
    "iat": int(time.time()),
    "sun": "ec2-user",
    "uid": 1000,
    "gid": 1000,
    "id": {
        "gecos": "EC2 User",
        "dir": "/home/ec2-user",
        "gids": [1000],
        "shell": "/bin/bash"
    }
}
a = JWT()
compact_jws = a.encode(message, signing_key, alg='HS256')
print(compact_jws)
```

Skrip akan mencetak JWT ke layar.

```
abcdefghijklmnopqrst...
```

Menggunakan Slurm REST API untuk manajemen pekerjaan di PCS AWS

Ikhtisar Slurm REST API

Slurm REST API menyediakan akses terprogram ke fungsi manajemen cluster melalui permintaan HTTP. Memahami karakteristik utama ini akan membantu Anda menggunakan API dengan AWS PCS secara efektif:

- **Protokol Akses:** API menggunakan HTTP (bukan HTTPS) untuk komunikasi dalam jaringan pribadi cluster Anda.
- **Detail Koneksi:** Akses API menggunakan alamat IP pribadi klaster Anda dan `slurmrestd` port (biasanya 6820). Format URL dasar lengkap adalah `http://<privateIpAddress>:6820`.
- **Pembuatan Versi API:** Versi API sesuai dengan instalasi Slurm Anda. Untuk Slurm 25.05, gunakan versi v0.0.43. Nomor versi berubah dengan setiap rilis Slurm. Anda dapat menemukan versi API yang saat ini didukung di catatan [rilis Slurm](#).

- Struktur URL: Struktur URL untuk Slurm REST API adalah.
`http://<privateIpAddress>:<port>/<api-version>/<endpoint>` Informasi penggunaan terperinci untuk endpoint REST API dapat ditemukan di dokumentasi [Slurm](#).

Prasyarat

Sebelum menggunakan Slurm REST API, pastikan Anda memiliki:

- Konfigurasi cluster: Cluster AWS PCS dengan Slurm 25.05+ dan REST API diaktifkan.
- Otentikasi: Token JWT yang valid dengan klaim identitas pengguna yang tepat.
- Akses jaringan: Konektivitas dalam VPC cluster Anda dengan grup keamanan yang memungkinkan port 6820.

Prosedur

Untuk mengirimkan pekerjaan menggunakan REST API

1. Buat permintaan pengiriman pekerjaan dengan parameter yang diperlukan:

```
{
  "job": {
    "name": "my-job",
    "partition": "compute",
    "nodes": 1,
    "tasks": 1,
    "script": "#!/bin/bash\nnecho 'Hello from Slurm REST API'"
  }
}
```

2. Kirim pekerjaan menggunakan permintaan HTTP POST:

```
curl -X POST \
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '<job-json>' \
  https://<privateIpAddress>:6820/slurm/v0.0.43/job/submit
```

3. Perhatikan ID pekerjaan yang dikembalikan dalam respons untuk tujuan pemantauan.

Untuk memantau status pekerjaan

1. Dapatkan informasi tentang pekerjaan tertentu:

```
curl -X GET -H "Authorization: Bearer <jwt>" \  
https://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

2. Daftar semua pekerjaan untuk pengguna yang diautentikasi:

```
curl -X GET -H "Authorization: Bearer <jwt>" \  
https://<privateIpAddress>:6820/slurm/v0.0.43/jobs
```

Untuk membatalkan tugas

- Kirim permintaan DELETE untuk membatalkan pekerjaan tertentu:

```
curl -X DELETE -H "Authorization: Bearer <jwt>" \  
https://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

Slurm REST API pertanyaan yang sering diajukan di PCS AWS

Bagian ini menjawab pertanyaan umum tentang Slurm REST API di AWS PCS.

Apa itu Slurm REST API?

Slurm REST API adalah antarmuka HTTP yang memungkinkan Anda berinteraksi dengan manajer beban kerja Slurm secara terprogram. Anda dapat menggunakan metode HTTP standar seperti GET, POST, dan DELETE untuk mengirimkan pekerjaan, memantau status kluster, dan mengelola sumber daya tanpa memerlukan akses baris perintah ke cluster.

Dapatkah saya menggunakan token yang dihasilkan oleh **scontrol token**?

Tidak, `scontrol token` output standar tidak kompatibel dengan AWS PCS. PCS Slurm REST API memerlukan token JWT yang diperkaya yang berisi klaim identitas spesifik yang mencakup username (`sun`), POSIX user ID (`uid`), dan group (`gid`). IDs `gid` Token Slurm standar tidak memiliki klaim yang diperlukan ini dan akan ditolak oleh API.

Dapatkah saya mengakses API dari luar VPC saya?

Tidak, titik akhir REST API hanya dapat diakses dari dalam VPC Anda menggunakan alamat IP pribadi pengontrol Slurm. Untuk mengaktifkan akses eksternal, terapkan AWS layanan seperti Application Load Balancer dengan VPC Link, API Gateway, atau buat koneksi VPC peering atau VPN untuk konektivitas yang aman.

Mengapa API menggunakan HTTP alih-alih HTTPS?

Slurm REST API dimaksudkan untuk menjadi endpoint internal dalam jaringan pribadi cluster Anda. Untuk penerapan produksi yang memerlukan enkripsi, Anda dapat menerapkan SSL/TLS penghentian pada tingkat yang lebih tinggi dalam arsitektur Anda, seperti melalui gateway API, penyeimbang beban, atau proxy terbalik.

Bagaimana cara mengontrol akses ke REST API?

Konfigurasi aturan grup keamanan kluster Anda untuk membatasi akses ke port 6820 pada pengontrol Slurm. Tetapkan aturan masuk untuk mengizinkan koneksi hanya dari rentang IP terpercaya atau sumber tertentu dalam VPC Anda, memblokir akses tidak sah ke titik akhir API.

Bagaimana cara memutar kunci penandatanganan JWT?

Letakkan kluster Anda dalam mode pemeliharaan tanpa instance aktif, lalu mulai rotasi kunci melalui AWS Secrets Manager. Setelah rotasi selesai, aktifkan kembali antrian. Semua token JWT yang ada akan menjadi tidak valid dan harus dibuat ulang menggunakan kunci penandatanganan baru dari Secrets Manager.

Apakah saya perlu mengaktifkan akuntansi Slurm untuk menggunakan REST API?

Tidak, akuntansi Slurm tidak diperlukan untuk operasi REST API dasar seperti pengajuan dan pemantauan pekerjaan. Namun, seluruh `/slurmdb` titik akhir membutuhkan akuntansi untuk aktif.

Alat pihak ketiga apa yang bekerja dengan AWS PCS REST API?

Banyak klien Slurm REST API yang ada harus bekerja dengan AWS PCS, termasuk Slurm Exporter untuk Prometheus, dan aplikasi khusus yang mengikuti format API Slurm REST SlurmWeb standar. Namun, alat yang diandalkan `scontrol` token untuk otentikasi akan memerlukan modifikasi untuk bekerja dengan persyaratan AWS PCS JWT.

Apakah ada biaya tambahan untuk menggunakan REST API?

Tidak, tidak ada biaya tambahan untuk mengaktifkan atau menggunakan fitur Slurm REST API. Anda hanya membayar sumber daya cluster yang mendasarinya seperti biasa.

Bagaimana saya bisa memecahkan masalah REST API?

- Masalah konektivitas jaringan

Jika Anda tidak dapat mencapai titik akhir API, Anda akan melihat batas waktu koneksi atau kesalahan “koneksi ditolak” saat membuat permintaan HTTP ke pengontrol cluster.

Apa yang harus dilakukan: Verifikasi klien Anda berada di VPC yang sama atau memiliki perutean jaringan yang tepat, dan konfirmasi grup keamanan Anda mengizinkan lalu lintas HTTP pada port 6820 dari IP sumber atau subnet Anda.

- Masalah otentikasi Slurm REST

Jika token JWT Anda tidak valid, kedaluwarsa, atau ditandatangani dengan tidak benar, permintaan API akan mengembalikan “Kesalahan otentikasi protokol” di bidang kesalahan respons.

Contoh pesan kesalahan:

```
{
  "errors": [
    {
      "description": "Batch job submission failed",
      "error_number": 1007,
      "error": "Protocol authentication error",
      "source": "slurm_submit_batch_job()"
    }
  ]
}
```

Apa yang harus dilakukan: Periksa apakah token JWT Anda diformat dengan benar, tidak kedaluwarsa, dan ditandatangani dengan kunci yang benar dari Secrets Manager. Verifikasi bahwa token telah dibentuk dengan benar dan menyertakan klaim yang diperlukan dan bahwa Anda menggunakan format header otentikasi yang benar.

- Job gagal berjalan setelah pengajuan

Jika token JWT Anda valid tetapi berisi struktur atau konten internal yang salah, pekerjaan mungkin telah memasukkan status jeda (PD) dengan kode alasan. JobAdminHead Gunakan `scontrol show job <job-id>` untuk memeriksa pekerjaan — Anda akan melihat `JobState=PENDING`, `Reason=JobHeldAdmin`, dan `SystemComment=slurm_cred_create failure, holding job`.

Apa yang harus dilakukan: Akar penyebabnya mungkin nilai yang salah di JWT. Verifikasi bahwa token terstruktur dengan benar dan menyertakan klaim yang diperlukan sesuai dokumentasi PCS.

- Masalah izin direktori kerja

Jika identitas pengguna yang ditentukan dalam JWT Anda tidak memiliki izin menulis ke direktori kerja pekerjaan, pekerjaan akan gagal dengan kesalahan izin, mirip dengan menggunakan `sbatch --chdir` dengan direktori yang tidak dapat diakses.

Apa yang harus dilakukan: Pastikan pengguna yang ditentukan dalam token JWT Anda memiliki izin yang sesuai untuk direktori kerja pekerjaan.

Mem-boot ulang node komputasi dengan Slurm di PCS AWS

AWS PCS mendukung perintah asli `scontrol reboot` Slurm. Gunakan perintah ini untuk me-reboot node komputasi tanpa penggantian instans EC2. Metode reboot lainnya (konsol Amazon EC2, patch otomatis AWS CLI, atau pemeliharaan sistem) menyebabkan AWS PCS menganggap instans EC2 tidak sehat dan menggantinya.

Manfaat reboot Slurm

Slurm reboot memberikan beberapa keuntungan untuk pemeliharaan cluster:

- Pertahankan kapasitas — Hindari kehilangan instans EC2 yang dibatasi kapasitas ke pelanggan lain.
- Mengurangi biaya — Hilangkan siklus penggantian instans yang tidak perlu dan penagihan lanjutan untuk node idle.
- Pemulihan lebih cepat - Tidak ada penundaan penyediaan dibandingkan dengan penggantian instans.
- Fleksibilitas operasional — Hapus kebocoran memori, hapus file sementara, dan pulihkan node dari status terdegradasi.

Kapan menggunakan reboot Slurm

Gunakan reboot Slurm untuk skenario pemeliharaan operasional umum:

- Pemecahan masalah — Mengatasi masalah kinerja atau proses yang tidak responsif, terutama untuk node GPU.
- Pembersihan sumber daya — Hapus kebocoran memori, file sementara/tmp, atau proses macet yang memengaruhi kinerja pekerjaan.
- Pemulihan — Memulihkan node dari status hang atau terdegradasi sebelum membutuhkan penggantian node penuh.

Batasan

- Hanya pengguna Slurm Admin (pengguna root) yang dapat menjalankan perintah reboot.
- Dukungan reboot terbatas `scontrol reboot` hanya.
- RebootProgram konfigurasi tidak didukung.
- Tidak ada antarmuka konsol — hanya baris perintah.

Topik

- [Reboot node komputasi menggunakan Slurm di PCS AWS](#)
- [Batalkan reboot yang tertunda di AWS PCS](#)
- [Slurm reboot pertanyaan yang sering diajukan di PCS AWS](#)
- [Memecahkan masalah reboot Slurm di PCS AWS](#)

Reboot node komputasi menggunakan Slurm di PCS AWS

Gunakan perintah reboot asli Slurm untuk menyelesaikan masalah kinerja, menghapus masalah sumber daya, atau memulihkan dari status terdegradasi tanpa kehilangan kapasitas instans EC2.

Prasyarat

- Hak istimewa Admin Slurm (akses pengguna root)
- Akses ke node login di cluster AWS PCS


Prosedur

1. Connect ke node login melalui konsol EC2.

- a. Di konsol EC2, pilih Instans.
 - b. Pilih instance node login Anda.
 - c. Pilih Hubungkan.
2. Identifikasi nama node komputasi target menggunakan `sinfo` atau `scontrol show node`.

```
sinfo
# or
scontrol show node
```

3. Jalankan perintah reboot menggunakan salah satu opsi ini:

 Warning

Jangan gunakan `nextstate=DOWN` dengan `scontrol reboot` perintah. Parameter ini menandai node sebagai tidak sehat dan memicu penggantian instance.

- Reboot dasar (menunggu node menjadi idle):

```
scontrol reboot nodename
```

- Reboot segera (menguras node dan reboot saat pekerjaan selesai):

```
scontrol reboot ASAP nodename
```

- Reboot dengan alasan:

```
scontrol reboot ASAP reason="troubleshooting" nodename
```

- Reboot dengan status resume:

```
scontrol reboot ASAP nextstate=RESUME nodename
```

4. Pantau kemajuan reboot menggunakan `scontrol show node`.

```
scontrol show node nodename
```

5. Verifikasi node kembali ke layanan setelah reboot selesai.

Batalkan reboot yang tertunda di AWS PCS

Batalkan reboot yang tertunda untuk menghindari downtime yang tidak perlu saat masalah telah diselesaikan atau saat reboot tidak lagi diperlukan.

Prasyarat

- Hak istimewa Admin Slurm
- Node harus memiliki reboot yang tertunda (menunjukkan status “reboot dikeluarkan”)
- Akses ke node login untuk eksekusi perintah

Prosedur

1. Connect ke node login.
2. Verifikasi bahwa node memiliki reboot yang tertunda menggunakan `scontrol show node`.

```
scontrol show node nodename
```

Cari “reboot issued” di status node.

3. Jalankan perintah batal.

```
scontrol cancel_reboot nodename
```

4. Verifikasi pembatalan reboot dan status node kembali normal.

```
scontrol show node nodename
```

Slurm reboot pertanyaan yang sering diajukan di PCS AWS

Temukan jawaban atas pertanyaan umum tentang penggunaan reboot Slurm di AWS PCS.

Apa itu dukungan reboot slurm?

Support untuk perintah Slurm `scontrol reboot` asli. Gunakan perintah ini untuk me-reboot node komputasi tanpa penggantian instans otomatis, yang mempertahankan kapasitas instans EC2 dan mengurangi biaya operasional.

Siapa yang dapat menggunakan perintah reboot Slurm?

Hanya pengguna Slurm Admin (pengguna root) yang dapat menjalankan perintah reboot. Pengguna biasa yang mencoba menggunakan `scontrol reboot` akan menerima kesalahan izin ditolak dari Slurm tanpa mempengaruhi node.

Apa yang terjadi dengan menjalankan pekerjaan selama reboot?

Secara default, pekerjaan selesai secara normal sebelum reboot terjadi. Dengan opsi `ASAP`, node dikeringkan untuk mencegah pekerjaan baru, dan reboot terjadi setelah pekerjaan saat ini selesai. Pekerjaan dapat dibatalkan atau diminta untuk reboot segera.

Apa bedanya dengan reboot konsol EC2?

Slurm reboot mempertahankan instans EC2 dan menghindari penggantian, sementara reboot konsol EC2 memicu PCS untuk mengganti instance karena pemeriksaan kesehatan yang gagal selama proses reboot.

Bisakah saya mengonfigurasi skrip reboot khusus?

Tidak, `RebootProgram` konfigurasi tidak didukung dalam rilis awal. Fitur ini menggunakan perilaku reboot Slurm standar tanpa dukungan skrip khusus.

Berapa lama waktu yang dibutuhkan reboot slurm?

Waktu reboot bervariasi berdasarkan jenis instans, proses boot pelanggan, konfigurasi AMI, dan apakah pekerjaan perlu diselesaikan terlebih dahulu. Prosesnya termasuk menunggu pekerjaan selesai, reboot fisik, pemeriksaan kesehatan, dan pendaftaran daemon `slurmd`.

Bisakah saya melihat sejarah reboot?

Peristiwa reboot direkam dalam log Slurm (`slurmctld` dan `slurmd`) yang dapat dipantau melalui CloudWatch Bidang alasan dalam status node menunjukkan alasan reboot selama proses berlangsung.

Bagaimana jika sebuah node macet saat reboot?

Jika sebuah node tidak menyelesaikan proses reboot di dalamnya `ResumeTimeout`, itu akan ditandai sebagai `DOWN`. Periksa CloudWatch log untuk kesalahan, verifikasi konektivitas jaringan, dan periksa log `slurmd`. Hubungi AWS Support jika masalah terus berlanjut.

Bisakah saya me-reboot beberapa node sekaligus?

Ya, Anda dapat menentukan beberapa node dalam perintah reboot:

```
scontrol reboot ASAP node1,node2,node3
```

Bagaimana saya bisa me-reboot node tanpa menunggu pekerjaan selesai?

Untuk reboot node langsung saat menghadapi masalah seperti node bermasalah yang memengaruhi pekerjaan multi-node, penurunan kinerja yang signifikan, atau perilaku GPU yang tidak stabil, Anda memiliki dua opsi:

- **Batalkan dan Reboot** — Pertama, batalkan pekerjaan yang terpengaruh menggunakan `scontrol <job_id>`, lalu memulai reboot langsung menggunakan `scontrol reboot ASAP <nodename>`. Pekerjaan yang sedang berjalan akan dihentikan dan perlu dikirimkan kembali setelah node pulih.
- **Drain dan Requeue** (kurang berdampak) - Mulailah dengan memulai drain dan reboot dengan `scontrol reboot ASAP <nodename>`, lalu requeue pekerjaan yang terpengaruh menggunakan `scontrol requeue <job_id>` Ini menempatkan pekerjaan kembali ke status tertunda alih-alih membatalkannya.

Apa yang terjadi jika saya menentukan `nextState=down`?

Jika Anda menentukan `nextState=DOWN`, node akan ditandai sebagai tidak sehat setelah reboot dan memicu penggantian instance. Untuk menghindari penggantian instance, jangan tentukan `nextState` atau gunakan `nextState=RESUME`

Sumber daya tambahan

- Untuk prosedur reboot dasar, lihat [Reboot node komputasi menggunakan Slurm di PCS AWS](#).
- Untuk mengatasi masalah reboot, lihat [Memecahkan masalah reboot Slurm di PCS AWS](#)
- Untuk dokumentasi reboot Slurm, lihat Dokumentasi [Slurm](#) `scontrol`.

Memecahkan masalah reboot Slurm di PCS AWS

Saat Anda mengalami masalah reboot node, periksa dulu status node menggunakan `scontrol show node <nodename>`. Kemudian periksa CloudWatch log untuk Slurm (`slurmctld` dan `slurmd`) dan log sistem untuk mengidentifikasi potensi kesalahan.

Untuk pemecahan masalah dasar, verifikasi konektivitas jaringan, periksa pengaturan grup keamanan, dan pastikan semua layanan yang diperlukan berjalan setelah reboot. Jika masalah berlanjut setelah langkah pemecahan masalah dasar, hubungi Support. AWS Saat menjangkau

dukungan, berikan kutipan log yang relevan, informasi status node, dan garis waktu upaya reboot untuk membantu mempercepat proses resolusi.

Sumber daya tambahan

- Untuk memantau instans AWS PCS yang menggunakan CloudWatch, lihat [Memantau instans AWS PCS menggunakan Amazon CloudWatch](#).
- Untuk pemecahan masalah umum, lihat [Memecahkan masalah di Layanan Komputasi AWS Paralel](#).
- Untuk dokumentasi Slurm, lihat Panduan Pemecahan Masalah [Slurm](#).

Mengkonfigurasi pengaturan Slurm khusus di PCS AWS

Gunakan pengaturan Slurm khusus untuk mengonfigurasi parameter Slurm tambahan di seluruh sumber daya Cluster, Antrian, dan Grup Node Komputasi. Rilis ini menambahkan dukungan untuk pengaturan Slurm pada sumber daya Antrian, memberikan kontrol terperinci atas perilaku khusus partisi.

Manfaat pengaturan Slurm kustom

Pengaturan Slurm khusus memberikan kontrol canggih atas lingkungan AWS HPC berbasis PC Anda. Anda dapat menerapkan akuntansi terperinci, menegakkan kontrol akses, dan mengoptimalkan eksekusi beban kerja melalui quality-of-service konfigurasi dan kebijakan preemption. Kemampuan ini memastikan pekerjaan penting menerima sumber daya yang diperlukan sambil mempertahankan pemanfaatan cluster yang efisien. Baik Anda mengelola beban kerja yang dipercepat GPU, menerapkan penjadwalan pembagian yang adil, atau mengontrol siklus hidup pekerjaan, pengaturan khusus membantu menyelaraskan infrastruktur HPC Anda dengan persyaratan operasional dan tujuan penelitian.

Mengkonfigurasi pengaturan khusus

Pengaturan Slurm khusus dapat dikonfigurasi melalui AWS Konsol, CLI, atau SDKs selama pembuatan sumber daya atau dimodifikasi nanti melalui operasi pembaruan.

Konsol Manajemen AWS

Arahkan ke pengaturan penjadwal tambahan di halaman buat atau edit untuk jenis sumber daya apa pun (cluster, antrian, atau grup node komputasi).

Untuk menambahkan pengaturan baru

1. Pilih Tambahkan pengaturan baru.
2. Pilih nama Parameter dari dropdown (yang mencakup deskripsi parameter singkat).
3. Berikan nilai yang sesuai.

Untuk membatalkan setelan kustom

1. Pilih Hapus di sebelah parameter/value pasangan yang relevan.
2. Buat atau perbarui sumber daya.

AWS CLI

Untuk manajemen terprogram pengaturan kustom, gunakan `SlurmCustomSettings` bidang dalam membuat atau memperbarui operasi.

Example— Memperbarui Prolog parameter pada cluster

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/prolog.sh"}]'
```

Example— Mengatur antrian untuk berada Default di cluster

```
aws pcs update-queue \  
--cluster-identifier my-cluster \  
--queue-identifier my-queue \  
--slurm-configuration \  
'SlurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Example— Mengatur kustom Features pada grup node komputasi

```
aws pcs update-compute-node-group \  
--cluster-identifier my-cluster \  
--compute-node-group-identifier my-cng-1 \  
--slurm-configuration \  
'SlurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

Validasi dan penanganan kesalahan

AWS PCS mengimplementasikan proses validasi berlapis-lapis untuk pengaturan Slurm kustom. Selama operasi membuat dan memperbarui, kami melakukan validasi sinkron yang mencakup:

- Pemeriksaan tingkat lapangan: Kami memvalidasi pengaturan individual untuk tipe data yang benar, nilai yang diizinkan, dan persyaratan format. Misalnya, kami memastikan nilai waktu berada dalam format Slurm yang benar dan nilai boolean menggunakan representasi boolean Slurm yang diterima.
- Validasi konteks-sadar: Beberapa pengaturan diperiksa terhadap konteks konfigurasi yang lebih luas. Misalnya, parameter tertentu hanya valid ketika akuntansi Slurm diaktifkan.
- Konsistensi antar-pengaturan: Kami memverifikasi bahwa opsi yang saling eksklusif tidak disetel bersama dan pengaturan yang saling bergantung dikonfigurasi dengan benar.

Jika validasi gagal, Anda akan menerima `ValidationException` dengan kode kesalahan tertentu (misalnya, `InvalidInput`), pesan kesalahan yang jelas yang menjelaskan masalah, dan daftar bidang yang tidak valid dan detail kesalahannya masing-masing.

Sementara banyak masalah tertangkap selama validasi awal ini, beberapa interaksi kompleks antara pengaturan mungkin hanya menjadi jelas saat menerapkan konfigurasi. Dalam kasus seperti itu, operasi akan gagal dengan pesan kesalahan informatif, dan perubahan sebagian akan dibatalkan.

Batasan

AWS PCS menerapkan pendekatan daftar izin untuk melindungi keamanan layanan dan stabilitas operasional. Pengaturan yang dapat membahayakan keamanan akun layanan atau mengganggu kemampuan layanan terkelola dibatasi. Namun, kami terus mengevaluasi kebutuhan pelanggan dan dapat menambahkan dukungan untuk pengaturan tambahan berdasarkan umpan balik pelanggan.


Topik

- [Pengaturan Slurm khusus untuk AWS cluster PCS](#)
- [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#)
- [Pengaturan Slurm khusus untuk AWS antrian PCS](#)
- [Memecahkan masalah pengaturan Slurm kustom di PCS AWS](#)

Pengaturan Slurm khusus untuk AWS cluster PCS


Pengaturan Slurm kustom berikut didukung di tingkat cluster:

- [AccountingStorageEnforce](#)

 Important

AWS PCS mendukung subset dari opsi untuk `AccountingStorageEnforce`. Untuk informasi selengkapnya, lihat [Akuntansi slurm di PCS AWS](#).

- [AccountingStorageTRES](#)
- [AccountingStoreFlags](#)
- [DefMemPerCPU](#)
- [Epilog](#)
- [EnforcePartLimits](#)
- [FairShareDampeningFactor](#)
- [HealthCheckInterval](#)
- [HealthCheckNodeState](#)
- [HealthCheckProgram](#)
- [JobRequeue](#)
- [LaunchParameters](#)
- [Licenses](#)
- [MinJobAge](#)

 Note

AWS PCS mendukung nilai minimum 5 detik untuk `MinJobAge`.

- [OverTimeLimit](#)
- [PreemptExemptTime](#)
- [PreemptMode](#)
- [PreemptParameters](#)

- [PreemptType](#)
- [PriorityCalcPeriod](#)
- [PriorityDecayHalfLife](#)
- [PriorityFavorSmall](#)
- [PriorityFlags](#)
- [PriorityMaxAge](#)
- [PriorityUsageResetPeriod](#)
- [PriorityWeightAge](#)
- [PriorityWeightAssoc](#)
- [PriorityWeightFairshare](#)
- [PriorityWeightJobSize](#)
- [PriorityWeightPartition](#)
- [PriorityWeightQOS](#)
- [PriorityWeightTRES](#)
- [PrivateData](#)
- [Prolog](#)
- [PrologFlags](#)
- [PropagatePrioProcess](#)
- [PropagateResourceLimits](#)
- [PropagateResourceLimitsExcept](#)
- [RequeueExit](#)
- [RequeueExitHold](#)
- [SchedulerParameters](#)
- [SelectTypeParameters](#)
- [SrunPortRange](#)
- [TaskEpilog](#)
- [TaskPluginParam](#)
- [TaskProlog](#)
- [UnkillableStepProgram](#)

- [UnkillableStepTimeout](#)

Pengaturan Slurm khusus untuk grup node komputasi AWS PCS

Pengaturan Slurm kustom berikut didukung pada tingkat grup node komputasi:

- [CpuSpecList](#)
- [Features](#)
- [MemSpecLimit](#)
- [RealMemory](#)
- [Weight](#)

Pengaturan Slurm khusus untuk AWS antrian PCS

Pengaturan Slurm kustom berikut didukung pada tingkat antrian:

- [AllowAccounts](#)
- [AllowQoS](#)
- [Default](#)
- [DefaultTime](#)
- [DenyAccounts](#)
- [DenyQoS](#)
- [ExclusiveUser](#)
- [GraceTime](#)
- [MaxTime](#)
- [OverSubscribe](#)
- [OverTimeLimit](#)
- [PreemptMode](#)
- [PriorityJobFactor](#)
- [PriorityTier](#)
- [QOS](#)

- [TRESBillingWeights](#)

Memecahkan masalah pengaturan Slurm kustom di PCS AWS

Jika Anda mengalami kesalahan saat membuat atau memperbarui sumber daya AWS PCS dengan pengaturan kustom Slurm, Anda dapat menggunakan logging untuk mendiagnosis dan menyelesaikan masalah.

Memecahkan masalah pengaturan kustom Slurm yang tidak kompatibel

Masalah: Anda menerima pesan galat yang mirip dengan berikut ini saat melakukan operasi cluster, grup node komputasi, atau antrian:

```
{OPERATION} failed. The Slurm custom settings of the cluster might be incompatible.  
Check the settings and try again.
```

Kesalahan ini dapat terjadi dengan operasi berikut:

- CreateCluster
- CreateComputeNodeGroup
- UpdateComputeNodeGroup
- CreateQueue
- UpdateQueue

Solusi: Aktifkan pencatatan untuk memahami masalah spesifik dan memecahkan masalah pengaturan yang tidak kompatibel.

Untuk memecahkan masalah pengaturan kustom Slurm yang tidak kompatibel

1. Buat klaster jika belum ada, atau pastikan klaster Anda yang ada dalam keadaan di mana logging dapat diaktifkan.
2. Aktifkan pencatatan untuk klaster Anda. Untuk petunjuk mendetail, lihat [Pencatatan dan pemantauan untuk AWS PCS](#).

Note

Logging dapat diaktifkan setelah cluster dalam pembuatan.

3. Tinjau log untuk mengidentifikasi masalah konfigurasi Slurm tertentu yang menyebabkan ketidakcocokan.
4. Perbaiki pengaturan kustom yang tidak kompatibel berdasarkan informasi log dan coba lagi operasi.

Untuk informasi tentang setelan kustom Slurm yang didukung, lihat:

- [Pengaturan Slurm khusus untuk AWS cluster PCS](#)
- [Pengaturan Slurm khusus untuk grup node komputasi AWS PCS](#)
- [Pengaturan Slurm khusus untuk AWS antrian PCS](#)

Perluas fungsionalitas Slurm pada AWS PCS dengan plugin SPANK

Gunakan plugin SPANK (Slurm Plug-in Architecture for Node and job Kontrol) untuk memperluas dan memodifikasi perilaku Slurm selama peluncuran pekerjaan dan eksekusi pada cluster PCS. AWS Plugin SPANK menyediakan antarmuka generik untuk mencegat dan memodifikasi tahap peluncuran pekerjaan.

Instal plugin SPANK pada AMI node komputasi Anda dan konfigurasi untuk menyesuaikan perilaku klaster Slurm Anda untuk kebutuhan beban kerja Anda. Untuk informasi selengkapnya tentang SPANK, lihat [dokumentasi SPANK di situs web](#) SchedMD.

Daftar Isi

- [Instal plugin SPANK pada PCS AWS](#)
- [Konfigurasi plugin SPANK pada PCS AWS](#)
- [Pertanyaan yang sering diajukan tentang plugin SPANK di PCS AWS](#)

Instal plugin SPANK pada PCS AWS

Ikuti dokumentasi plugin untuk menginstal plugin SPANK di AMI Anda.

Kompilasi plugin SPANK untuk versi Slurm tertentu di cluster Anda. Pemasang Slurm yang disediakan oleh AWS PCS menyimpan Slurm in. `/opt/aws/pcs/scheduler/slurm-version`. Saat Anda mengkompilasi plugin, tentukan versi Slurm.

Contoh berikut menunjukkan cara menentukan versi Slurm untuk beberapa plugin:

```
export CFLAGS="-I/opt/aws/pcs/scheduler/slurm-version/include"
```

Jika Anda memiliki beberapa versi Slurm di AMI, kompilasi plugin untuk setiap versi. Simpan plugin yang dikompilasi dalam folder berversi.

Contoh berikut menunjukkan cara menentukan folder tujuan untuk beberapa plugin:

```
export DESTDIR="your-preferred-versioned-path"
```

Important

Plugin mungkin memerlukan variabel yang berbeda. Lihat dokumentasi resmi untuk plugin yang Anda instal.

Konfigurasi plugin SPANK pada PCS AWS

Secara default, simpan file konfigurasi di `/etc/aws/pcs/scheduler/slurm-version/plugstack.conf.d/`.

Untuk menyimpan konfigurasi SPANK Anda di lokasi yang berbeda, tambahkan lokasi Anda ke file konfigurasi di direktori default.

Contoh berikut menunjukkan cara menyertakan file konfigurasi dari direktori lain:

```
# content of /etc/aws/pcs/scheduler/slurm-version/any-filename.conf
include path-to-your-configuration-folder/*.conf
include path-to-a-second-configuration-folder/*.conf
```

Simpan setiap konfigurasi dalam file khusus, atau dalam file umum. Anda dapat menggunakan beberapa file konfigurasi.

Contoh berikut menunjukkan contoh file konfigurasi:

```
# content of path-to-your-or-default-config-folder/filename-1.conf
required path-to-plugin-1 arguments
```

```
optional path-to-plugin-2 arguments
```

```
# content of path-to-your-or-default-config-folder/filename-2.conf  
required path-to-plugin-3 arguments
```

Untuk informasi tambahan tentang cara mengonfigurasi plugin Anda, lihat [dokumentasi konfigurasi SPANK](#) di situs web SchedMD.

Important

Atur izin folder untuk mencegah perubahan yang tidak sah pada konfigurasi plugin Anda.

Note

AWS PCS tidak mengelola plugin SPANK Anda. Jika Anda mendapatkan kesalahan yang terkait dengan plugin, periksa log kesalahan pada node komputasi Anda.

Note

Slurm salah mencatat kesalahan yang mirip dengan berikut ini saat memuat konfigurasi SPANK Anda:

```
error: "Include" failed in file /etc/slurm/plugstack.conf line 3
```

Anda dapat mengabaikan kesalahan ini. Itu tidak mempengaruhi cara kerja plugin SPANK.

Pertanyaan yang sering diajukan tentang plugin SPANK di PCS AWS

Bagian ini membahas pertanyaan umum tentang menginstal dan mengkonfigurasi plugin SPANK pada AWS cluster PCS.

Apakah saya perlu menginstal plugin SPANK pada node login dan node komputasi?

Beberapa plugin SPANK tidak memerlukan instalasi pada semua node; tetapi untuk kompatibilitas yang lebih baik, kami sarankan Anda menginstal semua plugin SPANK pada setiap node.

Konfigurasi tambahan apa yang diperlukan untuk penggunaan produksi plugin SPANK?

Di luar instalasi dan konfigurasi dasar yang ditunjukkan dalam contoh, penerapan produksi biasanya memerlukan pengaturan tambahan. Plugin berbasis container seperti Pyxis mungkin mengharuskan Anda untuk mengatur variabel lingkungan untuk Enroot, mengaktifkan PMI (Process Management Interface), dan mengonfigurasi izin untuk runtime container. Lihat dokumentasi plugin khusus untuk persyaratan penerapan produksi yang terperinci.

Bagaimana cara memecahkan masalah plugin SPANK?

AWS PCS tidak mengelola plugin SPANK. Periksa log kesalahan pada node komputasi Anda untuk memecahkan masalah.

Gunakan Plugin Filter CLI Slurm untuk menyesuaikan pengiriman pekerjaan di PCS AWS

AWS PCS mendukung Plugin Filter CLI Slurm untuk menjalankan skrip Lua kustom yang memvalidasi dan memodifikasi parameter pengiriman pekerjaan pada node login dan komputasi. Untuk informasi rinci tentang Plugin Filter CLI, lihat [dokumentasi API Plugin cli_filter](#) di situs web SchedMD.

Persyaratan

Plugin Filter CLI memerlukan Slurm versi 24.11 atau yang lebih baru dan skrip Lua digunakan untuk semua node login dan komputasi.

Important

Untuk versi Slurm 24.11 dan 25.05, CLI Filter Plugins memerlukan instalasi Slurm AWS menggunakan PCS Slurm installer (versi 24.11.6-2+ atau 25.05.4-1+). Untuk informasi lebih lanjut tentang menginstal Slurm, lihat [Langkah 3 - Instal Slurm](#)

Keterbatasan dan pertimbangan keamanan

- Penegakan keamanan - Plugin Filter CLI dapat dengan mudah dilewati oleh pengguna mana pun dan tidak boleh digunakan untuk kebijakan keamanan yang kritis. Pengguna dapat menonaktifkan

Plugin Filter CLI dengan menyediakan konfigurasi khusus yang telah `CLIFilterPlugins` dinonaktifkan saat mengirimkan pekerjaan.

- Hanya implementasi Lua - implementasi skrip Lua didukung. Implementasi C tidak didukung.

Topik

- [Konfigurasi Plugin Filter CLI Slurm pada cluster PCS AWS](#)
- [Gunakan Amazon S3 untuk menyebarkan skrip Plugin Filter CLI di PCS AWS](#)
- [Terjemahkan skrip plugin Slurm Job Submit untuk menggunakan Plugin Filter CLI di PCS AWS](#)
- [Pertanyaan yang sering diajukan tentang Plugin Filter Slurm CLI di PCS AWS](#)
- [Memecahkan masalah Slurm CLI Filter Plugin di PCS AWS](#)

Konfigurasi Plugin Filter CLI Slurm pada cluster PCS AWS

Konfigurasi Plugin Filter CLI saat Anda membuat cluster PCS baru AWS . Anda dapat mengaktifkan atau menonaktifkan Plugin Filter CLI pada cluster yang ada menggunakan API Pembaruan atau konsol tanpa membuat ulang cluster.

Prasyarat

Sebelum Anda mengkonfigurasi Plugin Filter CLI, selesaikan tugas-tugas ini:

- Tulis dan uji skrip Lua yang mengimplementasikan CLI Filter Plugin API
- Beri nama skrip Lua Anda dengan tepat `cli_filter.lua`
- Pilih metode untuk menyebarkan skrip Anda ke semua instance cluster (AMI, S3, atau sistem file)
- Verifikasi Anda menggunakan Slurm versi 24.11 atau yang lebih baru

Aktifkan Plugin Filter CLI pada cluster baru

AWS PCS console

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Di panel navigasi, pilih Kluster.
3. Pilih Buat kluster.
4. Pilih versi Slurm yang valid (versi 24.11 atau yang lebih baru).
5. Di bawah Pengaturan penjadwal, perluas Pengaturan penjadwal tambahan.

6. Tambahkan pengaturan kustom Slurm baru dengan nama Parameter diatur ke CliFilterPlugins dan nilai Parameter diatur ke. `cli_filter/lua`
7. Selesaikan konfigurasi cluster yang tersisa dan pilih Create cluster.

AWS PCS API

Berikan `slurmCustomSettings` konfigurasi dalam panggilan Anda ke tindakan `CreateCluster` API. Atur `parameterName` ke `CliFilterPlugins` dan `parameterValue` ke `cli_filter/lua`. Untuk informasi selengkapnya, lihat [CreateCluster](#) di Referensi AWS PCS API.

Contoh berikut menggunakan AWS CLI untuk memanggil tindakan `CreateCluster` API. Pengaturan khusus `CliFilterPlugins=cli_filter/lua` memungkinkan Plugin Filter CLI.

```
aws pcs create-cluster --cluster-name cluster-name \
--scheduler type=SLURM,version=24.11 \
--size SMALL \
--networking subnetIds=cluster-subnet-id,securityGroupIds=cluster-security-group-id \
--slurm-configuration \
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/
lua"}]'
```

Menyebarkan skrip Plugin Filter CLI

Untuk menyebarkan skrip Plugin Filter CLI ke cluster Anda

1. Pastikan semua yang AMIs digunakan dalam grup node komputasi memiliki Slurm diinstal melalui penginstal AWS PCS Slurm.

Note

Jika Anda menggunakan AWS PCS Sample AMI untuk semua grup node komputasi, lewati langkah ini. Slurm sudah diinstal.

2. Terapkan `cli_filter.lua` skrip Anda ke `/etc/aws/pcs/scheduler/slurm-<version>/cli_filter.lua` semua instance di cluster.

Misalnya, untuk Slurm versi 24.11:

```
/etc/aws/pcs/scheduler/slurm-24.11/cli_filter.lua
```

3. Luncurkan semua login dan hitung node menggunakan siap AMIs Anda.
4. Uji pengajuan pekerjaan untuk memverifikasi Plugin Filter CLI dijalankan dengan benar.

Mengaktifkan atau menonaktifkan Plugin Filter CLI pada cluster yang ada

Anda dapat mengaktifkan atau menonaktifkan Plugin Filter CLI pada cluster yang ada tanpa membangun kembali infrastruktur Anda. Untuk informasi selengkapnya, lihat [Memperbarui cluster di AWS PCS](#).

AWS PCS console

1. Buka konsol AWS PCS di <https://console.aws.amazon.com/pcs/>.
2. Pada panel navigasi, silakan pilih Klaster.
3. Pilih cluster yang akan diperbarui.
4. Pilih Edit tindakan.
5. Pada halaman Edit cluster, di bawah Pengaturan penjadwal tambahkan:
 - Untuk mengaktifkan Plugin Filter CLI: Tambahkan pengaturan kustom Slurm baru dengan nama Parameter diatur ke **CliFilterPlugins** dan nilai Parameter diatur ke `cli_filter/lua`
 - Untuk menonaktifkan Plugin Filter CLI: Hapus pengaturan yang ada. **CliFilterPlugins**
6. Pilih Perbarui klaster untuk mengirimkan perubahan.
7. Pantau status cluster, yang ditampilkan sebagai “Memperbarui” selama proses dan “Aktif” saat pembaruan selesai.

AWS PCS API

Gunakan tindakan `UpdateCluster` API untuk mengaktifkan atau menonaktifkan Plugin Filter CLI. Untuk informasi selengkapnya, lihat [UpdateCluster](#) di Referensi AWS PCS API.

Untuk mengaktifkan Plugin Filter CLI pada cluster yang ada:

```
aws pcs update-cluster --cluster-identifier my-cluster \
```

```
--slurm-configuration \  
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/  
lua"}]'
```

Untuk menonaktifkan Plugin Filter CLI pada cluster yang ada:

```
aws pcs update-cluster --cluster-identifier my-cluster \  
--slurm-configuration \  
'slurmCustomSettings=[]'
```

Hasil yang diharapkan

Setelah Anda menyelesaikan konfigurasi:

- Cluster Anda dibuat dengan Plugin Filter CLI diaktifkan
- Pengajuan Job memicu logika validasi kustom Anda sebelum mencapai pengontrol Slurm
- Pekerjaan yang tidak sesuai ditolak dengan pesan kesalahan khusus Anda
- Pekerjaan yang sesuai berjalan normal melalui penjadwal Slurm

Pemecahan masalah

Skrip Plugin Filter CLI hilang pada node apa pun

Gejala: Pengajuan Job segera gagal dengan kesalahan pemuatan plugin.

Kemungkinan penyebabnya: Skrip tidak diterapkan ke semua instance atau jalur atau nama file yang salah.

Resolusi: Verifikasi skrip ada di jalur yang benar pada semua node login dan komputasi dengan nama file `cli_filter.lua` yang tepat.

Konfigurasi Plugin Filter CLI tidak valid

Gejala: Pembuatan cluster gagal dengan kesalahan validasi.

Kemungkinan penyebab: `CliFilterPlugins` parameter tidak disetel ke `cli_filter/lua` format.

Resolusi: Gunakan nilai parameter yang tepat `cli_filter/lua` di `slurmCustomSettings`.

Gunakan Amazon S3 untuk menyebarkan skrip Plugin Filter CLI di PCS AWS

Gunakan S3 untuk menyebarkan skrip Plugin Filter CLI Anda saat Anda ingin memperbarui logika pengiriman pekerjaan di cluster langsung tanpa membangun kembali. AMIs Pendekatan ini mengunduh skrip dari S3 selama peluncuran instance menggunakan data pengguna.

Prasyarat

Sebelum Anda menerapkan skrip Anda menggunakan S3, selesaikan tugas-tugas ini:

- Buat bucket S3 dengan skrip CLI Filter Plugin Lua Anda
- Konfigurasi profil instans IAM dengan akses baca ke bucket S3
- Siapkan titik akhir S3 VPC Gateway untuk akses langsung tanpa internet
- Siapkan skrip data pengguna untuk diunduh dari S3

Untuk menyebarkan skrip Plugin Filter CLI menggunakan S3

1. Unggah `cli_filter.lua` skrip Anda ke bucket S3 Anda.
2. Konfigurasi profil instans IAM Anda dengan izin baca S3 untuk bucket.
3. Tambahkan kode shell ke data pengguna template peluncuran Anda untuk mengunduh skrip:

```
aws s3 cp s3://my-bucket/cli_filter.lua /etc/aws/pcs/scheduler/slurm-24.11/  
cli_filter.lua  
chmod 644 /etc/aws/pcs/scheduler/slurm-24.11/cli_filter.lua
```

4. Terapkan grup node komputasi dengan templat peluncuran Anda yang diperbarui.
5. Uji pengiriman pekerjaan untuk memverifikasi fungsionalitas skrip.

Hasil yang diharapkan

Setelah Anda menyelesaikan penerapan S3:

- Skrip Plugin Filter CLI secara otomatis diunduh ke semua instance selama peluncuran
- Pembaruan skrip di S3 tercermin pada instance yang baru diluncurkan
- Kebijakan pengajuan Job diberlakukan secara konsisten di seluruh klaster

Pemecahan masalah

Akses S3 ditolak

Gejala: Peluncuran instance gagal atau skrip tidak diunduh.

Kemungkinan penyebabnya: Izin IAM tidak ada atau titik akhir VPC S3.

Resolusi: Verifikasi profil instans IAM memiliki `s3:GetObject` izin dan titik akhir VPC S3 dikonfigurasi.

Terjemahkan skrip plugin Slurm Job Submit untuk menggunakan Plugin Filter CLI di PCS AWS

Terjemahkan skrip Job Submit Plugin Lua Anda yang ada ke Plugin Filter CLI saat Anda bermigrasi dari lingkungan Slurm lainnya. Proses penerjemahan melibatkan memperbarui nama fungsi dan pola akses lapangan untuk bekerja dengan CLI Filter Plugin API.

Prasyarat

Sebelum Anda menerjemahkan skrip Anda, selesaikan tugas-tugas ini:

- Tinjau skrip Job Submit Plugin Lua yang ada
- Memahami perbedaan antara Job Submit dan CLI Filter Plugin APIs
- Akses dokumentasi Plugin Filter CLI Slurm

Untuk menerjemahkan skrip Job Submit Plugin ke Plugin Filter CLI

1. Tinjau fungsi skrip Plugin Job Submit Anda yang ada (`slurm_job_submit`, `slurm_job_modify`).
2. Identifikasi fungsi Plugin Filter CLI yang setara:
 - `slurm_job_submit` menjadi `slurm_cli_pre_submit`
 - Tambahkan `slurm_cli_setup_defaults` untuk pengaturan parameter default
 - Tambahkan `slurm_cli_post_submit` untuk tindakan pasca-pengiriman
3. Terjemahkan logika validasi pekerjaan dari `job_desc` bidang ke akses `options` array:
 - `job_desc.account` menjadi `options["account"]`

- `job_desc.partition` menjadi `options["partition"]`
 - `job_desc.features` menjadi `options["constraint"]`
4. Perbarui panggilan logging dari `slurm.log_user()` ke `slurm.log_error()`.
 5. Uji skrip terjemahan Anda pada cluster pengembangan.
 6. Terapkan ke cluster produksi Anda mengikuti proses penerapan Plugin Filter CLI standar.

Hasil yang diharapkan

Setelah Anda menyelesaikan terjemahan:

- Skrip terjemahan Anda memberikan validasi pengiriman pekerjaan yang setara
- Pengguna melihat pesan kesalahan dan permintaan serupa seperti pada Plugin Job Submit asli Anda
- Kebijakan pengajuan Job dipertahankan selama migrasi ke AWS PCS

Pemecahan masalah

Kesalahan terjemahan skrip

Gejala: Pengajuan Job gagal dengan kesalahan eksekusi Lua.

Kemungkinan penyebabnya: Akses bidang atau panggilan fungsi yang salah dalam skrip yang diterjemahkan.

Resolusi: Tinjau dokumentasi CLI Filter Plugin API dan bandingkan pemetaan bidang antara antarmuka Job Submit dan CLI Filter.

Pertanyaan yang sering diajukan tentang Plugin Filter Slurm CLI di PCS AWS

Tinjau pertanyaan yang sering diajukan tentang Plugin Filter CLI ini.

Apa perbedaan antara Plugin Filter CLI dan Plugin Job Submit?

Plugin Filter CLI menjalankan sisi klien pada node login dan komputasi sebelum pengiriman pekerjaan mencapai pengontrol, sementara Job Submit Plugin menjalankan sisi server pada

pengontrol setelah pengiriman pekerjaan. Plugin Filter CLI dapat dilewati oleh pengguna tetapi tidak menahan kunci pengontrol, sementara Job Submit aman tetapi dapat memengaruhi kinerja cluster selama eksekusi.

Apakah AWS PCS mendukung Plugin Kirim Job Slurm?

Tidak, Job Submit Plugin tidak didukung di AWS PCS. Gunakan Plugin Filter CLI sebagai gantinya untuk validasi dan modifikasi pengiriman pekerjaan.

Dapatkah saya menggunakan Plugin Filter CLI untuk penegakan keamanan?

Tidak, Plugin Filter CLI dapat dilewati oleh pengguna yang ditentukan dan tidak boleh diandalkan untuk penegakan keamanan. Gunakan untuk peningkatan pengalaman pengguna, pengaturan parameter default, dan panduan kebijakan daripada kebijakan kritis keamanan.

Mengapa skrip harus ada di semua node komputasi, bukan hanya node login?

Perintah slurm seperti `srun` dapat dijalankan dalam skrip pekerjaan pada node komputasi, yang juga memicu eksekusi Plugin Filter CLI. Skrip harus tersedia di mana pun perintah Slurm dijalankan.

Bisakah saya memodifikasi skrip Plugin Filter CLI pada cluster langsung?

Ya, jika Anda menggunakan pendekatan penyebaran S3 atau sistem file. Instance baru akan mendapatkan skrip yang diperbarui, tetapi instance yang ada memerlukan skrip diperbarui secara manual atau melalui metode penerapan yang Anda pilih.

Dapatkah saya menggunakan skrip Plugin Filter CLI yang berbeda pada grup node komputasi yang berbeda?

Ya, tapi ini tidak disarankan. Anda dapat memberikan skrip dengan logika yang berbeda ke grup node komputasi yang berbeda, tetapi Anda bertanggung jawab untuk mengelola saling ketergantungan dan mencegah logika yang tumpang tindih. Sebagian besar pelanggan menyediakan satu set logika di seluruh cluster.

Bisakah saya menggunakan Plugin Filter CLI dengan implementasi C alih-alih Lua?

Implementasi C tidak didukung. Hanya implementasi skrip Lua yang didukung di AWS PCS. SchedMD merekomendasikan pelanggan menggunakan Lua over C untuk kemudahan penggunaan saat menerapkan Plugin Filter CLI.

Bisakah saya mengaktifkan atau menonaktifkan Plugin Filter CLI pada cluster yang ada?

Ya, Anda dapat mengaktifkan atau menonaktifkan Plugin Filter CLI pada cluster yang ada menggunakan API Pembaruan tanpa membuat ulang cluster.

Memecahkan masalah Slurm CLI Filter Plugin di PCS AWS

Gunakan informasi pemecahan masalah ini untuk menyelesaikan masalah Plugin Filter CLI yang umum.

Pengajuan Job segera gagal dengan kesalahan pemuatan plugin

Gejala: Pengguna menerima pesan kesalahan tentang Plugin Filter CLI yang hilang atau gagal saat mengirimkan pekerjaan.

Kemungkinan penyebabnya:

- Skrip Plugin Filter CLI hilang dari satu atau lebih node
- Nama file skrip salah (harus persis) `cli_filter.lua`
- Skrip disebar ke jalur direktori yang salah
- Skrip memiliki izin file yang salah

Resolusi:

- Verifikasi skrip ada `/etc/aws/pcs/scheduler/slurm-<version>/cli_filter.lua` di semua node login dan komputasi
- Periksa nama file skrip persis `cli_filter.lua`
- Pastikan skrip memiliki izin yang dapat dibaca (644 atau serupa)
- Uji penerapan skrip pada satu node login sebelum menerapkan ke cluster penuh

Pembuatan cluster gagal dengan kesalahan validasi Plugin Filter CLI

Gejala: Pembuatan cluster gagal dengan kesalahan tentang parameter yang tidak `validCliFilterPlugins`.

Kemungkinan penyebabnya:

- Format nilai parameter salah di `slurmCustomSettings`
- Kesalahan ketik dalam nama atau nilai parameter

Resolusi:

- Gunakan nama parameter yang tepat: `CliFilterPlugins`
- Gunakan nilai parameter yang tepat: `cli_filter/lua`
- Verifikasi sintaks JSON dalam array `slurmCustomSettings`

Skrip Plugin Filter CLI dijalankan tetapi validasi pekerjaan tidak berfungsi seperti yang diharapkan

Gejala: Pekerjaan berhasil dikirim tetapi logika validasi kustom tidak memicu atau menghasilkan hasil yang tidak terduga.

Kemungkinan penyebabnya:

- Kesalahan sintaks skrip Lua
- Pola akses bidang yang salah (menggunakan sintaks Job Submit Plugin alih-alih Plugin Filter CLI)
- Kesalahan logika dalam kondisi validasi

Resolusi:

- Tinjau skrip Lua untuk kesalahan sintaks
- Verifikasi akses lapangan menggunakan `options["field_name"]` format, bukan `job_desc.field_name`
- Tambahkan pernyataan logging ke alur eksekusi skrip debug
- Uji logika skrip dengan kasus validasi sederhana terlebih dahulu

Penerapan skrip S3 gagal

Gejala: Instans diluncurkan tetapi skrip Plugin Filter CLI tidak diunduh dari S3.

Kemungkinan penyebabnya:

- Profil instans IAM tidak memiliki izin baca S3
- Titik akhir VPC S3 tidak dikonfigurasi
- Bucket S3 atau jalur objek yang salah dalam data pengguna

Resolusi:

- Verifikasi profil instans IAM memiliki `s3:GetObject` izin untuk bucket Anda
- Konfigurasi titik akhir S3 VPC Gateway untuk akses langsung
- Periksa nama bucket S3 dan jalur objek di skrip data pengguna
- Tinjau log data pengguna instance untuk kesalahan unduhan S3

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 dari 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 sumber daya AWS PC Anda.

Topik

- [Perlindungan data dalam Layanan Komputasi AWS Paralel](#)
- [Akses AWS Parallel Computing Service menggunakan endpoint antarmuka \(\)AWS PrivateLink](#)
- [Identity and Access Management untuk Layanan Komputasi AWS Paralel](#)
- [Validasi kepatuhan untuk Layanan Komputasi AWS Paralel](#)
- [Ketahanan dalam Layanan Komputasi AWS Paralel](#)
- [Keamanan Infrastruktur dalam Layanan Komputasi AWS Paralel](#)
- [Analisis dan manajemen kerentanan dalam Layanan Komputasi AWS Paralel](#)
- [Pencegahan "confused deputy" lintas layanan](#)
- [Praktik terbaik keamanan untuk Layanan Komputasi AWS Paralel](#)

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. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial 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 autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 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. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi saat diam

Enkripsi diaktifkan secara default untuk data saat istirahat saat Anda membuat cluster AWS Parallel Computing Service (AWS PCS) dengan Konsol Manajemen AWS, AWS CLI, AWS PCS API, atau AWS SDKs. AWS PCS menggunakan kunci KMS yang AWS dimiliki untuk mengenkripsi data saat istirahat. Untuk informasi selengkapnya, lihat [Kunci dan AWS kunci pelanggan](#) di Panduan AWS KMS Pengembang. Anda juga dapat menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat [Kebijakan kunci KMS yang diperlukan untuk digunakan dengan volume EBS terenkripsi di PCS AWS](#).

Rahasia cluster disimpan AWS Secrets Manager dan dienkripsi dengan kunci KMS yang dikelola Secrets Manager. Untuk informasi selengkapnya, lihat [Bekerja dengan rahasia cluster di AWS PCS](#).

Dalam cluster AWS PCS, 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 [StateSaveLocation](#) dalam dokumentasi Slurm. AWS PCS menghapus 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 saat bergerak

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

AWS PCS menggunakan 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 ke VPC EC2 dan oleh karena itu tunduk pada enkripsi VPC 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 PCS menggunakan kunci KMS yang AWS dimiliki untuk mengenkripsi data. Untuk informasi selengkapnya, lihat [Kunci dan AWS kunci pelanggan](#) di Panduan AWS KMS Pengembang. Anda juga dapat menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat [Kebijakan kunci KMS yang diperlukan untuk digunakan dengan volume EBS terenkripsi di PCS AWS](#).

Rahasia cluster disimpan AWS Secrets Manager dan dienkripsi dengan kunci KMS yang dikelola Secrets Manager. Untuk informasi selengkapnya, lihat [Bekerja dengan rahasia cluster di AWS PCS](#).

Privasi lalu lintas antar jaringan

AWS PCS menghitung sumber daya untuk kluster berada dalam 1 VPC di akun pelanggan. Oleh karena itu, semua lalu lintas layanan AWS PCS internal dalam sebuah cluster tetap berada dalam AWS jaringan dan tidak melakukan perjalanan melalui internet. Komunikasi antara pengguna dan node AWS PCS dapat melakukan perjalanan di 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.

Menkripsi lalu lintas API

Untuk mengakses AWS PCS API, klien harus mendukung Transport Layer Security (TLS) 1.2 atau yang lebih baru. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3. Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau 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 menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan principal IAM. Anda juga dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Menkripsi lalu lintas data

Enkripsi data dalam perjalanan diaktifkan dari instans EC2 yang didukung yang mengakses titik akhir penjadwal dan antar ComputeNodeGroup instance dari dalam. AWS Cloud Untuk informasi selengkapnya, lihat [Enkripsi saat bergerak](#).

Kebijakan kunci KMS yang diperlukan untuk digunakan dengan volume EBS terenkripsi di PCS AWS

AWS PCS menggunakan [peran terkait layanan](#) untuk mendelegasikan izin ke yang lain. Layanan AWS Peran terkait layanan AWS PCS telah ditentukan sebelumnya dan mencakup izin yang diperlukan AWS PCS untuk memanggil orang lain Layanan AWS atas nama Anda. Izin yang telah ditentukan juga mencakup akses ke kunci yang dikelola pelanggan Anda Kunci yang dikelola AWS tetapi tidak ke kunci yang dikelola pelanggan Anda.

Topik ini menjelaskan cara menyiapkan kebijakan kunci yang diperlukan untuk meluncurkan instance saat Anda menentukan kunci terkelola pelanggan untuk enkripsi Amazon EBS.

Note

AWS PCS tidak memerlukan otorisasi tambahan untuk menggunakan default Kunci yang dikelola AWS untuk melindungi volume terenkripsi di akun Anda.

Daftar Isi

- [Ikhtisar](#)
- [Konfigurasi kebijakan utama](#)
- [Contoh 1: Bagian kebijakan utama yang memungkinkan akses ke kunci yang dikelola pelanggan](#)
- [Contoh 2: Bagian kebijakan utama yang memungkinkan akses lintas akun ke kunci yang dikelola pelanggan](#)
- [Edit kebijakan utama di AWS KMS konsol](#)

Ikhtisar

Anda dapat menggunakan yang berikut ini AWS KMS keys untuk enkripsi Amazon EBS saat AWS PCS meluncurkan instance:

- [Kunci yang dikelola AWS](#)— Kunci enkripsi di akun Anda yang dibuat, dimiliki, dan dikelola Amazon EBS. Ini adalah kunci enkripsi default untuk akun baru. Amazon EBS menggunakan enkripsi Kunci yang dikelola AWS for kecuali Anda menentukan kunci yang dikelola pelanggan.
- [Kunci terkelola pelanggan](#) — Kunci enkripsi khusus yang Anda buat, miliki, dan kelola. Untuk informasi selengkapnya, lihat [Membuat kunci KMS](#) di Panduan AWS Key Management Service Pengembang.

Note

Kuncinya harus simetris. Amazon EBS tidak mendukung kunci yang dikelola pelanggan asimetris.

Anda mengonfigurasi kunci terkelola pelanggan saat membuat snapshot terenkripsi atau templat peluncuran yang menentukan volume terenkripsi, atau saat Anda memilih untuk mengaktifkan enkripsi secara default.

Konfigurasi kebijakan utama

Kunci KMS Anda harus memiliki kebijakan kunci yang memungkinkan AWS PCS meluncurkan instans dengan volume Amazon EBS yang dienkripsi dengan kunci yang dikelola pelanggan.

Gunakan contoh di halaman ini untuk mengonfigurasi kebijakan kunci agar AWS PCS mengakses kunci terkelola pelanggan Anda. Anda dapat mengubah kebijakan kunci terkelola pelanggan saat Anda membuat kunci atau di lain waktu.

Kebijakan utama harus memiliki pernyataan berikut:

- Pernyataan yang memungkinkan identitas IAM yang ditentukan dalam `Principal` elemen untuk menggunakan kunci yang dikelola pelanggan secara langsung. Ini termasuk izin untuk melakukan `AWS KMS Encrypt`, `Decrypt`, `ReEncrypt*`, `GenerateDataKey*`, dan `DescribeKey` operasi pada kunci.
- Pernyataan yang memungkinkan identitas IAM yang ditentukan dalam `Principal` elemen untuk menggunakan `CreateGrant` operasi untuk menghasilkan hibah yang mendelegasikan subset dari izinnya sendiri untuk Layanan AWS yang terintegrasi dengan atau prinsipal lain. AWS KMS Ini memungkinkan mereka untuk menggunakan kunci untuk membuat sumber daya terenkripsi atas nama Anda.

Jangan mengubah pernyataan yang ada dalam kebijakan saat Anda menambahkan pernyataan kebijakan baru ke kebijakan utama Anda.

Untuk informasi lebih lanjut, lihat:

- [create-key](#) di Command Reference AWS CLI
- [put-key-policy](#) dalam Referensi AWS CLI Perintah
- [Temukan ID kunci dan kunci ARN di Panduan](#) Pengembang AWS Key Management Service
- [Peran terkait layanan untuk PCS AWS](#)
- [Enkripsi Amazon EBS](#) di Panduan Pengguna Amazon EBS
- [AWS Key Management Service](#) di Panduan AWS Key Management Service Pengembang

Contoh 1: Bagian kebijakan utama yang memungkinkan akses ke kunci yang dikelola pelanggan

Tambahkan pernyataan kebijakan berikut ke kebijakan kunci terkelola pelanggan. Ganti contoh ARN dengan ARN dari peran terkait layanan Anda. `AWSServiceRoleForPCS` Kebijakan contoh ini memberikan izin peran (`AWSServiceRoleForPCS`) terkait layanan AWS PCS untuk menggunakan kunci terkelola pelanggan.

```
{
```

```

    "Sid": "Allow service-linked role use of the customer managed key",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
      ]
    },
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }
}

```

```

{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
    ]
  },
  "Action": [
    "kms:CreateGrant"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
}

```

Contoh 2: Bagian kebijakan utama yang memungkinkan akses lintas akun ke kunci yang dikelola pelanggan

Jika Anda membuat kunci terkelola pelanggan di akun yang berbeda dari klaster AWS PCS Anda, Anda harus menggunakan hibah yang dikombinasikan dengan kebijakan kunci untuk mengizinkan akses lintas akun ke kunci tersebut.

Untuk memberikan akses ke kunci

1. Tambahkan pernyataan kebijakan berikut ke kebijakan kunci terkelola pelanggan. Ganti contoh ARN dengan ARN dari akun lain. Ganti `111122223333` dengan ID akun aktual dari tempat Akun AWS Anda ingin membuat cluster AWS PCS. Ini memungkinkan Anda untuk memberikan pengguna IAM atau peran dalam izin akun yang ditentukan untuk membuat hibah untuk kunci menggunakan perintah CLI yang mengikuti. Secara default, pengguna tidak memiliki akses ke kunci.

```
{
  "Sid": "Allow external account 111122223333 use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:root"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

```
{
  "Sid": "Allow attachment of persistent resources in external
account 111122223333",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:root"
    ]
  }
}
```

```

    ]
  },
  "Action": [
    "kms:CreateGrant"
  ],
  "Resource": "*"
}

```

2. Dari akun tempat Anda ingin membuat kluster AWS PCS, buat hibah yang mendelegasikan izin yang relevan ke peran terkait layanan AWS PCS. Nilai `grantee-principal` adalah ARN dari peran terkait layanan. Nilai `key-id` adalah ARN kunci.

Contoh perintah CLI [create-grant](#) berikut memberikan peran terkait layanan yang `AWSServiceRoleForPCS` dinamai dalam izin `111122223333` akun untuk menggunakan kunci terkelola pelanggan di akun. `444455556666`

```

aws kms create-grant \
  --region us-west-2 \
  --key-id arn:aws:kms:us-west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
  --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleForPCS \
  --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey" "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"

```

Note

Pengguna yang membuat permintaan harus memiliki izin untuk menggunakan `kms:CreateGrant` tindakan.

Contoh berikut kebijakan IAM memungkinkan identitas IAM (pengguna atau peran) di akun `111122223333` untuk membuat hibah untuk kunci yang dikelola pelanggan di akun. `444455556666`

JSON

```

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

```

```
{
  "Sid": "AllowCreationOfGrantForTheKMSKeyinExternalAccount444455556666",
  "Effect": "Allow",
  "Action": "kms:CreateGrant",
  "Resource": "arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
}
]
```

Untuk informasi selengkapnya tentang membuat hibah untuk kunci KMS yang berbeda Akun AWS, lihat [Hibah AWS KMS di Panduan AWS Key Management Service](#) Pengembang.

Important

Nama peran terkait layanan yang ditentukan sebagai kepala penerima hibah harus merupakan nama peran yang ada. Setelah membuat hibah, untuk memastikan bahwa hibah memungkinkan AWS PCS untuk menggunakan kunci KMS yang ditentukan, jangan hapus dan buat ulang peran terkait layanan.

Edit kebijakan utama di AWS KMS konsol

Contoh di bagian sebelumnya hanya menunjukkan cara menambahkan pernyataan ke kebijakan kunci, yang merupakan salah satu cara untuk mengubah kebijakan kunci. Cara termudah untuk mengubah kebijakan kunci adalah dengan menggunakan tampilan default AWS KMS konsol untuk kebijakan utama dan menjadikan identitas IAM (pengguna atau peran) sebagai salah satu pengguna utama untuk kebijakan kunci yang sesuai. Untuk informasi selengkapnya, lihat [Menggunakan tampilan Konsol Manajemen AWS default](#) di Panduan AWS Key Management Service Pengembang.

Warning

Pernyataan kebijakan tampilan default konsol menyertakan izin untuk melakukan AWS KMS Revoke operasi pada kunci yang dikelola pelanggan. Jika Anda mencabut hibah yang memberikan Akun AWS akses ke kunci yang dikelola pelanggan di akun Anda, pengguna yang Akun AWS kehilangan akses ke data terenkripsi dan kunci tersebut.

Akses AWS Parallel Computing Service menggunakan endpoint antarmuka ()AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC Anda dan AWS Parallel Computing Service ()AWS PCS. Anda dapat mengakses AWS PCS seolah-olah itu ada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi Direct Connect Instans di VPC Anda 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 Anda menyiapkan titik akhir antarmuka AWS PCS, tinjau [Akses layanan AWS menggunakan titik akhir VPC antarmuka](#) di Panduan.AWS PrivateLink

AWS PCS mendukung panggilan ke semua tindakan API-nya melalui titik akhir antarmuka.

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

Buat titik akhir antarmuka untuk AWS PCS

Anda dapat membuat titik akhir antarmuka untuk AWS PCS menggunakan konsol VPC 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 AWS Region untuk membuat titik akhir di, sepertius-east-1.

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

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah sumber daya IAM 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 VPC Anda, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, pengguna IAM, dan peran IAM).
- 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 titik akhir VPC 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 AWS Region 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. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya PCS. AWS IAM adalah 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 terkait layanan untuk PCS AWS](#)
- [Peran Amazon EC2 Spot untuk PCS AWS](#)
- [Izin minimum untuk AWS PCS](#)
- [Profil instans IAM 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 berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat [Pemecahan Masalah Identitas dan akses Layanan Komputasi AWS Paralel](#))

- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Bagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM](#))
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat [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 sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensial. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensial dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Apa itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensial sementara](#) di Panduan Pengguna IAM.

[Grup IAM](#) menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan [beralih dari pengguna ke peran IAM \(konsol\)](#) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukannya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat [Pilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya.

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

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam Panduan Pengguna AWS Organizations .

- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

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 Pengguna IAM.

Bagaimana Layanan Komputasi AWS Paralel bekerja dengan IAM

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

Fitur IAM yang dapat Anda gunakan dengan Layanan Komputasi AWS Paralel

Fitur IAM	AWS Dukungan PCS
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 (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin principal	Ya

Fitur IAM	AWS Dukungan PCS
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS PC dan AWS layanan lainnya bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk PCS AWS

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. 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 [Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkannya atau ditolakannya tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk PCS AWS

Untuk melihat contoh kebijakan berbasis identitas AWS PCS, lihat [Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS](#)

Kebijakan berbasis sumber daya dalam PCS AWS

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM 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 principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk AWS PCS

Mendukung tindakan kebijakan: Ya

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

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan AWS PCS, lihat [Tindakan yang Ditentukan oleh Layanan Komputasi AWS Paralel](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan di 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"  
]
```

Sumber daya kebijakan untuk AWS PCS

Mendukung sumber daya kebijakan: Ya

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

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya AWS PCS dan jenisnya ARNs, 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 berbasis identitas AWS PCS, lihat. [Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS](#)

Kunci kondisi kebijakan untuk AWS PCS

Mendukung kunci kondisi kebijakan khusus layanan: Yes

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

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. 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. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi AWS PCS, lihat Kunci 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 berbasis identitas AWS PCS, lihat. [Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS](#)

ACLs dalam AWS PCS

Mendukung ACLs: Tidak

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

ABAC dengan AWS PCS

Mendukung ABAC (tanda dalam kebijakan): Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda 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 selengkapnya tentang ABAC, lihat [Tentukan izin dengan otorisasi ABAC](#) dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) dalam Panduan Pengguna IAM.

Menggunakan kredensial sementara dengan PCS AWS

Mendukung kredensial sementara: Ya

Kredensial sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#) dan [Layanan AWS yang berfungsi dengan IAM](#) dalam Panduan Pengguna IAM.

Izin utama lintas layanan untuk PCS AWS

Mendukung sesi akses terusan (FAS): Ya

Sesi akses teruskan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses teruskan](#).

Peran layanan untuk AWS PCS

Mendukung peran layanan: Tidak

Peran layanan adalah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas AWS PCS. Edit peran layanan hanya ketika AWS PCS memberikan panduan untuk melakukannya.

Peran terkait layanan untuk PCS AWS

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. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan AWS PCS, lihat [Peran terkait layanan untuk PCS AWS](#)

Contoh kebijakan berbasis identitas untuk Layanan Komputasi Paralel AWS

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS PCS. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM \(konsol\) 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 sumber daya AWS PCS 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 yang dikelola AWS](#) atau [Kebijakan yang dikelola AWS untuk fungsi tugas](#) dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan 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 cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.

- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access 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 dengan IAM Access Analyzer](#) dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Amankan akses API dengan MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

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 sumber daya AWS PCS 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 sesuai dengan operasi API yang coba mereka lakukan.

Untuk informasi selengkapnya tentang izin minimum yang diperlukan untuk menggunakan konsol AWS PCS, lihat [Izin minimum untuk AWS PCS](#).

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ViewOwnUserInfo",
    "Effect": "Allow",
    "Action": [
      "iam:GetUserPolicy",
      "iam:ListGroupsWithUser",
      "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 operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat [Kebijakan terkelola AWS](#) dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AWSPCSCCompute NodePolicy

Anda dapat melampirkan AWSPCSCCompute NodePolicy ke entitas IAM Anda. Anda dapat melampirkan kebijakan ini ke peran IAM node komputasi AWS PCS yang Anda tentukan untuk mengizinkan node yang menggunakan peran tersebut untuk terhubung ke klaster AWS PCS.

AWS PCS melampirkan kebijakan ini ke peran grup node komputasi saat Anda menggunakan konsol untuk membuat grup node komputasi.

Detail izin

Kebijakan ini mencakup izin berikut.

- `pcs:RegisterComputeNodeGroupInstance`— Iizinkan node komputasi AWS PCS (instans EC2) untuk mendaftar dengan cluster AWS PCS.

Untuk melihat izin kebijakan ini, lihat [AWSPCSCComputeNodePolicy](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSPCSService RolePolicy

Anda tidak dapat melampirkan AWSPCSService RolePolicy ke entitas IAM Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan AWS PCS melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Peran terkait layanan untuk PCS AWS](#).

Detail izin

Kebijakan ini mencakup izin berikut.

- `ec2`— Memungkinkan AWS PCS untuk membuat dan mengelola sumber daya Amazon EC2.
- `iam`— Memungkinkan AWS PCS untuk membuat peran terkait layanan untuk armada Amazon EC2 dan meneruskan peran tersebut ke Amazon EC2.
- `cloudwatch`— Memungkinkan AWS PCS mempublikasikan metrik layanan ke Amazon CloudWatch.
- `secretsmanager`— Memungkinkan AWS PCS mengelola rahasia untuk sumber daya cluster AWS PCS.

Untuk melihat izin kebijakan ini, lihat [AWSPCSServiceRolePolicy](#) di Referensi Kebijakan AWS Terkelola.

AWS Pembaruan PCS ke kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk AWS PCS sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen AWS PCS.

Ubah	Deskripsi	Tanggal
AWSPCSServiceRolePolicy — Pembaruan ke kebijakan yang sudah ada	AWS PCS menambahkan izin baru untuk mendukung Blok Kapasitas untuk kapasitas komputasi yang dapat diprediksi. Menambahkan <code>ec2:DescribeCapacityReserva</code>	September 11, 2025

Ubah	Deskripsi	Tanggal
	tions izin untuk mengaktifkan AWS PCS menemukan dan menggunakan reservasi Blok Kapasitas untuk grup node komputasi.	
AWSPCSComputeNodePolicy – Kebijakan baru	AWS PCS menambahkan kebijakan baru untuk memberikan izin ke node komputasi AWS PCS untuk terhubung ke kluster AWS PCS. AWS PCS melampirkan kebijakan ini ke peran IAM saat Anda membuat grup node komputasi di konsol PCS. AWS	Juni 23, 2025
Memperbarui JSON dalam dokumen ini	Mengoreksi JSON dalam dokumen ini untuk disertakan. "arn:aws:ec2:*:*:spot-instances-request/*"	September 5, 2024
AWS PCS mulai melacak perubahan	AWS PCS mulai melacak perubahan untuk kebijakan AWS terkelolanya.	Agustus 28, 2024

Peran terkait layanan untuk PCS AWS

AWS Layanan Komputasi Paralel menggunakan peran [terkait layanan AWS Identity and Access Management](#) (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke PCS. AWS 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 PCS mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya AWS PCS yang dapat mengambil perannya. Izin-izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

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

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, silakan lihat [layanan AWS yang bisa digunakan dengan IAM](#) dan carilah layanan yang memiliki opsi Ya di kolom Peran terkait layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran terkait layanan untuk PCS AWS

AWS PCS menggunakan peran terkait layanan bernama `AWSServiceRoleForPCS` — Memberikan izin kepada AWS PCS untuk mengelola sumber daya Amazon EC2.

Peran terkait layanan `AWSService RoleFor PCS` mempercayai layanan berikut untuk mengambil peran:

- `pcs.amazonaws.com`

Kebijakan izin peran bernama [AWSPCSServiceRolePolicy](#) memungkinkan AWS PCS 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](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk PCS AWS

Anda tidak perlu membuat peran terkait layanan secara manual. AWS PCS membuat peran terkait layanan untuk Anda saat Anda membuat klaster.

Mengedit peran terkait layanan untuk PCS AWS

AWS PCS tidak memungkinkan Anda untuk mengedit peran terkait layanan `AWSService RoleFor PCS`. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena

berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk PCS AWS

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, sebaiknya hapus 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 layanan AWS PCS 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 sumber daya AWS PCS yang digunakan oleh AWSService RoleFor PCS

Anda harus menghapus semua cluster Anda untuk menghapus peran terkait layanan AWSService RoleFor PCS. Untuk informasi selengkapnya, lihat [Menghapus klaster](#).

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran terkait layanan AWSService RoleFor PCS. Untuk informasi selengkapnya, lihat [Menghapus peran terkait layanan](#) dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk AWS peran terkait layanan PCS

AWS PCS mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat [AWS Wilayah dan titik akhir](#).

Peran Amazon EC2 Spot untuk PCS AWS

Jika Anda ingin membuat grup node komputasi AWS PCS yang menggunakan Spot sebagai opsi pembeliannya, Anda juga harus memiliki peran terkait layanan AWSServiceRoleForEC2Spot 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 peran AWSServiceRoleForEC2Spot IAM.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole operation: Service role name AWSServiceRoleForEC2Spot has been taken in this account, please try a different suffix.
```

Izin minimum untuk AWS PCS

Bagian ini menjelaskan izin IAM minimum yang diperlukan untuk identitas IAM (pengguna, grup, atau peran) untuk menggunakan layanan.

Daftar Isi

- [Izin minimum untuk menggunakan tindakan API](#)
- [Izin minimum untuk menggunakan tag](#)
- [Izin minimum untuk mendukung log](#)
- [Izin minimum untuk menggunakan Blok Kapasitas](#)
- [Izin minimum untuk administrator layanan](#)

Izin minimum untuk menggunakan tindakan API

Tindakan API	Izin minimum	Izin tambahan untuk konsol
CreateCluster	<pre>ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups,</pre>	

Tindakan API	Izin minimum	Izin tambahan untuk konsol
	ec2:GetSecurityGroupsForVpc, iam:CreateServiceLinkedRole, secretsmanager:CreateSecret, secretsmanager:TagResource, secretsmanager:RotateSecret, pcs:CreateCluster	
ListClusters	pcs:ListClusters	
GetCluster	pcs:GetCluster	ec2:DescribeSubnets
DeleteCluster	pcs>DeleteCluster	

Tindakan API	Izin minimum	Izin tambahan untuk konsol
CreateComputeNodeGroup	ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:DescribeInstanceTypeOfferings, 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

Tindakan API	Izin minimum	Izin tambahan untuk konsol
UpdateComputeNodeGroup	<pre>ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:DescribeInstanceTypeOfferings, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags, iam:PassRole, iam:GetInstanceProfile, pcs:UpdateComputeNodeGroup</pre>	<pre>pcs:GetComputeNodeGroup, iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster</pre>
DeleteComputeNodeGroup	<pre>pcs>DeleteComputeNodeGroup</pre>	
CreateQueue	<pre>pcs>CreateQueue</pre>	<pre>pcs:ListComputeNodeGroups, pcs:GetCluster</pre>
ListQueues	<pre>pcs:ListQueues</pre>	<pre>pcs:GetCluster</pre>
GetQueue	<pre>pcs:GetQueue</pre>	
UpdateQueue	<pre>pcs:UpdateQueue</pre>	<pre>pcs:ListComputeNodeGroups, pcs:GetQueue</pre>

Tindakan API	Izin minimum	Izin tambahan untuk konsol
DeleteQueue	<code>pcs:DeleteQueue</code>	

Izin minimum untuk menggunakan tag

Izin berikut diperlukan untuk menggunakan tag dengan sumber daya Anda di AWS PCS.

```
pcs:ListTagsForResource,
pcs:TagResource,
pcs:UntagResource
```

Izin minimum untuk mendukung log

AWS PCS mengirimkan data log ke Amazon CloudWatch Logs (CloudWatch Log). Anda harus memastikan bahwa identitas 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 Amazon CloudWatch Logs.

Izin minimum untuk menggunakan Blok Kapasitas

Blok Kapasitas Amazon EC2 untuk ML adalah opsi pembelian Amazon EC2 yang memungkinkan Anda membayar di muka untuk memesan instans komputasi akselerasi berbasis GPU dalam rentang tanggal dan waktu tertentu untuk mendukung beban kerja berdurasi pendek. Untuk informasi selengkapnya, lihat [Menggunakan Blok Kapasitas Amazon EC2 untuk ML dengan PCS AWS](#).

Anda memilih untuk menggunakan Blok Kapasitas saat membuat atau memperbarui grup node komputasi. Identitas IAM yang Anda gunakan untuk membuat atau memperbarui grup node komputasi harus memiliki izin berikut:

```
ec2:DescribeCapacityReservations
```

Izin minimum untuk administrator layanan

Kebijakan IAM berikut menentukan izin minimum yang diperlukan untuk identitas IAM (pengguna, grup, atau peran) untuk mengonfigurasi dan mengelola layanan PCS. AWS

Note

Pengguna yang tidak mengonfigurasi dan mengelola layanan tidak memerlukan izin ini. Pengguna yang hanya menjalankan pekerjaan menggunakan shell aman (SSH) untuk terhubung ke cluster. AWS Identity and Access Management (IAM) tidak menangani otentikasi atau otorisasi untuk SSH.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PCSAccess",
      "Effect": "Allow",
      "Action": [
        "pcs:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2Access",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeImages",
        "ec2:GetSecurityGroupsForVpc",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:RunInstances",
        "ec2:CreateFleet",
        "ec2:CreateTags",
        "ec2:DescribeCapacityReservations"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  },
  {
    "Sid": "IamInstanceProfile",
    "Effect": "Allow",
    "Action": [
      "iam:GetInstanceProfile"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IamPassRole",
    "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": "SLRAccess",
    "Effect": "Allow",
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": [
      "arn:aws:iam::*:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleFor*",
      "arn:aws:iam::*:role/aws-service-role/spot.amazonaws.com/AWSServiceRoleFor*"
    ],
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": [

```

```
        "pcs.amazonaws.com",
        "spot.amazonaws.com"
    ]
}
},
{
    "Sid": "AccessKMSKey",
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
{
    "Sid": "SecretManagementAccess",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:CreateSecret",
        "secretsmanager:TagResource",
        "secretsmanager:UpdateSecret",
        "secretsmanager:RotateSecret"
    ],
    "Resource": "*"
},
{
    "Sid": "ServiceLogsDelivery",
    "Effect": "Allow",
    "Action": [
        "pcs:AllowVendedLogDeliveryForResource",
        "logs:PutDeliverySource",
        "logs:PutDeliveryDestination",
        "logs:CreateDelivery"
    ],
    "Resource": "*"
}
]
```

Profil instans IAM untuk Layanan Komputasi AWS Paralel

Aplikasi yang berjalan pada instans EC2 harus menyertakan AWS kredensial dalam permintaan AWS API apa pun yang mereka buat. Kami menyarankan Anda menggunakan peran IAM untuk mengelola kredensial sementara pada instans EC2. Anda dapat menentukan profil instance untuk melakukan ini, dan melampirkannya ke instance Anda. Untuk informasi selengkapnya, lihat [peran IAM untuk Amazon EC2](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Note

Saat Anda menggunakan Konsol Manajemen AWS untuk membuat peran IAM untuk Amazon EC2, konsol akan membuat profil instans secara otomatis dan memberinya nama yang sama dengan peran IAM. Jika Anda menggunakan AWS CLI, tindakan AWS API, atau AWS SDK untuk membuat peran IAM, Anda membuat profil instance sebagai tindakan terpisah. Untuk informasi selengkapnya, lihat [Profil instans](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Anda harus menentukan Nama Sumber Daya Amazon (ARN) dari profil instance saat membuat grup node komputasi. Anda dapat memilih profil instance yang berbeda untuk beberapa atau semua grup node komputasi.

Persyaratan

Peran IAM dari profil instance

Peran IAM yang terkait dengan profil instance harus ada `/aws-pcs/` di jalurnya, atau namanya harus dimulai dengan `AWSPCS`.

Contoh peran IAM ARNs

- `arn:aws:iam::*:role/AWSPCS-example-role-1`
- `arn:aws:iam::*:role/aws-pcs/example-role-2`

Izin

Peran IAM yang terkait dengan profil instans untuk AWS PCS harus menyertakan kebijakan berikut.

JSON

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

Kebijakan tambahan

Pertimbangkan untuk menambahkan kebijakan terkelola ke profil instance. Contoh:

- [AmazonS3 ReadOnlyAccess](#) menyediakan akses hanya-baca ke semua bucket S3.
- [Amazon SSMManaged InstanceCore](#) mengaktifkan fungsionalitas inti layanan AWS Systems Manager, seperti akses jarak jauh langsung dari Amazon Management Console.
- [CloudWatchAgentServerPolicy](#) berisi izin yang diperlukan untuk digunakan AmazonCloudWatchAgent di server.

Anda juga dapat menyertakan kebijakan IAM Anda sendiri yang mendukung kasus penggunaan spesifik Anda.

Buat profil instance untuk AWS PCS

AWS PCS console

Pilih Buat profil dasar saat Anda membuat grup node komputasi agar AWS PCS membuatnya untuk Anda dengan kebijakan minimum yang diperlukan.

Amazon EC2 console

Anda dapat membuat profil instans langsung dari konsol Amazon EC2. Untuk informasi selengkapnya, lihat [Menggunakan profil instans](#) di Panduan AWS Identity and Access Management Pengguna.

Important

Pastikan untuk menggunakan awalan yang diperlukan AWSPCS dalam nama peran IAM.

AWS CLI

Menyiapkan profil instans Dasar menggunakan AWS CLI

Note

Ganti *example-role* dalam contoh berikut dengan nama peran IAM Anda.

1. Buat peran IAM dengan `/aws-pcs/` atribut path atau nama yang dimulai dengan AWSPCS.
 - a. Salin dan tempel konten berikut ke file teks baru bernama `trust_policy.json`.

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

- b. Gunakan 1 dari perintah berikut untuk membuat peran IAM.

```
aws iam create-role --path /aws-pcs/ --role-name example-role --assume-role-policy-document file://trust_policy.json
```

atau

```
aws iam create-role --role-name AWSPCS-example-role --assume-role-policy-document file://trust_policy.json
```

2. Lampirkan izin.

- a. Salin dan tempel konten berikut ke file teks baru bernama `policy_document.json`.

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

- b. Lampirkan dokumen kebijakan ke peran. Perintah ini melampirkan kebijakan sebagai kebijakan inline.

```
aws iam put-role-policy \
  --role-name example-role \
  --policy-name pcsRegisterInstancePolicy \
  --policy-document file://policy_document.json
```

3. Buat profil instance. Ganti *example-profile* dengan nama profil instans Anda.

```
aws iam create-instance-profile --instance-profile-name example-profile
```

4. Kaitkan peran IAM dengan profil instance.

```
aws iam add-role-to-instance-profile \  
  --instance-profile-name example-profile \  
  --role-name example-role
```

Temukan profil instance yang digunakan dengan AWS PCS

1. Jika Anda tidak tahu nama persis peran IAM Anda untuk AWS PCS, gunakan AWS CLI perintah berikut untuk mencantumkan peran IAM yang memenuhi persyaratan nama AWS PCS.

```
aws iam list-roles --query "Roles[?starts_with(RoleName, 'AWSPCS') ||  
  contains(Path, '/aws-pcs/)].[RoleName]" --output text
```

2. Gunakan AWS CLI perintah berikut untuk daftar profil instance yang terkait dengan peran IAM tertentu. Ganti *role-name* dengan nama peran IAM yang memenuhi persyaratan nama AWS PCS.

```
aws iam list-instance-profiles-for-role --role-name role-name
```

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 dan IAM.

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 sumber daya AWS PCS 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 pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `pcs:GetWidget` rekaan.

```
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 bahwa Anda tidak berwenang untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke 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 pengguna IAM 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 sumber daya AWS PCS 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 mengetahui apakah AWS PCS mendukung fitur-fitur ini, lihat [Bagaimana 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 pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- 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 Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam 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 di 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. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat [Dokumentasi AWS Keamanan](#).

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 panggilan API yang AWS dipublikasikan untuk mengakses AWS PCS melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

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 membuat Antarmuka Jaringan Elastis (ENI) lintas akun di VPC Anda. Pengontrol Slurm menggunakan ENI untuk mengelola dan berkomunikasi dengan node komputasi di berbagai tempat Akun AWS, menjaga keamanan dan isolasi sumber daya sambil memfasilitasi HPC dan operasi yang efisien. 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](#).

Note

Pengontrol slurm tidak tersedia saat kami memperbaruinya. Menjalankan pekerjaan tidak terpengaruh. Pekerjaan yang dikirimkan saat pengontrol kluster tidak tersedia ditahan hingga pengontrol tersedia.

Anda bertanggung jawab atas keamanan infrastruktur yang mendasari di Akun AWS:

- Pertahankan kode Anda, termasuk pembaruan dan patch keamanan.
- Menambal dan memperbarui sistem operasi di Amazon Machine Image (AMI) untuk grup node komputasi Anda dan perbarui grup node komputasi Anda untuk menggunakan AMI yang diperbarui.
- Perbarui penjadwal untuk menyimpannya dalam versi yang didukung. Perbarui AMI untuk grup node komputasi Anda dan perbarui grup node komputasi Anda untuk menggunakan AMI yang diperbarui.
- Mengotentikasi dan mengenkripsi komunikasi antara klien pengguna dan node yang mereka sambungkan.

Untuk informasi selengkapnya tentang memperbarui AMI untuk grup node komputasi Anda, lihat [Gambar Mesin Amazon \(AMIs\) untuk AWS PCS](#).

Pencegahan "confused deputy" lintas layanan

Masalah deputy yang membingungkan adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. 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 principal 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) layanan lain ke sumber daya. Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan 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 "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks global `aws:SourceArn` dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:servicename:*:123456789012:*`.

Jika nilai `aws:SourceArn` tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global tersebut untuk membatasi izin.

Nilai `aws:SourceArn` harus berupa ARN cluster.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan `aws:SourceArn` dan kunci konteks kondisi `aws:SourceAccount` global di 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"
  }
}
}
}

```

Peran IAM untuk instans Amazon EC2 disediakan sebagai bagian dari grup node komputasi

AWS PCS secara otomatis mengatur kapasitas Amazon EC2 untuk setiap grup node komputasi yang dikonfigurasi dalam sebuah cluster. Saat membuat grup node komputasi, pengguna harus menyediakan profil instans IAM melalui bidang `iamInstanceProfileArn`. Profil instans menentukan izin yang terkait dengan instans EC2 yang disediakan. AWS PCS menerima peran apa pun yang memiliki AWSPCS awalan nama peran atau `/aws-pcs/` sebagai bagian dari jalur peran. `iam:PassRole` diperlukan pada identitas IAM (pengguna atau peran) yang membuat atau memperbarui grup node komputasi. Ketika pengguna memanggil tindakan `CreateComputeNodeGroup` atau `UpdateComputeNodeGroup` API, AWS PCS memeriksa untuk melihat apakah pengguna diizinkan untuk melakukan `iam:PassRole` tindakan.

Contoh kebijakan berikut memberikan izin untuk hanya meneruskan peran IAM 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 AWS Parallel Computing Service (AWS PCS). Untuk mempelajari lebih lanjut tentang praktik terbaik keamanan di AWS, lihat [Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan](#).

Keamanan terkait AMI

- Jangan gunakan sampel AWS PCS AMIs untuk beban kerja produksi. Sampel AMIs tidak didukung dan hanya ditujukan untuk pengujian.
- Secara teratur memperbarui sistem operasi dan perangkat lunak di AMI untuk grup node komputasi Anda untuk mengurangi kerentanan.
- Hanya gunakan paket AWS PCS resmi yang diautentikasi yang diunduh dari AWS sumber resmi.
- Perbarui paket AWS PCS secara teratur di AMI untuk grup node komputasi dan perbarui node komputasi untuk menggunakan AMI yang diperbarui. Pertimbangkan untuk mengotomatiskan proses ini untuk meminimalkan kerentanan.

Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

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 kehidupan dukungan (EOSL) dihentikan segera. Gunakan tautan di bagian atas halaman panduan pengguna untuk berlangganan umpan RSS dokumentasi AWS PCS untuk menerima pemberitahuan saat versi Slurm mendekati EOSL.

Untuk informasi selengkapnya, lihat [Versi slurm di PCS AWS](#).

- Rotasi rahasia cluster secara teratur untuk menjaga kepatuhan keamanan dan memulihkan potensi kompromi keamanan. Ini diperlukan untuk kepatuhan HIPAA dan FedRAMP.

Untuk informasi selengkapnya, lihat [Memutar rahasia cluster di AWS PCS](#).

Pencatatan dan pemantauan

- 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 kluster AWS PCS Anda di VPC terpisah untuk mengisolasi lingkungan HPC Anda dari lalu lintas jaringan lainnya.
- Gunakan grup keamanan dan daftar kontrol akses jaringan (ACLs) untuk mengontrol lalu lintas masuk dan keluar ke instance dan subnet AWS PCS.
- Gunakan AWS PrivateLink atau titik akhir VPC untuk menjaga lalu lintas jaringan antara kluster Anda dan AWS layanan lain di dalam jaringan. Untuk informasi selengkapnya, lihat [Akses AWS Parallel Computing Service menggunakan endpoint antarmuka \(\)AWS PrivateLink](#).

Pencatatan dan pemantauan untuk AWS PCS

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja AWS PCS dan sumber daya AWS Anda yang lain. AWS menyediakan 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 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 penggunaan CPU atau metrik lain dari instans Amazon EC2 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 instans Amazon EC2, 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 CloudTrail menangkap panggilan API 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, silakan lihat [Panduan Pengguna AWS CloudTrail](#).

Log penyelesaian pekerjaan di AWS PCS

Log penyelesaian pekerjaan memberi Anda detail penting tentang pekerjaan AWS Parallel Computing Service (AWS PCS) Anda saat selesai, tanpa biaya tambahan. Anda dapat menggunakan AWS layanan lain untuk mengakses dan memproses data log Anda, seperti Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3), dan Amazon Data Firehose AWS ; PCS merekam metadata tentang pekerjaan Anda, seperti berikut ini.

- ID dan nama Job
- Informasi pengguna dan grup
- Status pekerjaan (seperti COMPLETED, FAILED, CANCELLED)

- Partisi yang digunakan
- Batas waktu
- Mulai, akhiri, kirim, dan waktu yang memenuhi syarat
- Daftar dan hitungan simpul
- Jumlah prosesor
- Direktori kerja
- Penggunaan sumber daya (CPU, memori)
- Kode keluar
- Detail simpul (nama, contoh IDs, jenis contoh)

Daftar Isi

- [Prasyarat](#)
- [Siapkan log penyelesaian pekerjaan](#)
- [Cara menemukan log penyelesaian pekerjaan](#)
 - [CloudWatch Log](#)
 - [Amazon S3](#)
- [Bidang log penyelesaian pekerjaan](#)
- [Contoh log penyelesaian pekerjaan](#)

Prasyarat

Prinsipal IAM yang mengelola cluster AWS PCS harus memungkinkan `pcs:AllowVendedLogDeliveryForResource` tindakan.

Contoh berikut kebijakan IAM memberikan izin yang diperlukan.

JSON

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

```

    "Effect": "Allow",
    "Action": ["pcs:AllowVendedLogDeliveryForResource"],
    "Resource": [
        "arn:aws:pcs:*::cluster/*"
    ]
  }
]
}

```

Siapkan log penyelesaian pekerjaan

Anda dapat mengatur log penyelesaian pekerjaan untuk kluster AWS PCS Anda dengan Konsol Manajemen AWS atau AWS CLI.

Konsol Manajemen AWS

Untuk menyiapkan log penyelesaian pekerjaan dengan konsol

1. Buka [konsol AWS PCS](#).
2. Pada panel navigasi, silakan pilih Kluster.
3. Pilih cluster tempat Anda ingin menambahkan log penyelesaian pekerjaan.
4. Pada halaman detail cluster, pilih tab Log.
5. Di bawah Log Penyelesaian Pekerjaan, pilih Tambah untuk menambahkan hingga 3 tujuan pengiriman log dari antara CloudWatch Log, Amazon S3, dan Firehose.
6. Pilih Perbarui pengiriman log.

AWS CLI

Untuk mengatur log penyelesaian pekerjaan dengan AWS CLI

1. Buat tujuan pengiriman log:

```

aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn

```

Ganti:

- *region*— AWS Region Tempat Anda ingin membuat tujuan, seperti `us-east-1`
- *pcs-logs-destination*— Nama untuk tujuan
- *resource-arn*— Nama Sumber Daya Amazon (ARN) dari grup CloudWatch log Log, bucket S3, atau aliran pengiriman Firehose.

Untuk informasi selengkapnya, lihat [PutDeliveryDestination](#) di Referensi API Amazon CloudWatch Logs.

2. Tetapkan cluster PCS sebagai sumber pengiriman log:

```
aws logs put-delivery-source --region region \  
  --name cluster-logs-source-name \  
  --resource-arn cluster-arn \  
  --log-type PCS_JOBCOMP_LOGS
```

Ganti:

- *region*— AWS Region Cluster Anda, seperti `us-east-1`
- *cluster-logs-source-name*— Nama untuk sumbernya
- *cluster-arn*— ARN dari cluster PCS Anda AWS

Untuk informasi selengkapnya, lihat [PutDeliverySource](#) di Referensi API Amazon CloudWatch Logs.

3. Hubungkan sumber pengiriman ke tujuan pengiriman:

```
aws logs create-delivery --region region \  
  --delivery-source-name cluster-logs-source \  
  --delivery-destination-arn destination-arn
```

Ganti:

- *region*— yang AWS Region, seperti `us-east-1`
- *cluster-logs-source*— Nama sumber pengiriman Anda
- *destination-arn*— ARN tujuan pengiriman Anda

Untuk informasi selengkapnya, lihat [CreateDelivery](#) di Referensi API Amazon CloudWatch Logs.

Cara menemukan log penyelesaian pekerjaan

Anda dapat mengonfigurasi tujuan log di CloudWatch Log dan Amazon S3. AWS PCS menggunakan nama jalur terstruktur berikut dan nama file.

CloudWatch Log

AWS PCS menggunakan format nama berikut untuk aliran CloudWatch Log:

```
AWSLogs/PCS/cluster-id/jobcomp.log
```

Misalnya: AWSLogs/PCS/pcs_abc123de45/jobcomp.log

Amazon S3

AWS PCS menggunakan format nama berikut untuk jalur S3:

```
AWSLogs/account-id/PCS/region/cluster-id/jobcomp/year/month/day/hour/
```

Misalnya: AWSLogs/111122223333/PCS/us-east-1/pcs_abc123de45/jobcomp/2025/06/19/11/

AWS PCS menggunakan format nama berikut untuk file log:

```
PCS_jobcomp_year-month-day-hour_cluster-id_random-id.log.gz
```

Misalnya: PCS_jobcomp_2025-06-19-11_pcs_abc123de45_04be080b.log.gz

Bidang log penyelesaian pekerjaan

AWS PCS menulis data log penyelesaian pekerjaan sebagai objek JSON. Wadah JSON jobcomp menyimpan detail pekerjaan. Tabel berikut menjelaskan bidang di dalam jobcomp wadah. Beberapa bidang hanya ada dalam keadaan tertentu, seperti untuk pekerjaan array atau pekerjaan heterogen.

Bidang log penyelesaian pekerjaan

Nama	Nilai contoh	Diperlukan	Catatan
job_id	11	Ya	Selalu hadir dengan nilai
user	"root"	Ya	Selalu hadir dengan nilai
user_id	0	Ya	Selalu hadir dengan nilai
group	"root"	Ya	Selalu hadir dengan nilai
group_id	0	Ya	Selalu hadir dengan nilai
name	"wrap"	Ya	Selalu hadir dengan nilai
job_state	"COMPLETED"	Ya	Selalu hadir dengan nilai
partition	"Hydra-Mp iQueue-ab cdef01-7"	Ya	Selalu hadir dengan nilai
time_limit	"UNLIMITED"	Ya	Selalu hadir, tapi mungkin "UNLIMITED"
start_time	"2025-06- 19T10:58: 57"	Ya	Selalu hadir, tapi mungkin "Unknown"
end_time	"2025-06- 19T10:58: 57"	Ya	Selalu hadir, tapi mungkin "Unknown"
node_list	"Hydra-Mp iNG-abcde f01-2345- 1"	Ya	Selalu hadir dengan nilai
node_cnt	1	Ya	Selalu hadir dengan nilai
proc_cnt	1	Ya	Selalu hadir dengan nilai

Nama	Nilai contoh	Diperlukan	Catatan
<code>work_dir</code>	<code>"/root"</code>	Ya	Selalu hadir, tapi mungkin "Unknown"
<code>reservation_name</code>	<code>"weekly_maintenance"</code>	Ya	Selalu hadir, tetapi mungkin string kosong ""
<code>tres.cpu</code>	1	Ya	Selalu hadir dengan nilai
<code>tres.mem.val</code>	600	Ya	Selalu hadir dengan nilai
<code>tres.mem.unit</code>	"M"	Ya	Bisa "M" atau "bb"
<code>tres.node</code>	1	Ya	Selalu hadir dengan nilai
<code>tres.billing</code>	1	Ya	Selalu hadir dengan nilai
<code>account</code>	<code>"finance"</code>	Ya	Selalu hadir, tetapi mungkin string kosong ""
<code>qos</code>	<code>"normal"</code>	Ya	Selalu hadir, tetapi mungkin string kosong ""
<code>wc_key</code>	<code>"project_1"</code>	Ya	Selalu hadir, tetapi mungkin string kosong ""
<code>cluster</code>	<code>"unknown"</code>	Ya	Selalu hadir, tapi mungkin "unknown"
<code>submit_time</code>	<code>"2025-06-19T10:55:46"</code>	Ya	Selalu hadir, tapi mungkin "Unknown"

Nama	Nilai contoh	Diperlukan	Catatan
eligible_time	"2025-06-19T10:55:46"	Ya	Selalu hadir, tapi mungkin "Unknown"
array_job_id	12	tidak	Hanya hadir jika pekerjaan itu adalah pekerjaan array
array_task_id	1	tidak	Hanya hadir jika pekerjaan itu adalah pekerjaan array
het_job_id	10	tidak	Hanya hadir jika pekerjaan itu adalah pekerjaan yang heterogen
het_job_offset	0	tidak	Hanya hadir jika pekerjaan itu adalah pekerjaan yang heterogen
derived_exit_code_status	0	Ya	Selalu hadir dengan nilai
derived_exit_code_signal	0	Ya	Selalu hadir dengan nilai
exit_code_status	0	Ya	Selalu hadir dengan nilai
exit_code_signal	0	Ya	Selalu hadir dengan nilai
node_details[0].name	"Hydra-Mp iNG-abcde f01-2345- 1"	tidak	Selalu hadir, tapi node_details mungkin "[]"

Nama	Nilai contoh	Diperlukan	Catatan
node_details[0].instance_id	"i-0abcde f01234567 a"	tidak	Selalu hadir, tapi node_details mungkin "[]"
node_details[0].instance_type	"t4g.micro"	tidak	Selalu hadir, tapi node_details mungkin "[]"

Contoh log penyelesaian pekerjaan

Contoh berikut menunjukkan log penyelesaian pekerjaan untuk berbagai jenis dan status pekerjaan:

```
{ "jobcomp": { "job_id": 1, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:32:57", "end_time": "2025-06-19T16:33:03", "node_list": "Hydra-MpiNG-abcdef01-2345-[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name": "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2, "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T16:29:40", "eligible_time": "2025-06-19T16:29:41", "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name": "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 2, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T16:33:13", "end_time": "2025-06-19T16:33:14", "node_list": "Hydra-MpiNG-abcdef01-2345-[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/usr/bin", "reservation_name": "", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2, "billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T16:33:13", "eligible_time": "2025-06-19T16:33:13", "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id": "i-0abc123def45678", "instance_type": "t4g.micro" }, { "name": "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901", "instance_type": "t4g.micro" } ] } }
```

```

{ "jobcomp": { "job_id": 3, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T22:58:57", "end_time":
"2025-06-19T22:58:57", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T22:55:46",
"eligible_time": "2025-06-19T22:55:46", "derived_exit_code_status": 0,
"derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal":
0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id":
"i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 4, "user": "root", "user_id": 0, "group": "root",
"group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-
MpiQueue-abcdef01-7", "time_limit": "525600", "start_time": "2025-06-19T23:04:27",
"end_time": "2025-06-19T23:04:27", "node_list": "Hydra-MpiNG-abcdef01-2345-
[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir": "/root", "reservation_name":
"", "tres": { "cpu": 2, "mem": { "val": 1944, "unit": "M" }, "node": 2,
"billing": 2 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
"submit_time": "2025-06-19T23:01:38", "eligible_time": "2025-06-19T23:01:38",
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
"instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" }, { "name":
"Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def345abc67890", "instance_type":
"t4g.micro" } ] } }
{ "jobcomp": { "job_id": 5, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "FAILED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:00", "end_time":
"2025-06-19T23:09:00", "node_list": "(null)", "node_cnt": 0, "proc_cnt": 0,
"work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val":
1, "unit": "G" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key":
"", "cluster": "unknown", "submit_time": "2025-06-19T23:09:00", "eligible_time":
"2025-06-19T23:09:00", "derived_exit_code_status": 0, "derived_exit_code_signal": 0,
"exit_code_status": 0, "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 6, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-
abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:36",
"end_time": "2025-06-19T23:09:36", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
{ "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
"", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:35",
"eligible_time": "2025-06-19T23:09:36", "het_job_id": 6, "het_job_offset": 0,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
"exit_code_signal": 1, "node_details": [] } }

```

```
{ "jobcomp": { "job_id": 7, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "CANCELLED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:10:03", "end_time": "2025-06-19T23:10:03", "node_list": "(null)", "node_cnt": 0, "proc_cnt": 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:10:03", "eligible_time": "2025-06-19T23:10:03", "het_job_id": 7, "het_job_offset": 0, "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 1, "node_details": [] } }
{ "jobcomp": { "job_id": 8, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time": "2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt": 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23", "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 0, "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 9, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time": "2025-06-19T23:11:24", "node_list": "Hydra-MpiNG-abcdef01-2345-2", "node_cnt": 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23", "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 1, "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2", "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 10, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7", "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time": "2025-06-19T23:12:24", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt": 1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14", "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 0, "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1", "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
```

```
{ "jobcomp": { "job_id": 11, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
"2025-06-19T23:12:24", "node_list": "Hydra-MpiNG-abcdef01-2345-2", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 600, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
"eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 1,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-2",
"instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 13, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:57", "end_time":
"2025-06-19T23:47:58", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
"eligible_time": "2025-06-19T23:43:56" , "array_job_id": 12, "array_task_id": 1,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
"instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
{ "jobcomp": { "job_id": 12, "user": "root", "user_id": 0, "group": "root", "group_id":
0, "name": "wrap", "job_state": "COMPLETED", "partition": "Hydra-MpiQueue-abcdef01-7",
"time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:58", "end_time":
"2025-06-19T23:47:58", "node_list": "Hydra-MpiNG-abcdef01-2345-1", "node_cnt":
1, "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu":
1, "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
"qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
"eligible_time": "2025-06-19T23:43:56" , "array_job_id": 12, "array_task_id": 2,
"derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
0, "exit_code_signal": 0, "node_details": [ { "name": "Hydra-MpiNG-abcdef01-2345-1",
"instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
```

Log penjadwal di AWS PCS

Anda dapat mengonfigurasi AWS PCS untuk mengirim data pencatatan terperinci dari penjadwal kluster ke Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3), dan Amazon Data Firehose. Ini dapat membantu pemantauan dan pemecahan masalah.

Daftar Isi

- [Prasyarat](#)

- [Siapkan log penjadwal](#)
- [Jalur dan nama aliran log penjadwal](#)
- [Contoh catatan log penjadwal](#)

Prasyarat

Prinsipal IAM yang mengelola cluster AWS PCS harus memungkinkan `pcs:AllowVendedLogDeliveryForResource` tindakan.

Contoh berikut kebijakan IAM memberikan izin yang diperlukan.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PcsAllowVendedLogsDelivery",
      "Effect": "Allow",
      "Action": ["pcs:AllowVendedLogDeliveryForResource"],
      "Resource": [
        "arn:aws:pcs:*::cluster/*"
      ]
    }
  ]
}
```

Siapkan log penjadwal

Anda dapat mengatur log penjadwal untuk klaster AWS PCS Anda dengan Konsol Manajemen AWS atau AWS CLI.

Konsol Manajemen AWS

Untuk mengatur log penjadwal dengan konsol

1. Buka [konsol AWS PCS](#).
2. Pada panel navigasi, silakan pilih Klaster.

3. Pilih cluster tempat Anda ingin menambahkan log penjadwal.
4. Pada halaman detail cluster, pilih tab Log.
5. Di bawah Log Penjadwal, pilih Tambahkan untuk menambahkan hingga 3 tujuan pengiriman log dari antara CloudWatch Log, Amazon S3, dan Firehose.
6. Pilih Perbarui pengiriman log.

AWS CLI

Untuk mengatur log penjadwal dengan AWS CLI

1. Buat tujuan pengiriman log:

```
aws logs put-delivery-destination --region region \  
  --name pcs-logs-destination \  
  --delivery-destination-configuration \  
  destinationResourceArn=resource-arn
```

Ganti:

- *region*— AWS Region Tempat Anda ingin membuat tujuan, seperti `us-east-1`
- *pcs-logs-destination*— Nama untuk tujuan
- *resource-arn*— Nama Sumber Daya Amazon (ARN) dari grup CloudWatch log Log, bucket S3, atau aliran pengiriman Firehose.

Untuk informasi selengkapnya, lihat [PutDeliveryDestination](#) di Referensi API Amazon CloudWatch Logs.

2. Tetapkan cluster PCS sebagai sumber pengiriman log:

```
aws logs put-delivery-source --region region \  
  --name cluster-logs-source-name \  
  --resource-arn cluster-arn \  
  --log-type PCS_SCHEDULER_LOGS
```

Ganti:

- *region*— AWS Region Cluster Anda, seperti `us-east-1`
- *cluster-logs-source-name*— Nama untuk sumbernya

- *cluster-arn*— ARN dari cluster PCS Anda AWS

Untuk informasi selengkapnya, lihat [PutDeliverySource](#) di Referensi API Amazon CloudWatch Logs.

3. Hubungkan sumber pengiriman ke tujuan pengiriman:

```
aws logs create-delivery --region region \  
  --delivery-source-name cluster-logs-source \  
  --delivery-destination-arn destination-arn
```

Ganti:

- *region*— yang AWS Region, seperti us-east-1
- *cluster-logs-source*— Nama sumber pengiriman Anda
- *destination-arn*— ARN tujuan pengiriman Anda

Untuk informasi selengkapnya, lihat [CreateDelivery](#) di Referensi API Amazon CloudWatch Logs.

Jalur dan nama aliran log penjadwal

Jalur dan nama untuk log penjadwal AWS PCS 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/slurmctlid_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 penjadwal

AWS Log penjadwal PCS 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": "25.05",  
  "scheduler_patch_version": "3",  
  "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 instans EC2 yang diluncurkan oleh 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 instans menggunakan templat peluncuran EC2 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 penggunaan AWS PCS CloudWatch.

Topik

- [Memantau metrik AWS PCS menggunakan CloudWatch](#)
- [Memantau instans AWS PCS menggunakan Amazon CloudWatch](#)

Memantau metrik AWS PCS menggunakan CloudWatch

Anda dapat memantau kesehatan klaster AWS PCS 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 PCS menerbitkan metrik berikut ke dalam ruang nama AWS/PCS di. CloudWatch Mereka memiliki dimensi tunggal, `ClusterId`.

Nama	Deskripsi	Unit
ActualCapacity	IdleCapacity + UtilizedCapacity	Hitungan
CapacityUtilization	UtilizedCapacity / ActualCapacity	Hitungan
DesiredCapacity	ActualCapacity + PendingCapacity	Hitungan
IdleCapacity	Hitungan instance yang berjalan tetapi tidak dialokasikan untuk pekerjaan	Hitungan
UtilizedCapacity	Hitungan instance yang berjalan dan dialokasikan untuk pekerjaan	Hitungan

Memantau instans AWS PCS menggunakan Amazon CloudWatch

AWS PCS meluncurkan instans Amazon EC2 sesuai kebutuhan untuk memenuhi persyaratan penskalaan yang ditentukan dalam grup node komputasi PCS Anda. Anda dapat memantau instance ini saat sedang berjalan 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 volume EBS yang mendukung instance. Untuk menyimpan metrik atau data pencatatan dari instans yang diluncurkan oleh 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 instans AWS PCS

Untuk memantau instance yang diluncurkan oleh PCS, temukan instance yang sedang berjalan yang terkait dengan cluster atau grup node komputasi. Kemudian, di konsol EC2 untuk instance

tertentu, periksa bagian Status dan alarm dan Pemantauan. Jika akses login dikonfigurasi untuk instance tersebut, Anda dapat menghubungkannya dan memeriksa berbagai file log pada instance. Untuk informasi selengkapnya tentang mengidentifikasi instance mana yang dikelola oleh PCS, lihat [Menemukan instance grup node komputasi di PCS AWS](#).

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 tempat yang dapat diambil oleh instance PCS.
3. Tulis template peluncuran EC2 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, dan [Menggunakan template peluncuran Amazon EC2 dengan PCS AWS](#).

Buat konfigurasi CloudWatch Agen

Sebelum menerapkan CloudWatch agen pada instance Anda, Anda harus membuat file konfigurasi JSON 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 AWS CLI diinstal, buat file CloudWatch konfigurasi bernama config.json dengan konten yang mengikuti. Anda juga dapat menggunakan URL berikut 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 khusus PC yang berjalan selama konfigurasi instans
- `/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, seperti SSH, 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 komputasi PCS. 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 perintah AWS CLI yang mengikuti. Sebelum menjalankan perintah, buat penggantian ini:

- Ganti `amzn-s3-demo-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://amzn-s3-demo-bucket
```

Selanjutnya, unggah file ke ember.

```
aws s3 cp ./config.json s3://amzn-s3-demo-bucket/
```

Simpan sebagai parameter SSM

Untuk menyimpan file Anda sebagai parameter SSM, gunakan perintah berikut. Sebelum menjalankan perintah, buat penggantian ini:

- Ganti *region-code* dengan Wilayah AWS tempat Anda bekerja dengan AWS PCS.
- (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 parameter SSM di profil instance grup node Anda.

```
aws ssm put-parameter \  
  --region region-code \  
  --name "AmazonCloudWatch-PCS" \  
  --type String \  
  --value file://config.json
```

Tulis template peluncuran EC2

Detail spesifik untuk template peluncuran tergantung pada apakah file konfigurasi Anda disimpan dalam S3 atau SSM.

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:

- *amzn-s3-demo-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://amzn-s3-demo-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 instance IAM untuk grup node harus memiliki akses ke bucket. Berikut adalah contoh kebijakan IAM untuk bucket dalam skrip data pengguna di atas.

JSON

```

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

```

Perhatikan juga bahwa instance harus mengizinkan lalu lintas keluar ke S3 dan titik akhir. CloudWatch ini dapat dilakukan dengan menggunakan grup keamanan atau titik akhir VPC, tergantung pada arsitektur cluster Anda.

Gunakan konfigurasi yang disimpan di SSM

Skrip ini menginstal CloudWatch agen, mengimpor file konfigurasi dari parameter SSM, dan meluncurkan agen dengannya. CloudWatch 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 instans IAM untuk grup node harus memiliki CloudWatchAgentServerPolicy lampiran padanya.

Jika nama parameter Anda tidak dimulai dengan AmazonCloudWatch- Anda perlu secara khusus menambahkan akses baca ke parameter SSM di profil instance grup node Anda. Berikut adalah contoh kebijakan IAM yang menggambarkan ini untuk awalan. *DOC-EXAMPLE-PREFIX*

JSON

```
{
  "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 titik akhir CloudWatch. Ini dapat dilakukan dengan menggunakan grup keamanan atau titik akhir VPC, tergantung pada arsitektur cluster Anda.

Pencatatan panggilan API Layanan Komputasi AWS Paralel menggunakan AWS CloudTrail

AWS PCS terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS PCS. CloudTrail menangkap semua panggilan API untuk AWS PCS sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol AWS PCS dan panggilan kode ke operasi AWS PCS API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail 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 untuk 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 Informasi PCS 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 Akun AWS Anda, 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 notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua tindakan AWS PCS dicatat oleh CloudTrail dan didokumentasikan dalam [Referensi API 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 itu dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#).

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 panggilan API publik, jadi file tersebut 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": [
    {
      "computeNodeId": "abcdef0123"
    }
  ],
  "queueName": "all"
},
"responseElements": {
  "queue": {
    "arn": "arn:aws:pcs:us-east-1:609783872011:cluster/abcdef0123/queue/
abcdef0123",
    "clusterId": "abcdef0123",
    "computeNodeGroupConfigurations": [
      {
        "computeNodeId": "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 PCS AWS

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](#)
- [Kuota layanan](#)
 - [Kuota internal](#)
 - [Kuota yang relevan untuk layanan lain AWS](#)

Titik akhir layanan

Nama wilayah	Wilayah	Titik Akhir	Protokol
US East (Ohio)	us-east-2	pcs.us-east-2.amazonaws.com	HTTPS
		pcs-fips.us-east-2.amazonaws.com	
		pcs-fips.us-east-2.api.aws	
		pcs.us-east-2.api.aws	
US East (N. Virginia)	us-east-1	pcs.us-east-1.amazonaws.com	HTTPS

Nama wilayah	Wilayah	Titik Akhir	Protokol
		<p>pcs-fips.us-east-1 .amazonaws.com</p> <p>pcs-fips.us-east-1 .api.aws</p> <p>pcs.us-east-1.api.aws</p>	
US West (Oregon)	us-west-2	<p>pcs.us-west-2.amaz onaws.com</p> <p>pcs-fips.us-west-2 .amazonaws.com</p> <p>pcs-fips.us-west-2 .api.aws</p> <p>pcs.us-west-2.api.aws</p>	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	<p>pcs.ap-southeast-1 .amazonaws.com</p> <p>pcs.ap-tenggara 1.api.aws</p>	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	<p>pcs.ap-southeast-2 .amazonaws.com</p> <p>pcs.ap-tenggara 2.api.aws</p>	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	<p>pcs.ap-northeast-1 .amazonaws.com</p> <p>pcs.ap-northeast-1 .api.aws</p>	HTTPS

Nama wilayah	Wilayah	Titik Akhir	Protokol
Europe (Frankfurt)	eu-central-1	pcs.eu-central-1.a amazonaws.com pcs.eu-central-1.a pi.aws	HTTPS
Europe (Ireland)	eu-west-1	pcs.eu-west-1.amaz onaws.com pcs.eu-west-1.api.aws	HTTPS
Europe (London)	eu-west-2	pcs.eu-west-2.amaz onaws.com pcs.eu-west-2.api.aws	HTTPS
Europe (Stockholm)	eu-north-1	pcs.eu-north-1.ama zonaws.com pcs.eu-utara-1.api .aws	HTTPS
AWS GovCloud (AS- Timur)	us-gov-east-1	buah. us-gov-ea st-1.amazonaws.com pcs-fips. us-gov-ea st-1.amazonaws.com pcs-fips. us-gov-ea st-1.api.aws buah. us-gov-ea st-1.api.aws	HTTPS

Nama wilayah	Wilayah	Titik Akhir	Protokol
AWS GovCloud (AS-Barat)	us-gov-west-1	buah. us-gov-west-1.amazonaws.com pcs-fips. us-gov-west-1.amazonaws.com pcs-fips. us-gov-west-1.api.aws buah. us-gov-west-1.api.aws	HTTPS

Kuota layanan

Nama	Default	Dapat disesuaikan	Deskripsi
klaster	5	Ya	Jumlah maksimum cluster per. AWS Region

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 AWS Parallel Computing Service (PCS) di [Konsol Manajemen AWS](#) Untuk meminta peningkatan kuota untuk nilai yang ditampilkan sebagai dapat disesuaikan, lihat [Meminta Peningkatan Kuota di Panduan Pengguna Service Quotas](#).

⚠ Important

Ingatlah untuk memeriksa AWS Region pengaturan saat ini di Konsol Manajemen AWS.

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 AWS Region.
Hitung grup node per cluster	10	Tidak	Jumlah maksimum grup node komputasi per cluster.
Antrian per cluster	10	Tidak	Jumlah maksimum antrian per cluster.

Kuota yang relevan untuk layanan lain AWS

AWS PCS menggunakan AWS layanan lain. Kuota layanan Anda untuk layanan tersebut memengaruhi penggunaan AWS PCS Anda.

Kuota layanan Amazon EC2 yang memengaruhi PCS AWS

- Permintaan instans spot
- Menjalankan instance sesuai permintaan
- Templat peluncuran
- Luncurkan versi template
- Permintaan API Amazon EC2

Untuk informasi selengkapnya, lihat [kuota layanan Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud](#).

Memecahkan masalah di Layanan Komputasi AWS Paralel

Topik berikut memberikan panduan untuk memecahkan masalah beberapa masalah yang mungkin Anda temui di AWS PCS.

- [Pembaruan klaster](#)
- [Hitung masalah bootstrap node](#)
- [Pengaturan Slurm kustom](#)
- [Instans EC2 dihentikan setelah reboot](#)
- [Identitas dan akses](#)
- [Masalah reboot slurm](#)

Instans EC2 di AWS PCS dihentikan dan diganti setelah reboot

Ikhtisar masalah

Setelah instance EC2 dalam grup node komputasi di-boot ulang, AWS PCS secara otomatis mengakhiri dan menggantikan instance.

Mengapa ini terjadi

AWS PCS tidak mendukung reboot instance. Jika instans EC2 di-boot ulang, AWS PCS menganggap instance tidak sehat dan menggantikannya. Jika AWS PCS terus-menerus menghentikan dan mengganti instance Anda, itu mungkin karena sesuatu me-reboot instance Anda setelah diluncurkan. Beberapa contoh termasuk reboot dengan otomatisasi pada instans EC2 (seperti reboot otomatis setelah patch), otomatisasi eksternal untuk instans EC2 (seperti aplikasi manajemen jaringan), AWS layanan lain (seperti AWS Systems Manager), atau reboot manual oleh seseorang.

Apa yang harus dilakukan

Anda dapat memeriksa `slurmctld` atau `slurmd` log Anda untuk melihat apakah instance Anda di-boot ulang. Untuk informasi selengkapnya, lihat [Log penjadwal di AWS PCS](#) dan [Memantau instans AWS PCS menggunakan Amazon CloudWatch](#). Contoh entri `slurmctld` log berikut menunjukkan bahwa instance reboot:

Example

```
[2024-09-12T06:42:50.393+00:00] validate_node_specs: Node Login-1 unexpectedly rebooted  
boot_time=1726123354 last_response=1726123285
```

Mem-boot ulang karena menambal

Reboot sering diperlukan setelah Anda menerapkan tambalan. Jangan menerapkan patch langsung ke instance EC2 yang merupakan bagian dari grup node komputasi AWS PCS. Jika Anda harus menambal instans EC2, Anda harus menerapkan tambalan ke Amazon Machine Image (AMI) yang diperbarui dan memperbarui grup node komputasi Anda untuk menggunakan AMI yang diperbarui. Instans EC2 baru yang diluncurkan AWS PCS untuk grup node komputasi tersebut akan menggunakan AMI yang diperbarui (ditambal). Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Memecahkan masalah bootstrap node komputasi dan masalah pendaftaran di PCS AWS

Ketika node komputasi gagal melakukan bootstrap atau mendaftar dengan benar dengan cluster AWS PCS Anda, Anda mungkin mengalami gejala berikut:

- Pekerjaan tidak dimulai
- Anda tidak dapat terhubung ke instance di AWS Systems Manager
- Instans dimatikan secara tak terduga
- Instans terus diganti

Kegagalan ini dapat disebabkan oleh masalah selama peluncuran instans EC2 atau selama proses bootstrap node komputasi AWS PCS. Topik ini menjelaskan prosedur untuk membantu Anda memecahkan masalah selama proses bootstrap node AWS PCS. Untuk informasi selengkapnya tentang pemecahan masalah peluncuran instans EC2, lihat [Memecahkan masalah peluncuran instans Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud](#).

Kegagalan bootstrap terjadi ketika instans EC2 berhasil diluncurkan tetapi gagal selama proses bergabung dengan cluster AWS PCS. Proses bootstrap mencakup dua fase utama:

- Registrasi node - Instans EC2 memanggil tindakan [RegisterComputeNodeGroupInstance](#) AWS PCS API untuk mendaftar dengan layanan AWS PCS. Kegagalan dapat terjadi karena masalah berikut:
 - Izin
 - [Profil contoh yang salah](#)
 - Jaringan
 - [Tidak dapat terhubung ke titik akhir AWS PCS](#)
 - [Titik akhir AWS PCS yang salah dikonfigurasi](#)
 - [Instance dalam subnet publik tanpa IP publik](#)
 - [Instans multi-NIC di subnet publik](#)
 - Rahasia cluster
 - [Rahasia cluster telah dihapus atau ditandai untuk dihapus](#)
- Integrasi slurm — Instance berjalan `slurmd` dan bergabung dengan cluster Slurm. Kegagalan dapat terjadi karena masalah berikut:
 - Izin
 - [Konfigurasi grup keamanan](#)
 - [Slurmctld tidak dapat melakukan ping node komputasi](#)
 - Pengaturan AMI khusus
 - [Driver NVIDIA hilang](#)
 - [ResumeTimeout mencapai](#)

Bagaimana Slurm bekerja pada PCS AWS

Ini mungkin membantu Anda untuk membandingkan cara standar Slurm bekerja dengan cara Slurm bekerja pada PCS. AWS

Pemrosesan pekerjaan Slurm standar

Langkah-langkah berikut terjadi dalam pemrosesan pekerjaan Slurm standar:

1. Saat Anda mengirimkan pekerjaan, `slurmctld` memvalidasi dan mengantri pekerjaan.
2. Ketika sumber daya tersedia, `slurmctld` mengalokasikan node yang ada.
3. `slurmd` daemon menjalankan pekerjaan pada node yang dialokasikan.

Pemrosesan pekerjaan slurm pada PCS AWS

Langkah-langkah berikut terjadi dalam pemrosesan pekerjaan AWS PCS:

1. Saat Anda mengirimkan pekerjaan, `slurmctld` memvalidasi dan mengantri pekerjaan.
2. Ketika kapasitas tambahan diperlukan, AWS PCS menggunakan template peluncuran untuk grup node komputasi untuk meluncurkan instans EC2 baru.
3. Contoh baru bootstrap ke dalam cluster:
 - a. Instans mendaftar dengan AWS PCS.
 - b. Instance bergabung dengan cluster Slurm.
4. Ketika sumber daya siap, `slurmctld` alokasikan node (termasuk yang baru di-bootstrap).
5. `slurmd` menjalankan pekerjaan pada node yang dialokasikan.

Ambil log contoh

Langkah pertama dalam memecahkan masalah bootstrap node komputasi adalah mengambil log instance. Gunakan salah satu metode berikut:

AWS CLI

Ambil output konsol dari node komputasi menggunakan perintah berikut:

```
aws ec2 get-console-output --region us-east-1 --instance-id i-1234567890abcdef0 --output text
```

Ganti *us-east-1* dengan AWS Region Anda dan *i-1234567890abcdef0* dengan ID instans Anda.

AWS Systems Manager

Jika Anda dapat terhubung ke instance menggunakan Systems Manager, Anda dapat melihat file log bootstrap secara langsung:

1. Connect ke instance menggunakan Systems Manager. Untuk informasi selengkapnya, lihat [Memulai sesi](#) di Panduan Pengguna Systems Manager.
2. Lihat file log bootstrap:

```
sudo cat /var/log/amazon/pcs/bootstrap.log
```

Note

Jika ada masalah selama fase inisialisasi, Anda mungkin perlu menunggu sekitar 20 menit sebelum dapat terhubung ke instance. Systems Manager dan layanan SSH dimulai hanya setelah inisialisasi selesai atau ketika eksekusi bootstrap mencapai batas waktu jika terjadi kegagalan.

Mengambil VPC/Subnet/Security Grup dari ID instance

Untuk memecahkan masalah dengan node komputasi, Anda mungkin perlu mengambil informasi tentang VPC, subnet, dan grup keamanan yang terkait dengan instans Anda. Jika Anda tidak tahu contoh Anda IDs, lihat [Menemukan instance grup node komputasi di PCS AWS](#).

Konsol Manajemen AWS

Untuk mendapatkan VPC, subnet, dan grup keamanan

1. Buka [konsol Amazon EC2](#).
2. Pilih Instans.
3. Dalam tabel Instances, pilih ID instance.
4. Temukan ID VPC dan ID Subnet dalam ringkasan instance yang ditampilkan untuk instance tersebut.
5. Dalam ringkasan instance, pilih tab Keamanan.
6. Temukan Grup keamanan di tab Keamanan.

AWS CLI

Gunakan perintah berikut untuk mengambil VPC, subnet, dan informasi grup keamanan untuk instans Anda:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query  
'Reservations[*].Instances[*].  
{InstanceId:InstanceId,VpcId:VpcId,SubnetId:SubnetId,SecurityGroups:SecurityGroups[*]}.GroupI  
--output table
```

Masalah pendaftaran node

Registrasi node adalah tindakan pertama yang dijalankan oleh node komputasi selama bootstrap. Node memanggil titik akhir AWS PCS API untuk mendaftarkan dirinya dengan AWS PCS. Kegagalan pendaftaran biasanya menampilkan pesan kesalahan yang mirip dengan yang berikut ini:

```
<13>Nov  5 08:10:27 user-data: Recipe: aws-pcs-environment::node_registration
<13>Nov  5 08:10:27 user-data:  * ruby_block[Register NodeGroup Instance] action
  run[2024-11-05T08:10:27+00:00] INFO: Processing ruby_block[Register NodeGroup
  Instance] action run (aws-pcs-environment::node_registration line 19)
<13>Nov  5 08:15:46 user-data:
<13>Nov  5 08:15:46 user-data:
<13>Nov  5 08:15:46 user-data:
=====
<13>Nov  5 08:15:46 user-data:      Error executing action `run` on resource
  'ruby_block[Register NodeGroup Instance]'
<13>Nov  5 08:15:46 user-data:
=====
<13>Nov  5 08:15:46 user-data:
<13>Nov  5 08:15:46 user-data:      EOFError
```

Profil contoh yang salah

Jika instance tidak dapat mendaftar, verifikasi bahwa profil instance yang terkait dengan node komputasi memiliki `pcs:RegisterComputeNodeGroupInstance` izin.

Untuk informasi selengkapnya tentang cara membuat profil instans yang valid, lihat [Buat profil instance untuk AWS PCS](#).

Tidak dapat terhubung ke titik akhir AWS PCS

Jika node komputasi Anda berada dalam subnet pribadi, pastikan Anda telah mengonfigurasi titik akhir VPC untuk AWS PCS atau subnet Anda memiliki rute ke gateway NAT untuk akses internet. Untuk informasi selengkapnya, lihat berikut ini:

- [Akses AWS layanan menggunakan titik akhir VPC antarmuka](#) di panduan Amazon Virtual Private Cloud. AWS PrivateLink
- [Titik akhir dan kuota layanan untuk PCS AWS](#).
- [Hubungkan VPC Anda ke jaringan lain di Panduan](#) Pengguna Amazon Virtual Private Cloud
- [AWS Jaringan PCS](#)

Titik akhir AWS PCS yang salah dikonfigurasi

Jika Anda melihat pesan galat yang mirip dengan berikut ini, verifikasi kebijakan yang terkait dengan titik akhir AWS VPC PC Anda:

```
com.amazon.coral.security.AccessDeniedException: User: arn:aws:sts::xxx:assumed-role/rolename/i-instanceid is not authorized to perform: pcs:RegisterComputeNodeGroupInstance on resource: arn:aws:pcs:us-west-2:xxx:cluster/cluster-id as either the resource does not exist, some policy explicitly denies access, or no policy grants access
```

Untuk informasi selengkapnya tentang cara mengonfigurasi titik akhir antarmuka VPC untuk AWS PCS, lihat [Akses AWS Parallel Computing Service menggunakan endpoint antarmuka \(\)AWS PrivateLink](#)

Instance dalam subnet publik tanpa IP publik

Jika subnet Anda tidak mengaktifkan IP publik penetapan otomatis dan konfigurasi rute Anda menggunakan gateway internet, instance tidak dapat berkomunikasi dengan PCS API. AWS

Contoh di subnet dengan gateway internet harus memiliki alamat IP publik. Untuk mengatasi masalah ini, pilih salah satu opsi berikut:

- Tambahkan titik akhir VPC untuk AWS PCS ke VPC cluster Anda. Hal ini memungkinkan instance untuk berkomunikasi dengan AWS PCS tanpa perlu alamat IP publik untuk melewati gateway internet.
- Gunakan subnet pribadi dengan gateway NAT, sehingga alamat IP publik tidak diperlukan.
- Aktifkan penetapan alamat IP publik otomatis melalui subnet atau template peluncuran Anda sehingga instance dapat menghubungi API melalui gateway internet. Perhatikan bahwa opsi ini tidak berlaku untuk instance antarmuka multi-jaringan.

Instans multi-NIC di subnet publik

Anda harus menggunakan subnet pribadi jika Anda menggunakan jenis instance yang memiliki beberapa antarmuka jaringan ()NICs.

AWS Alamat IP publik hanya dapat ditetapkan ke instance yang diluncurkan dengan antarmuka jaringan tunggal. Untuk informasi selengkapnya tentang alamat IP, lihat [Menetapkan IPv4 alamat publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jenis instans multi-NIC memerlukan gateway NAT atau proxy internal di subnet untuk mengakses titik akhir PCS. AWS Atau, Anda dapat menambahkan titik akhir VPC untuk AWS PCS ke VPC cluster Anda.

Rahasia cluster telah dihapus atau ditandai untuk dihapus

Jika rahasia bersama Slurm di AWS Secrets Manager telah dihapus atau ditandai untuk dihapus, node komputasi akan gagal mendaftar dan klaster Anda akan mengalami gangguan.

AWS PCS secara otomatis membuat rahasia bersama Slurm di AWS Secrets Manager (dengan format nama:pcs!slurm-secret-<cluster-id>) saat Anda membuat cluster. Rahasia ini diperlukan untuk komunikasi yang aman di cluster. Untuk informasi selengkapnya, lihat [Bekerja dengan rahasia cluster di AWS PCS](#).

Jika rahasia ini dihapus atau ditandai untuk dihapus, node baru tidak akan dapat bergabung dengan cluster dan pengontrol atau daemon cluster lainnya (seperti slurmd dan slurmdbd) mungkin tidak dapat bergabung kembali dengan cluster jika dimulai ulang.

Untuk mengatasi masalah ini, Anda dapat memulihkan rahasia yang dihapus jika masih dalam jendela pemulihan. Untuk petunjuk selengkapnya, lihat [Mengembalikan AWS rahasia Secrets Manager](#).

Jika jendela pemulihan kedaluwarsa, rahasia tidak dapat dipulihkan dan cluster AWS PCS yang terpengaruh tidak dapat dipulihkan. Anda perlu membuat cluster baru dengan konfigurasi yang sama. AWS PCS secara otomatis membuat rahasia penjadwal baru.

Masalah bergabung dengan klaster slurm

Setelah registrasi node berhasil, node komputasi mencoba untuk bergabung dengan cluster Slurm. slurmdDaemon pada node menghubungi controller Slurm untuk mendaftar dengan cluster. Kegagalan bergabung slurm biasanya menampilkan pesan kesalahan yang mirip dengan yang berikut ini:

```
<13>Nov  5 17:20:29 user-data: [2024-11-05T17:20:28+00:00] FATAL:
  Mixlib::ShellOut::ShellCommandFailed: service[slurmd] (aws-pcs-slurm::finalize_slurm
  line 18) had an error: Mixlib::ShellOut::ShellCommandFailed: Expected process to exit
  with [0], but received '1'
<13>Nov  5 17:20:29 user-data: ---- Begin output of ["/usr/bin/systemctl", "--system",
  "start", "slurmd"] ----
<13>Nov  5 17:20:29 user-data: STDOUT:
```

```
<13>Nov 5 17:20:29 user-data: STDERR: Job for slurmd.service failed because the control process exited with error code. See "systemctl status slurmd.service" and "journalctl -xe" for details.
<13>Nov 5 17:20:29 user-data: ---- End output of ["/usr/bin/systemctl", "--system", "start", "slurmd"] ----
```

Konfigurasi grup keamanan

Verifikasi bahwa grup keamanan Anda dikonfigurasi dengan benar untuk memungkinkan komunikasi antara node komputasi dan pengontrol Slurm. Grup keamanan harus mengizinkan lalu lintas berikut:

- Port 6817 slurmd untuk berkomunikasi slurmctld
- Port 6818 untuk slurmctld melakukan ping slurmd

Untuk informasi selengkapnya tentang persyaratan grup keamanan, lihat topik berikut:

- [Buat grup keamanan untuk AWS PCS](#)
- [Buat template peluncuran untuk AWS PCS](#)
- [Persyaratan dan pertimbangan kelompok keamanan](#)

Important

Grup keamanan klaster yang Anda kaitkan dengan klaster selama pembuatan klaster juga harus dikonfigurasi dalam grup keamanan grup node komputasi Anda agar node komputasi dapat berkomunikasi dengan pengontrol.

Driver NVIDIA hilang

Jika instance bootstrap dengan benar tetapi pekerjaan tidak dimulai, dan Anda melihat pesan kesalahan yang mirip dengan yang berikut di log instance Anda, Anda mungkin kehilangan driver NVIDIA:

```
<13>Dec 2 13:52:00 user-data: [2024-12-02T13:52:00.094+00:00] - /opt/aws/pcs/bin/pcs_bootstrap_config_always.sh: INFO: nvidia-smi not found!
...
<13>Dec 2 13:54:10 user-data: Job for slurmd.service failed because the control process exited with error code. See "systemctl status slurmd.service" and "journalctl -xe" for details.
```

```
<13>Dec 2 13:54:12 user-data: [2024-12-02T13:54:12.718+00:00] - /opt/aws/pcs/bin/pcs_bootstrap_finalize.sh: INFO: systemctl could not start slurmd!
```

Jika Anda terhubung ke instance dan memeriksa status `slurmd` daemon, Anda mungkin melihat kesalahan yang mirip dengan berikut ini:

```
$ systemctl status slurmd
...
fatal: can't stat gres.conf file /dev/nvidia0: No such file or directory
```

Untuk mengatasi masalah ini, instal driver NVIDIA pada AMI kustom Anda. Untuk informasi selengkapnya, lihat [Langkah 4 - \(Opsional\) Instal driver tambahan, perpustakaan, dan perangkat lunak aplikasi](#).

ResumeTimeout mencapai

Jika node komputasi dan instans EC2-nya dihentikan karena node tidak sehat, AWS PCS mungkin tidak mendukung AMI atau mungkin ada masalah jaringan. Instans EC2 berjalan selama sekitar 30 menit sampai Slurm ResumeTimeout tercapai dan menandai node sebagai DOWN.

Jika instance tidak bootstrap dengan benar dan tidak terdaftar dengan AWS PCS (tidak ada `RegisterComputeNodeGroupInstance` panggilan untuk instans EC2), periksa log instance Anda untuk pesan kesalahan yang mirip dengan berikut ini:

```
/opt/aws/pcs/bin/pcs_bootstrap_init.sh: No such file or directory
```

Kesalahan ini menunjukkan bahwa perangkat lunak bootstrap AWS PCS bukan bagian dari AMI. Untuk mengatasi masalah ini, pastikan AMI kustom Anda menyertakan perangkat lunak bootstrap AWS PCS. Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon Kustom \(AMIs\) untuk AWS PCS](#).

Slurmctld tidak dapat melakukan ping node komputasi

Jika instance menjalankan prosedur bootstrap dengan benar dan terdaftar dengan AWS PCS, tetapi `slurmctld` tidak dapat melihatnya dan mengirimkan pekerjaan ke sana, instance diatur ke DOWN setelah beberapa waktu dan kemudian dihentikan.

Ini mungkin disebabkan oleh grup keamanan yang salah konfigurasi. Misalnya, jika port 6817 diaktifkan untuk memungkinkan `slurmd` untuk berkomunikasi dengan `slurmctld`, tetapi port 6818 tidak ada pada `slurmctld` untuk memungkinkan ping ke `slurmd`.

Verifikasi bahwa grup keamanan Anda menyertakan semua aturan yang diperlukan sebagaimana didokumentasikan dalam [Persyaratan dan pertimbangan kelompok keamanan](#).

Riwayat dokumen untuk Panduan Pengguna AWS PCS

Tabel berikut menjelaskan perubahan penting pada dokumentasi untuk AWS PCS.

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Maret 10, 2026	Agen PCS yang diperbarui	<p>Memperbarui topik AMI untuk agen AWS PCS 1.3.2-1. Memperbaiki masalah yang memengaruhi RHEL 8.10 dan Rocky Linux 8.10 compute node bootstrap . Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan AWS Versi agen PCS.</p>	N/A
Februari 11, 2026	AWS PCS dirilis di Asia Pasifik (Mumbai) dan Eropa (Paris)	<p>AWS PCS sekarang tersedia di Asia Pasifik (Mumbai) (ap-south-1) dan Eropa (Paris) (eu-west-3).</p> <p>CloudFormation template tersedia untuk memulai di Asia Pasifik (Mumbai) AWS Region dan Eropa (Paris) AWS Region. Untuk informasi selengkapnya, lihat Gunakan CloudFormation untuk membuat contoh cluster AWS PCS dan</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
		CloudFormation template untuk membuat contoh cluster AWS PCS.	
November 18, 2025	Fitur baru: Slurm REST API	Slurm REST API sekarang didukung untuk Slurm 25.05 atau yang lebih baru. Untuk informasi selengkapnya, lihat Slurm REST API di PCS AWS.	AWS SDK: 2025-11-18

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
November 10, 2025	Fitur baru: Dukungan plugin filter Slurm CLI	AWS PCS sekarang mendukung plugin filter Slurm CLI untuk menjalankan skrip Lua kustom yang memvalidasi dan memodifikasi parameter pengiriman pekerjaan sebelum mencapai pengontrol Slurm. Gunakan filter CLI untuk menerapkan kebijakan khusus, menetapkan parameter default, dan memberikan panduan pengguna selama pengiriman pekerjaan. Fitur ini membutuhkan Slurm versi 25.05 atau yang lebih baru. Untuk informasi selengkapnya, lihat Gunakan Plugin Filter CLI Slurm untuk menyesuaikan pengiriman pekerjaan di PCS AWS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
November 7, 2025	Agen PCS yang diperbarui	Memperbarui topik AMI untuk agen AWS PCS 1.3.1-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan AWS Versi agen PCS .	N/A
November 3, 2025	Diperbarui agen PCS dan installer Slurm	Memperbarui topik AMI untuk agen AWS PCS 1.3.0-1 dan installer Slurm 24.11.6-2, 24.05.8-2, dan 23.11.10-4. Daftar terbaru dari sistem operasi yang didukung. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan AWS Versi agen PCS .	N/A
Oktober 23, 2025	Diperbarui konten: pcs-multi-cluster-login - configure.sh	Memperbaiki beberapa kesalahan dalam skrip konfigurasi node login multi-cluster. Untuk informasi selengkapnya, lihat AWS Kode skrip konfigurasi node login multi-cluster PCS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Oktober 21, 2025	Fitur baru: Rotasi rahasia cluster	<p>AWS PCS sekarang mendukung rotasi rahasia cluster untuk meningkatkan keamanan. Untuk informasi selengkapnya, lihat Memutar rahasia cluster di AWS PCS.</p> <p>Memperbarui izin administrator minimum untuk mendukung rotasi rahasia cluster. Untuk informasi selengkapnya, lihat Izin minimum untuk AWS PCS.</p>	N/A
Oktober 17, 2025	Topik baru: skrip konfigurasi node login multi-cluster	<p>Menambahkan topik baru yang menyediakan skrip untuk mengonfigurasi node login mandiri untuk terhubung ke beberapa cluster AWS PCS. Skrip mengotomatiskan konfigurasi beberapa sackd daemon Slurm dan membuat skrip aktivasi untuk interaksi cluster.</p> <p>Untuk informasi selengkapnya, lihat Menghubungkan node login mandiri ke beberapa cluster di PCS AWS.</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Oktober 16, 2025	Diperbarui untuk Slurm 25.05	<p>Memperbarui panduan pengguna untuk dukungan Slurm 25.05. Slurm 25.05 sekarang menjadi versi default. Untuk informasi selengkapnya, lihat berikut ini:</p> <ul style="list-style-type: none">• Versi slurm di PCS AWS• Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS• Catatan rilis untuk sampel AWS PCS AMIs	N/A
Oktober 16, 2025	Agen PCS yang diperbarui	<p>Memperbarui topik AMI untuk agen AWS PCS 1.2.2-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan AWS Versi agen PCS.</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Oktober 2, 2025	Fitur baru: Slurm node reboot, pembaruan cluster, dan pengaturan Slurm kustom	<p>AWS PCS menambahkan dukungan untuk beberapa fitur baru:</p> <ul style="list-style-type: none">• Slurm node reboot - Gunakan <code>scontrol reboot</code> perintah asli Slurm untuk me-reboot node komputasi tanpa penggantian instance. Untuk informasi selengkapnya, lihat Mem-boot ulang node komputasi dengan Slurm di PCS AWS.• Pembaruan cluster — Ubah konfigurasi klaster pasca-pembuatan tanpa membangun kembali. Untuk informasi selengkapnya, lihat Memperbarui cluster di AWS PCS.• Pengaturan kustom slurm - Konfigurasi parameter Slurm lanjutan di seluruh sumber daya Cluster, Antrian, dan Grup Node Komputasi. Untuk informasi selengkapnya, lihat Mengkonfigurasi pengaturan	2025-10-01

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
		Slurm khusus di PCS AWS.	
September 23, 2025	Topik pemecahan masalah baru: Menghitung masalah bootstrap node	Menambahkan panduan pemecahan masalah untuk mendiagnosis dan menyelesaikan masalah bootstrap node komputasi. Untuk informasi selengkapnya, lihat Memecahkan masalah bootstrap node komputasi dan masalah pendaftaran di PCS AWS.	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
September 17, 2025	Fitur baru: Blok Kapasitas untuk ML	<p>AWS PCS sekarang mendukung Blok Kapasitas Amazon EC2 untuk ML, yang memungkinkan Anda memesan instans komputasi akselerasi berbasis GPU untuk cluster Anda. Untuk informasi selengkapnya, lihat Menggunakan Blok Kapasitas Amazon EC2 untuk ML dengan PCS AWS.</p> <p>Izin minimum untuk mendukung Blok Kapasitas sekarang menjadi bagian dari izin minimum untuk administrator layanan. Untuk informasi selengkapnya, lihat Izin minimum untuk AWS PCS.</p>	2025-09-17
September 11, 2025	Pembaruan kebijakan terkelola AWS	<p>AWS PCS memperbarui AWSPCSService RolePolicy untuk mendukung Blok Kapasitas. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola untuk Layanan Komputasi AWS Paralel.</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Agustus 14, 2025	Diperbarui dokumentasi profil contoh	<p>Meningkatkan dokumentasi profil instans dengan instruksi CLI komprehensif untuk membuat peran IAM dan profil instans. Menambahkan step-by-step prosedur untuk menyiapkan profil instans menggunakan AWS CLI dan panduan yang ditingkatkan untuk menemukan profil instance yang digunakan dengan AWS PCS.</p> <p>Untuk informasi selengkapnya, lihat Profil instans IAM untuk Layanan Komputasi AWS Paralel.</p>	2025-08-14

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Agustus 1, 2025	Topik baru: Plugin SPANK	<p>Menambahkan dokumentasi untuk plugin SPANK (Slurm Plug-in Architecture for Node and job Kontrol) yang dapat Anda gunakan untuk memperluas dan memodifikasi perilaku Slurm selama peluncuran pekerjaan dan eksekusi pada cluster PCS. AWS</p> <p>Untuk informasi selengkapnya, lihat Perluas fungsionalitas Slurm pada AWS PCS dengan plugin SPANK.</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Agustus 1, 2025	IPv6 dukungan jaringan	<p>Menambahkan dukungan untuk IPv6 jaringan saat membuat cluster AWS PCS. Sekarang Anda dapat memilih IPv6 sebagai jenis jaringan untuk kluster Anda, dengan pembaruan yang sesuai dengan persyaratan VPC, konfigurasi subnet, pengaturan grup keamanan, dan prosedur pembuatan kluster.</p> <p>Untuk informasi selengkapnya, lihat AWS PCS VPC dan persyaratan subnet dan pertimbangan dan Membuat cluster di AWS PCS.</p>	2025-08-01

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Juli 3, 2025	AWS PCS dirilis di Eropa (London)	<p>AWS PCS sekarang tersedia di Eropa (London) (eu-west-2).</p> <p>CloudFormation template tersedia untuk memulai di Eropa (London) AWS Region. Untuk informasi selengkapnya, lihat Gunakan CloudFormation untuk membuat contoh cluster AWS PCS dan CloudFormation template untuk membuat contoh cluster AWS PCS.</p>	N/A
Juli 1, 2025	Instruksi konsol yang diperbarui	<p>Anda sekarang dapat meminta AWS PCS membuat profil instance dasar dan grup keamanan untuk Anda saat Anda membuat cluster dan grup node komputasi di konsol. Untuk informasi lebih lanjut, lihat:</p> <ul style="list-style-type: none">• Membuat cluster di AWS PCS• Membuat grup node komputasi di AWS PCS• Profil instans IAM untuk Layanan Komputasi AWS Paralel	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Juni 23, 2025	Kebijakan terkelola baru: AWSPCSCompute NodePolicy	Menambahkan kebijakan terkelola baru yang memberikan izin ke node komputasi AWS PCS untuk terhubung ke kluster AWS PCS. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola: AWSPCSCompute NodePolicy .	N/A
Juni 19, 2025	Topik baru: log penyelesaian pekerjaan	Gunakan log penyelesaian pekerjaan untuk mencatat detail tentang pekerjaan ketika mereka selesai, tanpa biaya tambahan. Untuk informasi selengkapnya, lihat Log penyelesaian pekerjaan di AWS PCS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Juni 18, 2025	AWS Rilis PCS di AWS GovCloud (US)	<p>AWS PCS sekarang tersedia di AWS GovCloud (AS-Timur) (us-gov-east-1) dan AWS GovCloud (AS-Barat) (us-gov-west-1).</p> <p>CloudFormation template tersedia untuk memulai di AWS GovCloud (US) Regions. Untuk informasi selengkapnya, lihat Gunakan CloudFormation untuk membuat contoh cluster AWS PCS dan CloudFormation template untuk membuat contoh cluster AWS PCS.</p> <p>Untuk informasi selengkapnya tentang titik akhir layanan AWS PCS di AWS GovCloud (US) Regions, lihat Titik akhir dan kuota layanan untuk PCS AWS.</p> <p>Untuk informasi selengkapnya tentang perbedaan AWS GovCloud (US) Regions, lihat AWS PCS AWS GovCloud (US) di Panduan AWS GovCloud (US) Pengguna.</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Juni 18, 2025	Agen PCS yang diperbarui	Memperbarui topik AMI untuk agen AWS PCS 1.2.1-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS .	N/A
15 Mei 2025	Fitur baru: akuntansi	Akuntansi slurm sekarang didukung untuk Slurm 24.11 atau yang lebih baru. Untuk informasi selengkapnya, lihat Akuntansi slurm di PCS AWS .	AWS SDK: 2025-05-15
15 Mei 2025	Diperbarui untuk Slurm 24.11	<p>Memperbarui panduan pengguna untuk dukungan Slurm 24.11.5. Untuk informasi selengkapnya, lihat berikut ini:</p> <ul style="list-style-type: none"> • Versi slurm di PCS AWS • Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS • Catatan rilis untuk sampel AWS PCS AMIs 	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
5 Mei 2025	Diperbarui versi Slurm FAQ	Memperbarui versi Slurm pertanyaan yang sering diajukan (FAQ) tentang versi Slurm mendekati atau melampaui akhir kehidupan (EOL). Untuk informasi selengkapnya, lihat Pertanyaan yang sering diajukan tentang versi Slurm di PCS AWS .	N/A
17 April 2025	Topik baru: cara mendapatkan detail grup node komputasi	Pelajari cara mendapatkan detail untuk grup node komputasi AWS PCS, seperti ID, ARN, dan ID AMI. Untuk informasi selengkapnya, lihat Dapatkan detail grup node komputasi di AWS PCS .	N/A
April 2, 2025	Pemasang Slurm yang diperbarui	Memperbarui topik AMI untuk penginstal Slurm 24.05.7-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Maret 28, 2025	Menambahkan kuota untuk jumlah maksimum grup node komputasi dan antrian	Menambahkan kuota internal yang tidak dapat disesuaikan untuk jumlah maksimum grup node komputasi per cluster dan jumlah antrian maksimum per cluster. Untuk informasi selengkapnya, lihat Kuota internal .	N/A
Maret 14, 2025	Mengubah kunci properti di CloudFormation template	Idsekarang TemplateId untuk CustomLaunchTemplate properti di CloudFormation template. Untuk informasi selengkapnya, lihat Sumber Daya di Bagian dari CloudFormation template untuk AWS PCS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Maret 13, 2025	Ditambahkan informasi versi untuk agen AWS PCS dan Slurm	<p>Menambahkan topik baru yang menjelaskan perubahan untuk setiap versi agen AWS PCS. Untuk informasi selengkapnya, lihat AWS Versi agen PCS.</p> <p>Menambahkan informasi lebih lanjut ke topik versi Slurm yang menjelaskan tanggal dukungan penting dan catatan rilis terperinci untuk dukungan AWS PCS untuk Slurm. Untuk informasi selengkapnya, lihat Versi slurm di PCS AWS.</p>	N/A
7 Maret 2025	Agen PCS yang diperbarui	Memperbarui topik AMI untuk agen AWS PCS 1.2.0-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Februari 3, 2025	Menambahkan topik tentang menggunakan AWS CloudFormation dengan AWS PCS	Menambahkan topik ke panduan pengguna yang memberikan contoh cara menggunakan CloudFormation dengan AWS PCS. Topik menyediakan prosedur untuk menggunakan CloudFormation template sampel untuk membuat cluster AWS PCS sampel, dan menjelaskan secara singkat bagian-bagian dari template itu. Untuk informasi selengkapnya, lihat Memulai dengan CloudFormation dan AWS PCS .	N/A
Desember 18, 2024	Diperbarui untuk Slurm 24.05	Memperbarui panduan pengguna untuk dukungan Slurm 24.05. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan Catatan rilis untuk sampel AWS PCS AMIs .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Desember 18, 2024	Versi NVIDIA yang diperbarui untuk sampel Slurm 23.11 AMIs	Diperbarui driver NVIDIA dan versi CUDA dalam sampel Slurm 23.11. AMIs Untuk informasi selengkapnya, lihat Catatan rilis untuk sampel AWS PCS AMIs .	N/A
Desember 17, 2024	Pemasang Slurm yang diperbarui	Memperbarui topik AMI untuk penginstal Slurm 23.11.10-3. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS .	N/A
Desember 13, 2024	Agen PCS yang diperbarui	Memperbarui topik AMI untuk agen AWS PCS 1.1.1-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS .	N/A
Desember 6, 2024	Agen PCS yang diperbarui dan penginstal Slurm	Memperbarui topik AMI untuk agen AWS PCS 1.1.0-1 dan penginstal Slurm 23.11.10-2. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
Desember 6, 2024	Menambahkan topik tentang dukungan OS	Untuk informasi selengkapnya, lihat Sistem operasi yang didukung di AWS PCS .	N/A
November 8, 2024	Panduan pengguna yang direorganisasi	Kami mengatur ulang panduan pengguna untuk membawa topik ke tingkat atas, memindahkan beberapa topik ke halaman mereka sendiri, dan mengelompokkan topik serupa bersama-sama.	N/A
November 7, 2024	Topik AMI yang diperbarui	<p>Memperbarui topik AMI untuk Slurm 23.11.10 dan libjwt 17.0. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan Langkah 3 - Instal Slurm.</p> <p>Menyederhanakan dan mengoreksi catatan rilis untuk AMIs. Untuk informasi selengkapnya, lihat Catatan rilis untuk sampel AWS PCS AMIs.</p>	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
November 7, 2024	Menambahkan topik baru tentang menggunakan volume EBS terenkripsi dengan PCS AWS	Menambahkan topik yang menjelaskan kebijakan kunci KMS yang diperlukan untuk volume EBS terenkripsi di PCS. AWS Untuk informasi selengkapnya, lihat Kebijakan kunci KMS yang diperlukan untuk digunakan dengan volume EBS terenkripsi di PCS AWS .	N/A
Oktober 18, 2024	AWS Agen PCS 1.0.1-1 dirilis	Diperbarui dokumentasi terkait AMI untuk merujuk ke agen AWS PCS versi 1.0.1-1. Untuk informasi selengkapnya, lihat Pemasang perangkat lunak untuk membangun kustom AMIs untuk AWS PCS dan Langkah 2 - Instal agen AWS PCS .	N/A
Oktober 10, 2024	Menambahkan chapter pemecahan masalah	Menambahkan chapter pemecahan masalah dengan topik tentang instans EC2 yang diganti secara otomatis setelah reboot. Untuk informasi selengkapnya, lihat Memecahkan masalah di Layanan Komputasi AWS Paralel .	N/A

Date	Ubah	Pembaruan dokumentasi	Versi API diperbarui
September 23, 2024	Memperbarui izin minimum untuk menggunakan tindakan API dan untuk administrator layanan	ec2:DescribeInstancesTypeOffering. Izin sekarang diperlukan untuk tindakan CreateComputeNodeGroup dan UpdateComputeNodeGroup API. Untuk informasi selengkapnya, lihat Izin minimum untuk AWS PCS .	N/A
September 5, 2024	Memperbarui contoh kebijakan IAM untuk izin minimum untuk administrator layanan	Untuk informasi selengkapnya, lihat Izin minimum untuk administrator layanan .	N/A
September 5, 2024	Menambahkan izin yang hilang ke JSON di halaman kebijakan terkelola	Ini hanya koreksi pada dokumentasi. Kebijakan terkelola yang sebenarnya tidak diubah. Untuk informasi selengkapnya, lihat AWS kebijakan terkelola untuk Layanan Komputasi AWS Paralel .	N/A
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 Rilis PCS	Rilis awal panduan pengguna AWS PCS.	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.