



AWS ParallelCluster Panduan Pengguna (v3)

AWS ParallelCluster



AWS ParallelCluster: AWS ParallelCluster Panduan Pengguna (v3)

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

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

Table of Contents

Apa itu AWS ParallelCluster	1
Harga	1
Menyiapkan AWS ParallelCluster	2
Menyiapkan Akun AWS	2
Daftar Akun AWS	2
Membuat pengguna administratif	3
Membuat pasangan kunci	4
Memasang AWS ParallelCluster CLI	4
Instal AWS ParallelCluster di lingkungan virtual (disarankan)	5
Menginstal AWS ParallelCluster di lingkungan non-virtual menggunakan pip	6
Instal AWS ParallelCluster sebagai aplikasi mandiri	8
Langkah-langkah yang harus diambil setelah instalasi	9
Menginstal AWS ParallelCluster UI	10
Instal AWS ParallelCluster UI	10
Buat domain khusus	13
Opsi kumpulan pengguna Amazon Cognito	15
Identifikasi versi AWS ParallelCluster dan AWS ParallelCluster UI	18
Perbarui AWS ParallelCluster UI ke AWS ParallelCluster versi baru	19
AWS ParallelClusterBiaya UI	19
Memulai	19
Konfigurasi dan buat cluster dengan AWS ParallelCluster CLI	20
Konfigurasi dan buat cluster dengan AWS ParallelCluster UI	30
Terhubung ke klaster	31
Beberapa akses pengguna ke cluster	32
Buat Direktori Aktif	33
Membuat klaster dengan domain AD	34
Masuk ke klaster yang terintegrasi dengan domain AD	37
Lowongan kerja Running MPI	38
Contoh AWS Managed Microsoft AD konfigurasi klaster LDAP (S)	38
Praktik terbaik	42
Praktik terbaik: pemilihan jenis instance node kepala	42
Praktik terbaik: kinerja jaringan	43
Praktik terbaik: peringatan anggaran	45
Praktik terbaik: memindahkan klaster ke versi AWS ParallelCluster minor atau patch baru	45

Pindah dari AWS ParallelCluster 2.x ke 3.x	46
Kustom Bootstrap Tindakan	46
AWS ParallelCluster2.x dan 3.x menggunakan sintaks file konfigurasi yang berbeda	47
Bahasa inklusif	53
Dukungan Scheduler	53
CLI AWS ParallelCluster	53
Pembaruan konfigurasi IMDS	56
Wilayah yang Didukung untuk AWS ParallelCluster	56
Menggunakan AWS ParallelCluster	59
AWS ParallelClusterUI	59
AWS LambdaKonfigurasi VPC diAWS ParallelCluster	61
AWS Identity and Access Management izin di AWS ParallelCluster	63
AWS ParallelCluster Peran instans EC2	64
AWS ParallelCluster contoh kebijakan <code>pccluster</code> pengguna	64
AWS ParallelCluster kebijakan contoh pengguna untuk mengelola sumber daya IAM	79
AWS ParallelCluster parameter konfigurasi untuk mengelola izin IAM	85
Konfigurasi jaringan	100
AWS ParallelClusterdi subnet publik	102
AWS ParallelClustermenggunakan dua subnet	104
AWS ParallelClusterdalam satu subnet pribadi yang terhubung menggunakanAWS Direct Connect	105
AWS ParallelClusterdenganAWS Batch scheduler	106
AWS ParallelClusterdalam subnet tunggal tanpa akses internet	108
Node masuk	114
Tindakan bootstrap kustom	117
Konfigurasi	120
Argumen	124
Contoh cluster dengan tindakan bootstrap kustom	124
Contoh memperbarui skrip bootstrap kustom untuk IMDSv2	126
Contoh memperbarui konfigurasi untuk IMDSv1	126
Bekerja dengan Amazon S3	127
Contoh	127
Mengerjakan Instans ITSpot	128
Skenario 1: Instans Spot tanpa pekerjaan berjalan terganggu	129
Skenario 2: Instance Spot yang menjalankan pekerjaan node tunggal terganggu	129
Skenario 3: Instance Spot yang menjalankan pekerjaan multi-node terganggu	129

Penjadwal didukung oleh AWS ParallelCluster	130
Slurm Workload Manager	130
AWS Batch	185
Penyimpanan bersama	193
Konfigurasi penyimpanan bersama	196
Bekerja dengan penyimpanan bersama	199
Kuota	203
Penandaan	204
Pemantauan AWS ParallelCluster dan log	207
Integrasi dengan Amazon CloudWatch Logs	208
CloudWatchDasbor Amazon	211
CloudWatchAlarm Amazon untuk metrik kluster	213
AWS ParallelClusterrotasi log yang dikonfigurasi	216
pcclusterLog CLI	217
Log keluaran konsol EC2	218
AmbilAWS ParallelCluster UI danAWS ParallelCluster runtime log	219
Mengambil dan melestarikan log	221
AWS CloudFormation sumber daya khusus	224
Tumpukan penyedia dihosting oleh AWS ParallelCluster	225
Sumber daya cluster	226
Operasi cluster	230
Memecahkan masalah tumpukan yang menyertakan sumber daya kustom AWS ParallelCluster	230
Elastic Fabric Adapter	231
Aktifkan Intel MPI	232
API AWS ParallelCluster	233
AWS ParallelClusterDokumentasi API	233
Menyebarkan dengan AWS CLI	234
Memperbarui API	237
Memanggil API AWS ParallelCluster	238
Mengakses log dan metrik API	240
Connect ke node kepala melalui NICE DCV	241
Sertifikat HTTPS NICE DCV	241
Lisensi NICE DCV	241
Menggunakan pcluster update-cluster	242
Kebijakan Pembaruan: definisi	242

<code>pcluster update-cluster</code> contoh	245
AWS ParallelCluster Kustomisasi AMI	248
AWS ParallelCluster Pertimbangan kustomisasi AMI	249
Lakukan tes validasi komponen kustom	249
Pantau proses Image Builder dengan <code>pcluster</code> perintah untuk membantu debugging	250
Pertimbangan lainnya	250
Meluncurkan instans dengan OCR (Cadangan Kapasitas Sesuai Permintaan)	251
Menggunakan ODCR dengan AWS ParallelCluster	251
Penambalan AMI dan penggantian instans EC2	260
Pembaruan atau penggantian instans simpul kepala	261
Simpan data dari drive fana	262
Menghentikan dan memulai node kepala kluster	262
Sistem Operasi)	264
Pertimbangan Sistem Operasi	264
Referensi untuk AWS ParallelCluster	267
AWS ParallelClusterversi 3 perintah CLI	267
<code>pcluster</code>	268
<code>pcluster3-config-converter</code>	311
File konfigurasi	312
File konfigurasi cluster	313
Membangun file konfigurasi gambar	444
AWS ParallelClusterReferensi API	452
<code>BuildImage</code>	453
<code>createCluster</code>	458
<code>HapusCluster</code>	464
<code>deleteClusterInstances</code>	467
<code>DeletelImage</code>	469
<code>DescribCluster</code>	472
<code>describeClusterInstances</code>	480
<code>describeComputeFleet</code>	484
<code>DescribelImage</code>	485
<code>getClusterLogAcara</code>	492
<code>getClusterStackAcara</code>	496
<code>getImageLogAcara</code>	500
<code>getImageStackAcara</code>	504
<code>ListClusters</code>	508

listClusterLogAliran	512
listImageLogAliran	516
ListImages	520
listOfficialImages	523
UpdateCluster	526
updateComputeFleet	532
AWS ParallelClusterAPI pustaka Python	535
AWS ParallelClusterOtorisasi perpustakaan Python	535
InstalAWS ParallelClusterPustaka Python	535
Operasi API cluster	536
Hitung operasi API armada	539
Operasi cluster dan stack log	542
Operasi API gambar	544
Operasi log gambar dan tumpukan	547
Contoh	550
AWS LambdauntukAWS ParallelClusterPustaka Python	551
Bagaimana AWS ParallelCluster berhasil	553
AWS ParallelCluster proses	553
clustermgtd	553
clusterstatusmgtd	554
computemgtd	554
AWS layanan yang digunakan oleh AWS ParallelCluster	555
Amazon API Gateway	556
AWS Batch	556
AWS CloudFormation	556
Amazon CloudWatch	557
CloudWatch Acara Amazon	557
CloudWatch Log Amazon	557
AWS CodeBuild	557
Amazon DynamoDB	558
Amazon Elastic Block Store	558
Amazon Elastic Compute Cloud	558
Amazon Elastic Container Registry	559
Amazon EFS	559
Amazon FSx for Lustre	559
Amazon FSx untuk ONTAP NetApp	559

Amazon FSx for OpenZFS	560
AWS Identity and Access Management	560
AWS Lambda	560
Amazon RDS	560
Amazon Route 53	561
Amazon Simple Notification Service	561
Amazon Simple Storage Service	561
Amazon VPC	561
Elastic Fabric Adapter	562
EC2 Image Builder	562
NICE DCV	562
AWS ParallelClusterDirektori internal	562
Tutorial	564
Menjalankan pekerjaan pertama Anda diAWS ParallelCluster	564
Memverifikasi instalasi Anda	565
Membuat cluster pertama Anda	565
Masuk ke node kepala Anda	566
Menjalankan pekerjaan pertama Anda menggunakan Slurm	567
Membangun kebiasaanAWS ParallelClusterAMI	568
Cara menyesuaikanAWS ParallelClusterAMI	569
Bangun kustomAWS ParallelClusterAMI	569
MemodifikasiAWS ParallelClusterAMI	576
Mengintegrasikan Active Directory	579
Mengonfigurasi enkripsi penyimpanan bersama dengan kunci AWS KMS	610
Buat kebijakan	611
Mengonfigurasi dan membuat cluster	612
Menjalankan pekerjaan dalam kluster mode antrian ganda	614
Konfigurasi kluster Anda	614
Buat cluster Anda	616
Masuk ke node kepala	617
Jalankan pekerjaan dalam beberapa mode antrian	617
Menggunakan API AWS ParallelCluster ini	621
Membuat cluster dengan akuntansi Slurm	636
Langkah 1: Buat VPC dan subnet untuk AWS ParallelCluster	637
Langkah 2: Buat tumpukan database	637
Langkah 3: Buat cluster dengan Slurm akuntansi diaktifkan	638

Mengembalikan ke versi dokumen Manajer AWS Sistem sebelumnya	638
Kembalikan ke versi dokumen SSM sebelumnya	639
Membuat cluster dengan AWS CloudFormation	641
Pembuatan cluster dengan CloudFormation tumpukan cepat	642
Pembuatan cluster dengan AWS CloudFormation Command Line Interface (CLI)	644
Lihat keluaran CloudFormation kluster	646
Akses kluster Anda	647
Hapus	647
AWS ParallelClusterIntegrasi UI dengan Pusat Identitas	648
Aktifkan Pusat Identitas IAM	648
Menambahkan Aplikasi Anda ke Pusat Identitas IAM	651
Memecahkan Masalah AWS ParallelCluster	659
Mencoba membuat cluster	660
failureCodeadalah OnNodeConfiguredExecutionFailure	660
failureCodeadalah OnNodeConfiguredDownloadFailure	660
failureCodeadalah OnNodeConfiguredFailure	661
failureCodeadalah OnNodeStartExecutionFailure	661
failureCodeadalah OnNodeStartDownloadFailure	662
failureCodeadalah OnNodeStartFailure	662
failureCodeadalah EbsMountFailure	662
failureCodeadalah EfsMountFailure	663
failureCodeadalah FsxMountFailure	663
failureCodeadalah RaidMountFailure	663
failureCodeadalah AmiVersionMismatch	664
failureCodeadalah InvalidAmi	664
failureCodeadalah HeadNodeBootstrapFailure dengan failureReason Gagal mengatur simpul kepala.	664
failureCodeHeadNodeBootstrapFailuredengan waktu pembuatan failureReason Cluster habis.	665
failureCodeHeadNodeBootstrapFailuredengan failureReason Gagal mem- bootstrap simpul kepala.	666
failureCodeadalah ResourceCreationFailure	666
failureCodeadalah ClusterCreationFailure	667
Melihat WaitCondition timed out... di CloudFormation tumpukan	667
Melihat Resource creation cancelled di CloudFormation tumpukan	667

Melihat Failed to run cfn-init... atau kesalahan lain di AWS CloudFormation tumpukan	667
Melihat chef-client.log berakhir dengan INFO: Waiting for static fleet capacity provisioning	667
Melihat Failed to run preinstall or postinstall in cfn-init.log	668
Melihat This AMI was created with xxx, but is trying to be used with xxx... di CloudFormation tumpukan	668
Melihat This AMI was not baked by AWS ParallelCluster... di CloudFormation tumpukan	668
Melihat pcluster create-cluster perintah gagal dijalankan secara lokal	668
Dukungan tambahan	668
Mencoba menjalankan pekerjaan	669
srun pekerjaan interaktif gagal dengan kesalahan srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf	669
Job terjebak dalam CF keadaan dengan squeue perintah	669
Menjalankan pekerjaan skala besar dan melihat nfsd: too many open connections, consider increasing the number of threads in /var/log/messages	670
Menjalankan pekerjaan MPI	670
Mencoba memperbarui cluster	671
pcluster update-cluster perintah gagal dijalankan secara lokal	671
Melihat clusterStatus adalah UPDATE_FAILED dengan pcluster describe-cluster perintah	671
Waktu pembaruan cluster habis	671
Mencoba mengakses penyimpanan	672
Menggunakan sistem file Amazon FSx for Lustre eksternal	672
Menggunakan sistem file Amazon Elastic File System eksternal	672
Mencoba menghapus cluster	672
pcluster delete-cluster Perintah gagal dijalankan secara lokal	672
Tumpukan cluster gagal dihapus	672
Mencoba memutakhirkan tumpukan AWS ParallelCluster API	672
Melihat kesalahan dalam inisialisasi node komputasi	673
Melihat Node bootstrap error di clustermgtd.log	673
Saya mengonfigurasi reservasi kapasitas permintaan (ODCR) atau Instans Cadangan zona	673

Melihat An error occurred (VcpuLimitExceeded)slurm_resume.log ketika saya gagal menjalankan pekerjaan, atau masukclustermgtd.log, ketika saya gagal membuat cluster	674
Melihat An error occurred (InsufficientInstanceCapacity)slurm_resume.log ketika saya gagal menjalankan pekerjaan, atau masukclustermgtd.log, ketika saya gagal membuat cluster	675
Melihat node dalam DOWN keadaan dengan Reason (Code:InsufficientInstanceCapacity)... ..	675
Melihat cannot change locale (en_US.utf-8) because it has an invalid name di slurm_resume.log	675
Tak satu pun dari skenario sebelumnya berlaku untuk situasi saya	676
Pemecahan masalah metrik metrik metrik metrik metrik Kondisi Kondisi	676
Melihat grafik Kesalahan Penyediaan Instans	676
Melihat grafik Kesalahan Instans Tidak Sehat	678
Melihat grafik Compute Fleet Idle Time	680
Memecahkan masalah penerapan klaster	681
Lihat AWS CloudFormation acara di CREATE_FAILED	682
Gunakan CLI untuk melihat aliran log	684
Buat ulang cluster yang gagal dengan rollback-on-failure	686
Memecahkan masalah penskalaan	687
Log kunci untuk debugging	688
Melihat InsufficientInstanceCapacity kesalahan slurm_resume.log ketika saya gagal menjalankan pekerjaan, atau clustermgtd.log ketika saya gagal membuat cluster	675
Memecahkan masalah inialisasi node	690
Memecahkan masalah penggantian dan penghentian node yang tidak terduga	693
Mengganti, mengakhiri, atau mematikan instance dan node yang bermasalah	694
Status antrian (partisi) Inactive	695
Memecahkan masalah node dan pekerjaan lain yang diketahui	695
Grup penempatan dan masalah peluncuran instance	695
Direktori yang tidak dapat diganti	696
Memecahkan masalah di NICE DCV	696
Log untuk NICE DCV	696
Masalah Ubuntu NICE DCV	696
Memecahkan masalah dalam cluster dengan integrasi AWS Batch	697

Masalah simpul kepala	697
Masalah komputasi	698
Kegagalan Job	698
Connect timeout pada kesalahan URL endpoint	698
Memecahkan masalah integrasi multi-pengguna dengan Active Directory	698
Pemecahan masalah khusus Active Directory	699
Aktifkan mode debug	700
Cara pindah dari LDAPS ke LDAP	700
Cara menonaktifkan verifikasi sertifikat server LDAPS	701
Cara masuk dengan kunci SSH daripada kata sandi	701
Cara mengatur ulang kata sandi pengguna dan kata sandi yang kedaluwarsa	701
Cara memverifikasi domain yang bergabung	702
Cara memecahkan masalah dengan sertifikat	702
Cara memverifikasi bahwa integrasi dengan Active Directory berfungsi	705
Cara memecahkan masalah masuk untuk menghitung node	705
Masalah yang diketahui dengan pekerjaan SimCenter StarCCM+ di lingkungan multi- pengguna	706
Masalah yang diketahui dengan resolusi nama pengguna	706
Cara mengatasi masalah pembuatan direktori home	707
Memecahkan masalah AMI khusus	708
Memecahkan masalah batas waktu pembaruan klaster saat cfn-hup tidak berjalan	709
Pemecahan masalah jaringan	709
Cluster dalam satu masalah subnet publik	709
Pembaruan cluster gagal pada tindakan onNodeUpdated kustom	710
Melihat kesalahan dengan Slurm konfigurasi khusus	710
Alarm Cluster	710
Dukungan tambahan	711
AWS ParallelCluster kebijakan dukungan	712
Keamanan	713
Informasi keamanan untuk layanan yang digunakan oleh AWS ParallelCluster	714
Perlindungan data	714
Enkripsi data	715
Lihat juga	716
Identity and Access Management	717
Validasi kepatuhan	718
Menegakkan TLS 1.2	718

Tentukan Protokol yang Didukung Saat Ini	719
Kompilasi OpenSSL dan Python	720
Catatan rilis dan riwayat dokumen	722
.....	dcccxi

Apa itu AWS ParallelCluster

AWS ParallelCluster adalah alat manajemen cluster open source yang AWS didukung yang membantu Anda untuk menyebarkan dan mengelola komputasi kinerja tinggi (HPC) cluster di AWS Cloud Secara otomatis mengatur sumber daya komputasi yang diperlukan, scheduler, dan filesystem bersama. Anda dapat menggunakan AWS ParallelCluster dengan AWS Batch dan Slurm penjadwal.

Dengan AWS ParallelCluster, Anda dapat dengan cepat membangun dan menyebarkan bukti konsep dan lingkungan komputasi HPC produksi. Anda juga dapat membangun dan menerapkan alur kerja tingkat tinggi di atasnya AWS ParallelCluster, seperti portal genomik yang mengotomatiskan seluruh alur kerja sekuensing DNA.

Anda dapat mengakses AWS ParallelCluster dengan menggunakan metode berikut:

- [AWS ParallelCluster antarmuka baris perintah \(CLI\)](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster UI](#) (ditambahkan dengan rilis 3.5.0)
- [AWS ParallelCluster API pustaka Python](#) (ditambahkan dengan rilis 3.5.0)
- Sebagai [AWS CloudFormation sumber daya khusus](#) (ditambahkan dengan rilis 3.6.0)

Harga

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Layanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelCluster UI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Biaya UI](#).

Menyiapkan AWS ParallelCluster

Topik

- [Menyiapkan Akun AWS](#)
- [Membuat pasangan kunci](#)
- [Menginstal antarmuka baris AWS ParallelCluster perintah \(CLI\)](#)
- [Langkah-langkah yang harus diambil setelah instalasi](#)
- [Menginstal AWS ParallelCluster UI](#)
- [Memulai dengan AWS ParallelCluster](#)
- [Beberapa akses pengguna ke cluster](#)
- [Praktik terbaik](#)
- [Pindah dari AWS ParallelCluster 2.x ke 3.x](#)
- [Wilayah yang Didukung untuk AWS ParallelCluster](#)

Menyiapkan Akun AWS

Siapkan AWS akun untuk digunakan AWS ParallelCluster.

Daftar Akun AWS

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

Untuk mendaftar Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.
2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar Akun AWS, Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan hanya gunakan pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

Membuat pengguna 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.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Pada halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

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

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

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

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna administratif.

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 administratif

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

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

Membuat pasangan kunci

Untuk menyebarkan cluster, AWS ParallelCluster meluncurkan instance EC2 untuk membuat node kepala cluster dan node komputasi. Untuk melakukan tugas kluster, seperti menjalankan dan memantau pekerjaan, atau mengelola pengguna, Anda harus dapat mengakses node kepala cluster. Untuk memverifikasi bahwa Anda dapat mengakses instance head node menggunakan SSH, Anda harus menggunakan key pair EC2. Untuk mempelajari cara membuat key pair, lihat [Membuat key pair](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Menginstal antarmuka baris AWS ParallelCluster perintah (CLI)

AWS ParallelCluster didistribusikan sebagai paket Python dan diinstal menggunakan manajer paket pip Python. Untuk petunjuk tentang cara menginstal paket Python, lihat [Menginstal paket di Panduan Pengguna](#) Kemasan Python.

Cara menginstal AWS ParallelCluster:

- [Instal AWS ParallelCluster di lingkungan virtual \(disarankan\)](#)
- [Menginstal AWS ParallelCluster di lingkungan non-virtual menggunakan pip](#)
- [Instal AWS ParallelCluster sebagai aplikasi mandiri](#)

Anda dapat menemukan nomor versi CLI terbaru di [halaman rilis](#) di GitHub. Dalam panduan ini, contoh perintah mengasumsikan bahwa Anda telah menginstal versi Python yang lebih lambat dari versi 3.6. Contoh pip perintah menggunakan pip3 versi.

Kelola AWS ParallelCluster 2 dan AWS ParallelCluster 3

Untuk pelanggan yang menggunakan AWS ParallelCluster 2 dan AWS ParallelCluster 3 dan ingin mengelola CLI untuk kedua paket, kami sarankan Anda menginstal AWS ParallelCluster 2 dan AWS ParallelCluster 3 di [lingkungan virtual](#) yang berbeda. Ini memastikan bahwa Anda dapat terus menggunakan setiap versi AWS ParallelCluster dan sumber daya cluster yang terkait.

Instal AWS ParallelCluster di lingkungan virtual (disarankan)

Kami menyarankan Anda menginstal AWS ParallelCluster di lingkungan virtual untuk menghindari konflik versi persyaratan dengan pip paket lain.

Prasyarat

- AWS ParallelCluster membutuhkan Python 3.7 atau yang lebih baru. Jika Anda belum menginstalnya, [unduh versi yang kompatibel](#) untuk platform Anda di [python.org](#).

Untuk menginstal AWS ParallelCluster di lingkungan virtual

1. Jika `virtualenv` tidak diinstal, instal `virtualenv` menggunakan `pip3`. Jika `python3 -m virtualenv help` menampilkan informasi bantuan, lanjutkan ke langkah 2.

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

Jalankan `exit` untuk meninggalkan jendela terminal saat ini dan buka jendela terminal baru untuk mengambil perubahan pada lingkungan.

2. Buat lingkungan virtual dan beri nama.

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

Atau, Anda dapat menggunakan `-p` opsi untuk menentukan versi Python tertentu.

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

3. Aktifkan lingkungan virtual baru Anda.

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

4. Instal AWS ParallelCluster ke lingkungan virtual Anda.

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

5. Instal Node Version Manager dan versi Long-Term Support (LTS) Node.js terbaru. AWS Cloud Development Kit (AWS CDK) (AWS CDK) membutuhkan Node.js CloudFormation untuk pembuatan template.

Note

Jika instalasi Node.js Anda tidak berfungsi pada platform Anda, Anda dapat menginstal versi LTS sebelum versi LTS terbaru. Untuk informasi selengkapnya, lihat [jadwal rilis Node.js](#) dan prasyarat [AWS CDK](#).

Contoh perintah instalasi Node.js:

```
$ nvm install --lts=Hydrogen
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

6. Verifikasi bahwa AWS ParallelCluster sudah terpasang dengan benar.

```
$ pcluster version
{
  "version": "3.7.0"
}
```

Anda dapat menggunakan perintah `deactivate` untuk keluar dari lingkungan virtual. Setiap kali Anda memulai sesi, Anda harus [mengaktifkan kembali lingkungan](#).

Untuk meng-upgrade ke versi terbaru AWS ParallelCluster, jalankan perintah instalasi lagi.

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

Menginstal AWS ParallelCluster di lingkungan non-virtual menggunakan pip

Prasyarat

- AWS ParallelCluster membutuhkan Python 3.7 atau yang lebih baru. Jika Anda belum menginstalnya, [unduh versi yang kompatibel](#) untuk platform Anda di [python.org](#).

Instal AWS ParallelCluster

1. Gunakan pip untuk menginstal AWS ParallelCluster.

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

Saat Anda menggunakan `--user` sakelar, pip instal AWS ParallelCluster ke `~/.local/bin`.

2. Instal Node Version Manager dan versi Long-Term Support (LTS) Node.js terbaru. AWS Cloud Development Kit (AWS CDK) (AWS CDK) membutuhkan Node.js CloudFormation untuk pembuatan template.

Note

Jika instalasi Node.js Anda tidak berfungsi pada platform Anda, Anda dapat menginstal versi LTS sebelum versi LTS terbaru. Untuk informasi selengkapnya, lihat [jadwal rilis Node.js](#) dan prasyarat [AWS CDK](#).

```
$ nvm install --lts=Gallium
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

3. Verifikasi yang AWS ParallelCluster diinstal dengan benar.

```
$ pcluster version
{
  "version": "3.7.0"
}
```

4. Untuk memperbarui ke versi terbaru, jalankan perintah pemasangan lagi.

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

Instal AWS ParallelCluster sebagai aplikasi mandiri

Instal AWS ParallelCluster sebagai aplikasi mandiri di lingkungan Anda. Ikuti instruksi untuk menginstal AWS ParallelCluster pada OS yang tersedia di bagian berikut.

Prasyarat

- Lingkungan dengan sistem operasi yang kompatibel dengan versi installer yang tersedia.

Note

AWS ParallelCluster membutuhkan NodeJS. AWS ParallelCluster Installer menyertakan versi bundel NodeJS (v18), yang diinstal jika belum ada. Jika sistem Anda tidak kompatibel dengan NodeJS v18, Anda harus menginstal NodeJS sebelum menginstal. AWS ParallelCluster

Linux

Linux x86 (64-bit)

Instal AWS ParallelCluster di lingkungan Anda.

1. Unduh [penginstal pcluster](#) terbaru.
2. Buka zip bundel penginstal dan instal AWS ParallelCluster dengan menggunakan perintah berikut:

```
$ unzip pcluster-installer-bundle-3.6.1.209-node-v16.19.0-Linux_x86_64-signed.zip
-d pcluster-installer-bundle
$ cd pcluster-installer-bundle
$ chmod +x install_pcluster.sh
```

3. Jalankan skrip instalasi berikut.

```
$ bash install_pcluster.sh
```

4. Verifikasi bahwa AWS ParallelCluster sudah terpasang dengan benar.

```
$ pcluster version
{
```

```
"version": "3.7.0"  
}
```

Memecahkan masalah kesalahan instalasi **pcluster**

- Jika AWS ParallelCluster versi tidak dikembalikan pada langkah 4, restart terminal atau `source bash_profile` untuk memperbarui PATH variabel untuk menyertakan direktori biner baru seperti yang ditunjukkan dalam contoh berikut:

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

- Jika Anda menggunakan `pcluster` instalasi untuk membuat kluster dengan CustomActions ditentukan sebagai sumber daya HTTPS, bukan URI S3, Anda mungkin melihat WARNING pesan yang menunjukkan bahwa sumber daya ini mungkin tidak diverifikasi ([SSL : CERTIFICATE_VERIFY_FAILED]). Ini disebabkan oleh masalah yang diketahui dan Anda dapat mengabaikan peringatan ini jika Anda mempercayai keaslian sumber daya yang ditentukan.

Versi bundel installer sebelumnya

- Tidak ada

Langkah-langkah yang harus diambil setelah instalasi

Anda dapat memverifikasi bahwa AWS ParallelCluster diinstal dengan benar dengan menjalankan [pcluster version](#).

```
$ pcluster version  
{  
  "version": "3.7.0"  
}
```

AWS ParallelCluster diperbarui secara berkala. Untuk memperbarui ke versi terbaru AWS ParallelCluster, jalankan perintah instalasi lagi. Untuk informasi selengkapnya tentang versi terbaru AWS ParallelCluster, lihat [catatan AWS ParallelCluster rilis](#).

```
$ pip3 install aws-parallelcluster --upgrade --user
```

Untuk menghapus instalasi AWS ParallelCluster, gunakan `pip3 uninstall`.

```
$ pip3 uninstall aws-parallelcluster
```

Jika Anda tidak memiliki Python dan `pip3`, gunakan prosedur untuk lingkungan Anda.

Menginstal AWS ParallelCluster UI

AWS ParallelClusterUI adalah antarmuka pengguna berbasis web yang mencerminkan AWS ParallelCluster `pccluster` CLI, sambil memberikan pengalaman seperti konsol. Anda menginstal dan mengakses AWS ParallelCluster UI di Akun AWS. Saat Anda menjalankannya, AWS ParallelCluster UI mengakses instance API yang dihosting di Amazon AWS ParallelCluster API Gateway di situs Akun AWS. Untuk informasi selengkapnya tentang AWS ParallelCluster UI, lihat [AWS ParallelClusterUI](#).

Prasyarat:

- Sebuah Akun AWS
- Akses ke AWS Management Console

Topik

- [Instal AWS ParallelCluster UI](#)
- [Buat domain khusus](#)
- [Opsi kumpulan pengguna Amazon Cognito](#)
- [Identifikasi versi AWS ParallelCluster dan AWS ParallelCluster UI](#)
- [Perbarui AWS ParallelCluster UI ke AWS ParallelCluster versi baru](#)
- [AWS ParallelCluster Biaya UI](#)

Instal AWS ParallelCluster UI

Untuk menginstal instance AWS ParallelCluster UI, Anda memilih tautan buat AWS CloudFormation cepat untuk tempat Wilayah AWS Anda membuat kluster. URL quick-create membawa Anda ke Create Stack Wizard di mana Anda menyediakan input template tumpukan cepat dan menyebarkan tumpukan. Untuk informasi selengkapnya tentang tumpukan CloudFormation buat cepat, lihat [Membuat tautan buat cepat untuk tumpukan](#) di Panduan Pengguna. AWS CloudFormation

Note

Anda hanya dapat membuat dan mengedit cluster atau membuat gambar dengan AWS ParallelCluster versi yang sama yang Anda gunakan untuk menginstal AWS ParallelCluster UI.

AWS ParallelClusterUI cepat membuat tautan menurut Wilayah

UI cepat membuat tautan

[kami-timur-1](#)

[kami-timur-2](#)

[kami-barat-1](#)

[kami-barat-2](#)

[eu-barat-1](#)

[eu-barat-2](#)

[eu-barat-3](#)

[eu-sentral-1](#)

[eu-utara-1](#)

[saya-selatan-1](#)

[sa-timur-1](#)

[ca-pusat-1](#)

[ap-timur laut-1](#)

[ap-timur laut-2](#)

[ap-selatan-1](#)

UI cepat membuat tautan

[ap-tenggara 1](#)

[ap-tenggara 2](#)

[us-gov-west-1](#)

Gunakan tautan AWS CloudFormation buat cepat untuk menerapkan tumpukan AWS ParallelCluster UI dengan tumpukan Amazon Cognito, API Gateway, dan Amazon EC2 Systems Manager bersarang.

1. Masuk ke AWS Management Console.
2. Terapkan AWS ParallelCluster UI dengan memilih tautan Wilayah AWS buat cepat dari tabel di awal bagian ini. Ini akan membawa Anda ke CloudFormation Create Stack Wizard di konsol.
3. Masukkan alamat email yang valid untuk Email Admin.

Setelah penerapan berhasil diselesaikan, AWS ParallelCluster UI akan mengirimkan kata sandi sementara ke alamat email ini. Anda menggunakan kata sandi sementara untuk mengakses AWS ParallelCluster UI. Jika Anda menghapus email sebelum menyimpan atau menggunakan kata sandi sementara, Anda harus menghapus tumpukan dan menginstal ulang AWS ParallelCluster UI.

4. Biarkan sisa formulir kosong atau masukkan nilai untuk parameter (opsional) untuk menyesuaikan build AWS ParallelCluster UI.
5. Perhatikan nama tumpukan untuk digunakan di langkah selanjutnya.
6. Arahkan ke Kemampuan. Setuju dengan CloudFormation kemampuan.
7. Pilih Buat. Dibutuhkan sekitar 15 menit untuk menyelesaikan penerapan AWS ParallelCluster API dan AWS ParallelCluster UI.
8. Lihat detail tumpukan saat tumpukan dibuat.
9. Setelah penerapan selesai, buka email admin yang dikirim ke alamat yang Anda masukkan. Ini berisi kata sandi sementara yang Anda gunakan untuk mengakses AWS ParallelCluster UI. Jika Anda menghapus email secara permanen dan belum masuk ke AWS ParallelCluster UI, Anda harus menghapus tumpukan AWS ParallelCluster UI yang Anda buat dan instal ulang AWS ParallelCluster UI.

10. Di daftar tumpukan AWS CloudFormation konsol, pilih tautan ke nama tumpukan yang Anda catat di langkah sebelumnya.
11. Di Stack details, pilih Outputs dan pilih link untuk kunci bernama **Stackname** URL untuk membuka UI. AWS ParallelCluster **Stackname** adalah nama yang Anda catat di langkah sebelumnya.
12. Masukkan kata sandi sementara. Ikuti langkah-langkah untuk membuat kata sandi Anda sendiri dan masuk.
13. Anda sekarang berada di halaman beranda AWS ParallelCluster UI di Wilayah AWS yang Anda pilih.
14. Untuk mulai menggunakan AWS ParallelCluster UI, lihat [Konfigurasi dan buat cluster dengan AWS ParallelCluster UI](#).

Note

Sesi PCUI memiliki durasi default 5 menit, yang merupakan nilai minimum yang diberikan oleh Cognito pada PCUI 2023.12.0. Oleh karena itu, diharapkan pengguna yang dihapus dari Cognito User Pools masih dapat mengakses sistem hingga sesi berakhir.

Buat domain khusus

Pelajari cara membuat domain khusus untuk AWS ParallelCluster UI. UI di-host di Amazon API Gateway di AndaAkun AWS. Anda dapat membuat domain khusus di konsol API Gateway.

Prasyarat:

- Anda memiliki Akun AWS.
- Anda memiliki instance AWS ParallelCluster UI yang dapat Anda akses.
- Anda memiliki domain.
- Anda dapat mengubah pengaturan dasar Sistem Nama Domain (DNS).

Langkah 1: Buat domain baru di Amazon API Gateway

1. Di bagian AWS Management Console, navigasikan ke [API Gateway](#) di mana Anda dapat melihat API AWS ParallelCluster UI Anda terdaftar.

2. Di panel navigasi, pilih Custom domain names (Nama domain khusus).
3. Pilih Buat.
4. Dalam detail Domain, masukkan nama domain Anda.
5. Dalam konfigurasi Endpoint, pilih sertifikat ACM yang ada, atau pilih Buat Sertifikat ACM baru.

(Opsional) Buat sertifikat

- a. Di konsol ACM, pilih Permintaan.
- b. Dalam nama Domain, masukkan nama domain Anda.
- c. Dalam metode Validasi, pilih metode validasi.

Jika Anda memilih Validasi email, email akan dikirim ke alamat email yang ada di file dengan registrar domain.

- d. Pilih Saya menyetujui untuk mengaktifkan sertifikat.

Langkah 2: Siapkan pemetaan API

1. Di [API Gateway](#), Nama domain khusus your-domain-name, pilih Konfigurasi pemetaan API.
2. Pilih Nama domain khusus.
3. Pilih Tambahkan pemetaan baru.
4. Pilih AWS ParallelCluster UI API, \$default Stage, dan Save.
5. Di nama domain API Gateway, salin nilai untuk digunakan pada langkah selanjutnya.

Langkah 3: Siapkan DNS

- Buat aturan DNS CNAME yang mengarahkan domain Anda ke domain API Gateway. Masukkan hanya domain. Misalnya, jangan tambahkan panggung, seperti beta atau prod. Ganti *abcde12345* dengan API Gateway API ID Anda, dan ganti *us-east-2* dengan API. Wilayah AWS

Aturan	Sumber	Tujuan
CNAME	<i>example.com</i>	d- <i>abcde12345</i> .eksekusi api. <i>us-timur-2</i> .amazonaws.com

Langkah 4: Tambahkan domain ke kumpulan pengguna Amazon Cognito Anda

1. Arahkan ke konsol [Amazon Cognito](#).
2. Pilih tautan kumpulan pengguna Anda.
3. Pilih Integrasi aplikasi.
4. Di Domain, pilih Tindakan, Buat domain kustom.
5. Masukkan domain Kustom Anda dan pilih sertifikat ACM Anda.
6. Pilih Buat domain khusus.

Langkah 5: Konfigurasikan URL callback API Gateway

1. Arahkan ke konsol [Amazon Cognito](#).
2. Di kumpulan pengguna Amazon Cognito Integrasi aplikasi, klien Aplikasi, dan analitik, pilih tautan aplikasi.
3. Di UI yang Dihosting, pilih Edit.
4. Di URL callback yang diizinkan, pilih Tambahkan URL lain dan masukkan URL panggilan balik seperti `example.com/login`

Langkah 6: Konfigurasikan fungsi Lambda

1. Arahkan ke konsol [Lambda](#).
2. Di panel navigasi, pilih Fungsi.
3. Filter daftar fungsi untuk menemukan `ParallelClusterUIFunction` dan memilih tautan.
4. Pilih Konfigurasi, variabel Lingkungan.
5. Pilih Edit.
6. Untuk `SITE_URL` nilainya, masukkan domain kustom Anda.
7. Arahkan ke domain Anda, seperti `example.com`, dan autentikasi untuk terhubung ke AWS ParallelCluster UI.

Opsi kumpulan pengguna Amazon Cognito

Bagian berikut mengacu pada tautan CloudFormation buat cepat atau URL yang dibuat cepat. URL quick-create membawa Anda ke Create Stack Wizard di mana Anda menyediakan input template

tumpukan cepat dan menyebarkan tumpukan. Untuk informasi selengkapnya tentang tumpukan CloudFormation buat cepat, lihat [Membuat tautan buat cepat untuk tumpukan](#) di Panduan Pengguna. AWS CloudFormation

Untuk mempertahankan kumpulan pengguna Amazon Cognito yang dapat Anda gunakan dengan beberapa instance AWS ParallelCluster UI, pertimbangkan opsi berikut:

- Gunakan instance AWS ParallelCluster UI yang sudah ada yang menautkan ke kumpulan pengguna Amazon Cognito yang dibuat dari tumpukan bersarang CloudFormation . Inilah yang dibuat saat Anda menerapkan AWS ParallelCluster UI dengan menggunakan tautan buat cepat dan Anda membiarkan semua parameter Amazon Cognito kosong.
- Gunakan kumpulan pengguna Amazon Cognito mandiri yang diterapkan sebelum UI diterapkan AWS ParallelCluster. Kemudian, terapkan instance AWS ParallelCluster UI baru yang ditautkan ke kumpulan pengguna Amazon Cognito mandiri yang telah Anda terapkan. Dengan cara ini, Anda memisahkan penerapan Amazon Cognito dari penerapan UI. AWS ParallelCluster Selain itu, CloudFormation tumpukan AWS ParallelCluster UI yang tidak bersarang lebih mudah diperbarui.

Menggunakan kumpulan pengguna Amazon Cognito yang ada dengan instance UI baru AWS ParallelCluster

1. Di CloudFormation konsol, pilih tumpukan AWS ParallelCluster UI yang berisi kumpulan pengguna Amazon Cognito yang ingin Anda gunakan dengan beberapa instance AWS ParallelCluster UI.
2. Arahkan ke tumpukan bersarang yang membuat userpool Amazon Cognito.
3. Pilih tab Outputs.
4. Salin nilai parameter berikut:
 - UserPoolId
 - UserPoolAuthDomain
 - SNSRole
5. Terapkan instance AWS ParallelCluster UI baru dengan menggunakan tautan buat cepat, dan isi semua External AWS ParallelCluster UI Amazon Cognito parameter dengan output yang Anda salin. Ini mencegah tumpukan AWS ParallelCluster UI baru membuat kumpulan baru dan menautkannya ke kumpulan pengguna Amazon Cognito yang ada yang dibuat dari tumpukan bersarang. Anda dapat menerapkan instance AWS ParallelCluster UI

baru berikutnya yang memiliki nilai parameter yang sama, dan Anda dapat menautkannya ke kumpulan pengguna Amazon Cognito.

Buat userpool Amazon Cognito mandiri

AWS ParallelClusterUI Amazon Cognito membuat tautan cepat berdasarkan Wilayah

UI Amazon Cognito membuat tautan cepat

[kami-timur-1](#)

[kami-timur-2](#)

[kami-barat-1](#)

[kami-barat-2](#)

[eu-barat-1](#)

[eu-barat-2](#)

[eu-barat-3](#)

[eu-sentral-1](#)

[eu-utara-1](#)

[saya-selatan-1](#)

[sa-timur-1](#)

[ca-pusat-1](#)

[ap-timur laut-1](#)

[ap-timur laut-2](#)

[ap-selatan-1](#)

[ap-tenggara 1](#)

UI Amazon Cognito membuat tautan cepat

[ap-tenggara 2](#)

[us-gov-west-1](#)

1. Luncurkan tumpukan khusus Amazon Cognito dengan memilih tautan buat cepat berlabel sama dengan saat Anda menerapkan instance Wilayah AWS UI. AWS ParallelCluster Lihat tautan buat cepat di awal bagian ini.
2. Setelah pembuatan tumpukan selesai, pilih tab Output dan salin nilai parameter berikut:
 - `UserPoolId`
 - `UserPoolAuthDomain`
 - `SNSRole`
3. Menerapkan instance AWS ParallelCluster UI baru dengan memilih tautan mulai cepat AWS ParallelCluster UI dan mengisi semua `External AWS ParallelCluster UI Amazon Cognito` parameter dengan nilai yang Anda salin. Instans AWS ParallelCluster UI baru menautkan ke kumpulan pengguna Amazon Cognito mandiri dan tidak membuat tumpukan bersarang atau kumpulan pengguna baru. Anda dapat menerapkan instance AWS ParallelCluster UI baru berikutnya yang memiliki nilai parameter yang sama, dan Anda dapat menautkannya ke kumpulan pengguna Amazon Cognito mandiri.

Identifikasi versi AWS ParallelCluster dan AWS ParallelCluster UI

1. Di CloudFormation konsol, pilih tumpukan AWS ParallelCluster UI.
2. Pilih tab Parameter.
3. `AWS ParallelClusterVersi` adalah nilai parameter `Versi`.
4. Versi AWS ParallelCluster UI ada di akhir `PublicEcrImageUri` nilai. Misalnya, jika nilainya `public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02`, maka versinya adalah `2023.02`.

Perbarui AWS ParallelCluster UI ke AWS ParallelCluster versi baru

Untuk memperbarui AWS ParallelCluster UI ke AWS ParallelCluster versi terbaru, luncurkan tumpukan baru dengan memilih tautan [buat cepat](#).

AWS ParallelCluster Biaya UI

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Tabel berikut Layanan AWS mencantumkan tempat AWS ParallelCluster UI bergantung dan batas tingkat bebasnya. Penggunaan tipikal diperkirakan biaya kurang dari satu dolar setiap bulan.

Layanan	AWSTingkat Gratis
Amazon Cognito	50.000 pengguna aktif bulanan
Amazon API Gateway	1 juta panggilan API sisanya
AWS Lambda	1 juta permintaan gratis setiap bulan dan 400.000 GB-detik waktu komputasi setiap bulan
EC2 Image Builder	Tanpa biaya, kecuali EC2
Amazon Elastic Compute Cloud	15 menit pembuatan gambar kontainer satu kali
AWS CloudFormation	Data 5 GB (konsumsi, penyimpanan arsip, dan data yang dipindai oleh kueri Logs Insights)

Memulai dengan AWS ParallelCluster

Mulailah dengan mengkonfigurasi dan membuat cluster dengan menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau antarmuka pengguna berbasis web (UI). AWS ParallelCluster UI ditambahkan dalam rilis 3.5.0.

Topik

- [Konfigurasi dan buat cluster dengan antarmuka baris AWS ParallelCluster perintah](#)
- [Konfigurasi dan buat cluster dengan AWS ParallelCluster UI](#)
- [Terhubung ke klaster](#)

Konfigurasi dan buat cluster dengan antarmuka baris AWS ParallelCluster perintah

Setelah Anda menginstal AWS ParallelCluster, selesaikan langkah-langkah konfigurasi berikut.

Verifikasi bahwa AWS Akun Anda memiliki peran yang mencakup izin yang diperlukan untuk menjalankan `pcluster` CLI. Untuk informasi selengkapnya, lihat [AWS ParallelCluster contoh kebijakan pcluster pengguna](#).

Siapkan AWS kredensi Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi AWS CLI](#) dalam panduan AWS CLI pengguna.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [us-east-1]: us-east-1
Default output format [None]:
```

Wilayah AWS Tempat cluster diluncurkan harus memiliki setidaknya satu key pair Amazon EC2. Untuk informasi selengkapnya, lihat [Pasangan kunci Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI), Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS layanan yang digunakan oleh AWS ParallelCluster](#).

Konfigurasi dan buat cluster pertama Anda

Buat cluster pertama Anda dengan menggunakan perintah `pcluster configure` CLI untuk memulai wizard yang meminta Anda untuk semua informasi yang diperlukan untuk mengkonfigurasi dan membuat cluster Anda. Rincian urutan berbeda saat menggunakan AWS Batch sebagai penjadwal dibandingkan dengan menggunakan Slurm.

Slurm

```
$ pcluster configure --config config-file.yaml
```

Dari daftar Wilayah AWS pengidentifikasi yang valid, pilih Wilayah AWS tempat yang Anda inginkan untuk menjalankan kluster Anda.

Note

Daftar yang Wilayah AWS ditampilkan didasarkan pada partisi akun Anda, dan hanya mencakup Wilayah AWS yang diaktifkan untuk akun Anda. Untuk informasi selengkapnya tentang mengaktifkan Wilayah AWS akun Anda, lihat [Mengelola Wilayah AWS](#) di Referensi Umum AWS Contoh yang ditampilkan adalah dari partisi AWS Global. Jika akun Anda ada di AWS GovCloud (US) partisi, hanya Wilayah AWS di partisi yang terdaftar (gov-us-east-1 dan gov-us-west-1). Demikian pula, jika akun Anda berada di partisi AWS China, hanya cn-north-1 dan cn-northwest-1 ditampilkan. Untuk daftar lengkap yang Wilayah AWS didukung oleh AWS ParallelCluster, lihat [Wilayah yang Didukung untuk AWS ParallelCluster](#).

Allowed values for Wilayah AWS ID:

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

Wilayah AWS ID [ap-northeast-1]:

Key pair dipilih dari pasangan kunci yang terdaftar dengan Amazon EC2 di yang dipilih. Wilayah AWS Pilih key pair:

Allowed values for EC2 Key Pair Name:

```
1. your-key-1
2. your-key-2
EC2 Key Pair Name [your-key-1]:
```

Pilih penjadwal yang akan digunakan dengan cluster Anda.

```
Allowed values for Scheduler:
1. slurm
2. awsbatch
Scheduler [slurm]:
```

Pilih sistem operasi.

```
Allowed values for Operating System:
1. alinux2
2. centos7
3. ubuntu2204
4. ubuntu2004
5. rhel8
Operating System [alinux2]:
```

Pilih jenis instance node kepala:

```
Head node instance type [t2.micro]:
```

Pilih konfigurasi antrian. Catatan: Jenis instans tidak dapat ditentukan untuk beberapa sumber daya komputasi dalam antrean yang sama.

```
Number of queues [1]:
Name of queue 1 [queue1]:
Number of compute resources for queue1 [1]: 2
Compute instance type for compute resource 1 in queue1 [t2.micro]:
Maximum instance count [10]:
```

Aktifkan EFA untuk menjalankan aplikasi yang memerlukan komunikasi antar-instans tingkat tinggi dalam skala tanpa AWS biaya tambahan:

- Pilih jenis instans yang [mendukung Elastic Fabric Adapter \(EFA\)](#).
- Aktifkan [EFA](#).

- Tentukan nama [Grup Penempatan](#) yang ada. Jika Anda membiarkannya kosong, AWS ParallelCluster buatlah satu untuk Anda.

```
Compute instance type for compute resource 2 in queue1 [t2.micro]: c5n.18xlarge
Enable EFA on c5n.18xlarge (y/n) [y]: y
Maximum instance count [10]:
Placement Group name []:
```

Setelah langkah-langkah sebelumnya selesai, putuskan apakah akan menggunakan VPC yang ada atau biarkan AWS ParallelCluster membuat VPC untuk Anda. Jika Anda tidak memiliki VPC yang dikonfigurasi dengan benar, AWS ParallelCluster dapat membuat yang baru untuk Anda. Ini baik menempatkan node head dan compute di subnet publik yang sama, atau hanya node kepala di subnet publik dengan semua node komputasi dalam subnet pribadi. Jika Anda membiarkan AWS ParallelCluster membuat VPC, Anda harus memutuskan apakah semua node berada di subnet publik. Untuk informasi selengkapnya, lihat [Konfigurasi jaringan](#).

Jika Anda mengonfigurasi kluster untuk menggunakan tipe instans yang memiliki beberapa antarmuka jaringan atau kartu jaringan, lihat [Konfigurasi jaringan](#) persyaratan jaringan tambahan.

Anda dapat mencapai kuota Anda untuk jumlah VPC yang diizinkan dalam a. Wilayah AWS Kuota default adalah lima VPC untuk a. Wilayah AWS Untuk informasi selengkapnya tentang kuota ini dan cara meminta peningkatan, lihat [VPC dan subnet](#) di Panduan Pengguna Amazon VPC.

Important

VPC yang dibuat oleh AWS ParallelCluster tidak mengaktifkan VPC Flow Logs secara default. VPC Flow Logs memungkinkan Anda untuk menangkap informasi tentang lalu lintas IP yang menuju dan dari antarmuka jaringan di VPC Anda. Untuk informasi selengkapnya, lihat [Log Alur VPC](#) di Panduan Pengguna Amazon VPC.

Jika Anda membiarkan AWS ParallelCluster membuat VPC, pastikan Anda memutuskan apakah semua node akan berada di subnet publik.

Note

Jika Anda memilih 1. Head node in a public subnet and compute fleet in a private subnet, AWS ParallelCluster buat gateway NAT yang menghasilkan biaya tambahan, bahkan jika Anda menentukan sumber daya tingkat gratis.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
subnet]: 1
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Jika Anda tidak membuat VPC baru, Anda harus memilih VPC yang ada.

Jika Anda memilih untuk AWS ParallelCluster membuat VPC, catat ID VPC sehingga Anda dapat menggunakan AWS CLI untuk menghapusnya nanti.

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

Setelah VPC dipilih, putuskan apakah akan menggunakan subnet yang ada atau membuat yang baru.

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

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

AWS Batch

```
$ pcluster configure --config config-file.yaml
```

Dari daftar Wilayah AWS pengidentifikasi yang valid, pilih Wilayah AWS tempat yang Anda inginkan untuk menjalankan kluster Anda.

Note

Daftar yang Wilayah AWS ditampilkan didasarkan pada partisi akun Anda. Ini hanya mencakup Wilayah AWS yang diaktifkan untuk akun Anda. Untuk informasi selengkapnya tentang mengaktifkan Wilayah AWS akun Anda, lihat [Mengelola Wilayah AWS](#) di Referensi Umum AWS Contoh yang ditampilkan adalah dari partisi AWS Global. Jika akun Anda ada di AWS GovCloud (US) partisi, hanya Wilayah AWS di partisi yang terdaftar (gov-us-east-1 dan gov-us-west-1). Demikian pula, jika akun Anda berada di partisi AWS China, hanya cn-north-1 dan cn-northwest-1 ditampilkan. Untuk daftar lengkap yang Wilayah AWS didukung oleh AWS ParallelCluster, lihat [Wilayah yang Didukung untuk AWS ParallelCluster](#).

Allowed values for Wilayah AWS ID:

1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ca-central-1
9. eu-central-1
10. eu-north-1
11. eu-south-1
12. eu-west-1
13. eu-west-2

```
14. eu-west-3
15. me-south-1
16. sa-east-1
17. us-east-1
18. us-east-2
19. us-west-1
20. us-west-2
Wilayah AWS ID [us-east-1]:
```

Key pair dipilih dari pasangan kunci yang terdaftar dengan Amazon EC2 di yang dipilih. Wilayah AWS Pilih key pair:

```
Allowed values for EC2 Key Pair Name:
1. your-key-1
2. your-key-2
EC2 Key Pair Name [your-key-1]:
```

Pilih penjadwal yang akan digunakan dengan cluster Anda.

```
Allowed values for Scheduler:
1. slurm
2. awsbatch
Scheduler [slurm]: 2
```

Kapan awsbatch dipilih sebagai penjadwal, alinux2 digunakan sebagai sistem operasi. Jenis instance head node dimasukkan:

```
Head node instance type [t2.micro]:
```

Pilih konfigurasi antrian. AWS Batch Penjadwal hanya berisi satu antrian. Ukuran maksimum cluster node komputasi dimasukkan. Ini diukur dalam vCPUs.

```
Number of queues [1]:
Name of queue 1 [queue1]:
Maximum vCPU [10]:
```

Putuskan apakah akan menggunakan VPC yang ada atau biarkan AWS ParallelCluster membuat VPC untuk Anda. Jika Anda tidak memiliki VPC yang dikonfigurasi dengan benar, AWS ParallelCluster dapat membuat yang baru. Ini baik menggunakan node head dan compute di subnet publik yang sama, atau hanya node kepala di subnet publik dengan semua node di

subnet pribadi. Anda dapat mencapai kuota Anda pada jumlah VPC yang diizinkan di suatu Wilayah. Jumlah default VPC adalah lima. Untuk informasi selengkapnya tentang kuota ini dan cara meminta peningkatan, lihat [VPC dan subnet](#) di Panduan Pengguna Amazon VPC.

Important

VPC yang dibuat oleh AWS ParallelCluster tidak mengaktifkan VPC Flow Logs secara default. VPC Flow Logs memungkinkan Anda untuk menangkap informasi tentang lalu lintas IP yang menuju dan dari antarmuka jaringan di VPC Anda. Untuk informasi selengkapnya, lihat [Log Alur VPC](#) di Panduan Pengguna Amazon VPC.

Jika Anda membiarkan AWS ParallelCluster membuat VPC, pastikan Anda memutuskan apakah semua node akan berada di subnet publik.

Note

Jika Anda memilih 1. Head node in a public subnet and compute fleet in a private subnet, AWS ParallelCluster buat gateway NAT yang menghasilkan biaya tambahan, bahkan jika Anda menentukan sumber daya tingkat gratis.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
subnet]: *1*
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Jika Anda tidak membuat VPC baru, Anda harus memilih VPC yang ada.

Jika Anda memilih untuk AWS ParallelCluster membuat VPC, catat ID VPC sehingga Anda dapat menggunakan AWS CLI atau AWS Management Console menghapusnya nanti.

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

Setelah VPC dipilih, pastikan Anda memutuskan apakah akan menggunakan subnet yang ada atau membuat yang baru.

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

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

Ketika Anda telah menyelesaikan langkah-langkah sebelumnya, sebuah cluster sederhana diluncurkan ke VPC. VPC menggunakan subnet yang ada yang mendukung alamat IP publik. Tabel rute untuk subnet adalah $0.0.0.0/0 \Rightarrow igw-xxxxxx$. Perhatikan kondisi berikut:

- VPC harus memiliki DNS Resolution = yes dan DNS Hostnames = yes
- VPC juga harus memiliki opsi DHCP dengan benar domain-name untuk. Wilayah AWS DHCP Option Set default sudah menentukan yang diperlukan. AmazonProvidedDNS Jika menentukan lebih dari satu server nama domain, lihat [set opsi DHCP di Panduan Pengguna Amazon VPC](#). Saat menggunakan subnet pribadi, gunakan gateway NAT atau proxy internal untuk mengaktifkan akses web untuk node komputasi. Untuk informasi selengkapnya, lihat [Konfigurasi jaringan](#).

Ketika semua pengaturan berisi nilai yang valid, Anda dapat meluncurkan cluster dengan menjalankan perintah create.

```
$ pcluster create-cluster --cluster-name test-cluster --cluster-configuration cluster-config.yaml
{
  "cluster": {
    "clusterName": "test-cluster",
```

```
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/test-cluster/
abcdef0-f678-890a-5abc-021345abcdef",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  },
  "validationMessages": []
}
```

Ikuti kemajuan klaster:

```
$ pcluster describe-cluster --cluster-name test-cluster
```

atau

```
$ pcluster list-clusters --query 'clusters[?clusterName==`test-cluster`]'
```

Setelah klaster mencapai "clusterStatus": "CREATE_COMPLETE" status, Anda dapat menghubungkannya dengan menggunakan pengaturan klien SSH normal Anda. Untuk informasi selengkapnya tentang menghubungkan ke instans Amazon EC2, lihat Panduan Pengguna [EC2 di Panduan Pengguna](#) Amazon EC2 untuk Instans Linux. Atau Anda dapat menghubungkan cluster melalui

```
$ pcluster ssh --cluster-name test-cluster -i ~/path/to/keyfile.pem
```

Untuk menghapus cluster, jalankan perintah berikut.

```
$ pcluster delete-cluster --region us-east-1 --cluster-name test-cluster
```

Setelah cluster dihapus, Anda dapat menghapus sumber daya jaringan di VPC dengan menghapus tumpukan jaringan. CloudFormation Nama tumpukan dimulai dengan "parallelclusternetworking-" dan berisi waktu pembuatan dalam format "YYYYMMDDHHMMSS". Anda dapat membuat daftar tumpukan menggunakan [list-stacks](#) perintah.

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

```
"parallelclusternetworking-pubpriv-20191029205804"
```

Tumpukan dapat dihapus menggunakan [delete-stack](#) perintah.

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

VPC yang dibuat untuk [pcluster configure](#) Anda tidak dibuat di tumpukan CloudFormation jaringan. Anda dapat menghapus VPC itu secara manual di konsol atau dengan menggunakan file. AWS CLI

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

Konfigurasi dan buat cluster dengan AWS ParallelCluster UI

AWS ParallelCluster UI adalah antarmuka pengguna berbasis web yang mencerminkan AWS ParallelCluster `pcluster` CLI, sambil memberikan pengalaman seperti konsol. Anda menginstal dan mengakses AWS ParallelCluster UI di Akun AWS. Saat Anda menjalankannya, AWS ParallelCluster UI mengakses instance API yang dihosting di Amazon AWS ParallelCluster API Gateway di situs Anda Akun AWS.

Note

Wizard AWS ParallelCluster UI mungkin tidak memiliki opsi UI untuk semua fitur yang didukung dalam AWS ParallelCluster versi terbaru yang didukung. Anda dapat mengedit file konfigurasi secara manual sesuai kebutuhan atau menggunakan AWS ParallelCluster CLI.

Di bagian ini, kami memandu Anda melalui konfigurasi dan pembuatan cluster dengan menggunakan AWS ParallelCluster UI.

Prasyarat:

- Akses ke instance AWS ParallelCluster UI yang sedang berjalan. Untuk informasi selengkapnya, lihat [Menginstal AWS ParallelCluster UI](#).

Konfigurasi dan buat cluster

1. Dalam tampilan AWS ParallelCluster UI Clusters, pilih Create cluster, Step by step.

2. Di Cluster, Name, masukkan nama untuk cluster Anda.
3. Pilih VPC dengan subnet publik untuk klaster Anda, dan pilih Berikutnya.
4. Di Head node, pilih Add SSM session, dan pilih Next.
5. Dalam Antrian, Menghitung sumber daya, pilih 1 untuk node Statis.
6. Untuk jenis Instance, hapus jenis instans default yang dipilih, pilih t2.micro, dan pilih Berikutnya.
7. Di Penyimpanan, pilih Berikutnya.
8. Dalam konfigurasi Cluster, tinjau konfigurasi cluster YAMM dan pilih Dry run untuk memvalidasinya.
9. Pilih Buat untuk membuat klaster Anda, berdasarkan konfigurasi yang divalidasi.
10. Setelah beberapa detik, AWS ParallelCluster UI secara otomatis menavigasi Anda kembali ke Cluster, di mana Anda dapat memantau status pembuatan klaster dan peristiwa Stack.
11. Pilih Detail untuk melihat detail klaster, seperti versi dan status.
12. Pilih Instans untuk melihat daftar instans dan status EC2.
13. Pilih Stack events untuk melihat peristiwa tumpukan cluster, dan AWS Management Console link ke CloudFormation stack yang membuat cluster.
14. Di Detail, setelah pembuatan klaster selesai, pilih Lihat YAMG untuk melihat atau mengunduh file YAMM konfigurasi cluster.
15. Setelah pembuatan cluster selesai, pilih Shell untuk mengakses node kepala cluster.

Note

Saat Anda memilih Shell, AWS ParallelCluster buka sesi Amazon EC2 Systems Manager dan menambahkan `ssm-user` ke `/etc/sudoers` Untuk informasi selengkapnya, lihat [Mengaktifkan atau menonaktifkan izin administratif `ssm-user` akun](#) di Panduan Pengguna Amazon EC2 Systems Manager.

16. Untuk membersihkan, dalam tampilan Clusters, pilih cluster, dan pilih Actions, Delete cluster.

Terhubung ke klaster

Saat menggunakan AWS ParallelCluster, Anda dapat terhubung ke node kepala klaster untuk menjalankan pekerjaan, melihat hasil, mengelola pengguna, dan memantau klaster dan status pekerjaan. Hubungkan ke contoh node kepala cluster dengan menggunakan metode berikut:

- Masuk ssh dengan menggunakan [pasangan kunci](#). Tentukan kunci pribadi di [HeadNode/KeyName](#) dalam konfigurasi cluster. Untuk informasi selengkapnya, lihat [Menyambungkan ke instans Linux Anda menggunakan SSH](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Masuk dengan menggunakan perintah `pcluster ssh` command line interface (CLI). Tentukan kunci pribadi dalam konfigurasi cluster [HeadNode/KeyName](#). Untuk informasi selengkapnya, lihat [pcluster ssh](#).
- Hubungkan ke node kepala cluster dengan menggunakan sesi SSM. Anda harus menambahkan kebijakan AmazonSSManagedInstanceCore terkelola ke [HeadNode/AdditionalIamPolicies](#) dalam konfigurasi klaster untuk terhubung dengan menggunakan sesi SSM. Untuk informasi selengkapnya, lihat [manajer sesi SSM](#) di Panduan Pengguna SSM.
- Hubungkan ke node kepala cluster dengan menggunakan NICE DCV. Untuk informasi selengkapnya, lihat [Connect ke node kepala melalui NICE DCV](#).
- Saat Anda menggunakan AWS ParallelCluster UI, Anda juga dapat terhubung ke node kepala klaster dengan menggunakan perintah EC2 Connect yang disediakan UI.

Beberapa akses pengguna ke cluster

Pelajari cara menerapkan dan mengelola beberapa akses pengguna ke satu cluster.

Dalam topik ini, AWS ParallelCluster pengguna mengacu pada pengguna sistem untuk instance komputasi. Contohnya adalah `ec2-user` untuk instance AWS EC2.

AWS ParallelCluster dukungan akses multi-pengguna tersedia di semua Wilayah AWS tempat yang saat AWS ParallelCluster ini tersedia. Ia bekerja dengan yang lain Layanan AWS, termasuk [Amazon FSx for Lustre](#) dan [Amazon Elastic File System](#).

Anda dapat menggunakan [AWS Directory Service for Microsoft Active Directory](#) atau [Simple AD](#) untuk mengelola akses klaster. Pastikan untuk memeriksa [Wilayah AWSketersediaan](#) untuk layanan ini. Untuk menyiapkan cluster, tentukan [AWS ParallelCluster DirectoryService](#) konfigurasi. AWS Directory Service direktori dapat dihubungkan ke beberapa cluster. Hal ini memungkinkan manajemen identitas terpusat di berbagai lingkungan dan pengalaman login terpadu.

Saat Anda menggunakan AWS Directory Service untuk AWS ParallelCluster beberapa akses pengguna, Anda dapat masuk ke klaster dengan kredensyal pengguna yang ditentukan dalam

direktori. Kredensi ini terdiri dari nama pengguna dan kata sandi. Setelah Anda masuk ke cluster untuk pertama kalinya, kunci SSH pengguna dibuat secara otomatis. Anda dapat menggunakannya untuk masuk tanpa kata sandi.

Anda dapat membuat, menghapus, dan memodifikasi pengguna atau grup kluster setelah layanan direktori Anda diterapkan. Dengan AWS Directory Service, Anda dapat melakukan ini di AWS Management Console atau dengan menggunakan alat Pengguna dan Komputer Direktori Aktif. Alat ini dapat diakses dari instans EC2 apa pun yang bergabung dengan Active Directory Anda. Untuk informasi selengkapnya, lihat [Menginstal alat administrasi Direktori Aktif](#).

Jika Anda berencana untuk menggunakan AWS ParallelCluster dalam satu subnet tanpa akses internet, lihat [AWS ParallelCluster dalam subnet tunggal tanpa akses internet](#) persyaratan tambahan.

Topik

- [Buat Direktori Aktif](#)
- [Membuat kluster dengan domain AD](#)
- [Masuk ke kluster yang terintegrasi dengan domain AD](#)
- [Lowongan kerja Running MPI](#)
- [Contoh AWS Managed Microsoft AD konfigurasi kluster LDAP \(S\)](#)

Buat Direktori Aktif

Pastikan Anda membuat Active Directory (AD) sebelum membuat kluster. Untuk informasi tentang cara memilih jenis direktori aktif untuk kluster Anda, lihat [Yang harus dipilih](#) dalam Panduan AWS Directory Service Administrasi.

Jika direktori kosong, tambahkan pengguna dengan nama pengguna dan kata sandi. Untuk informasi selengkapnya, lihat dokumentasi yang khusus untuk [AWS Directory Service for Microsoft Active Directory](#) atau [Simple AD](#).

Note

AWS ParallelCluster mengharuskan setiap direktori pengguna Active Directory berada di `/home/$user` direktori.

Membuat kluster dengan domain AD

Warning

Bagian pengantar ini menjelaskan cara mengatur AWS ParallelCluster dengan server Managed Active Directory (AD) melalui Lightweight Directory Access Protocol (LDAP). LDAP adalah protokol yang tidak aman. Untuk sistem produksi, kami sangat merekomendasikan penggunaan sertifikat TLS (LDAPS) seperti yang dijelaskan di [Contoh AWS Managed Microsoft AD konfigurasi kluster LDAP \(S\)](#) bagian berikut.

Konfigurasi cluster Anda untuk diintegrasikan dengan direktori dengan menentukan informasi yang relevan di `DirectoryService` bagian file konfigurasi cluster. Untuk informasi selengkapnya, lihat bagian [DirectoryService](#) konfigurasi.

Anda dapat menggunakan contoh berikut ini untuk mengintegrasikan cluster Anda dengan AWS Managed Microsoft AD over Lightweight Directory Access Protocol (LDAP).

Definisi spesifik yang diperlukan untuk konfigurasi AWS Managed Microsoft AD over LDAP:

- Anda harus mengatur `ldap_auth_disable_tls_never_use_in_production` parameter ke `True` bawah [DirectoryService/AdditionalSssdConfigs](#).
- Anda dapat menentukan nama host pengontrol atau alamat IP untuk [DirectoryService/DomainAddr](#).
- [DirectoryService/DomainReadOnlyUsers](#) sintaks harus sebagai berikut:

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Dapatkan data AWS Managed Microsoft AD konfigurasi Anda:

```
$ aws ds describe-directories --directory-id "d-abcdef01234567890"
```

```
{
  "DirectoryDescriptions": [
    {
      "DirectoryId": "d-abcdef01234567890",
      "Name": "corp.example.com",
    }
  ]
}
```

```

    "DnsIpAddrs": [
      "203.0.113.225",
      "192.0.2.254"
    ],
    "VpcSettings": {
      "VpcId": "vpc-021345abcdef6789",
      "SubnetIds": [
        "subnet-1234567890abcdef0",
        "subnet-abcdef01234567890"
      ],
      "AvailabilityZones": [
        "region-idb",
        "region-idd"
      ]
    }
  }
]
}

```

Konfigurasi cluster untuk AWS Managed Microsoft AD:

```

Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:

```



```

DomainName: dc=corp,dc=example,dc=com
DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
AdditionalSssdConfigs:
  ldap_auth_disable_tls_never_use_in_production: True

```

Untuk menggunakan konfigurasi ini untuk Simple AD, ubah nilai **DomainReadOnlyUser** properti di **DirectoryService** bagian:

```

DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:SimpleAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True

```

Pertimbangan:

- Kami menyarankan Anda menggunakan LDAP melalui TLS/SSL (atau LDAPS) daripada LDAP saja. TLS/SSL memastikan bahwa koneksi dienkripsi.
- Nilai [DomainAddr](#) properti [DirectoryService](#)/cocok dengan entri dalam `DnsIpAddrs` daftar dari `describe-directories` output.
- Kami menyarankan agar kluster Anda menggunakan subnet yang terletak di Availability Zone yang sama dengan yang [DirectoryServiceDomainAddr](#) ditunjuk/. Jika Anda menggunakan [konfigurasi Dynamic Host Configuration Protocol \(DHCP\) kustom](#) yang direkomendasikan untuk VPC direktori dan subnet Anda tidak terletak di [DirectoryService/DomainAddr](#) Availability Zone, lalu lintas antar Availability Zones dimungkinkan. Penggunaan konfigurasi DHCP kustom tidak diperlukan untuk menggunakan fitur integrasi AD multi-pengguna.
- Nilai [DomainReadOnlyUser](#) properti [DirectoryService](#)/menentukan pengguna yang harus dibuat dalam direktori. Pengguna ini tidak dibuat secara default. Kami menyarankan agar Anda tidak memberikan izin pengguna ini untuk memodifikasi data direktori.
- Nilai [PasswordSecretArn](#) properti [DirectoryService](#)/menunjuk ke AWS Secrets Manager rahasia yang berisi kata sandi pengguna yang Anda tentukan untuk [DomainReadOnlyUser](#) properti [DirectoryService](#)/. Jika kata sandi pengguna ini berubah, perbarui nilai rahasia dan perbarui kluster. Untuk memperbarui cluster untuk nilai rahasia baru,

Anda harus menghentikan armada komputasi dengan `pcluster update-compute-fleet` perintah. Jika Anda mengonfigurasi cluster Anda untuk digunakan [LoginNodes](#), hentikan [LoginNodes/Pool](#) dan perbarui cluster setelah menyetel [LoginNodes/Pool/Count](#) ke 0. Kemudian, jalankan perintah berikut dari dalam node kepala cluster.

```
sudo /opt/parallelcluster/scripts/directory_service/  
update_directory_service_password.sh
```

Untuk contoh lain, lihat juga [Mengintegrasikan Active Directory](#).

Masuk ke klaster yang terintegrasi dengan domain AD

Jika Anda mengaktifkan fitur integrasi domain Pengiriman Aktif (AD), autentikasi dengan kata sandi diaktifkan pada node kepala cluster. Direktori home pengguna AD dibuat pada login pengguna pertama ke node kepala atau pertama kali `sudo-user` beralih ke pengguna AD di node kepala.

Otentikasi kata sandi tidak diaktifkan untuk node komputasi cluster. Pengguna AD harus masuk untuk menghitung node dengan kunci SSH.

Secara default, kunci SSH diatur di `/${HOME}/.ssh` direktori pengguna AD pada login SSH pertama ke node kepala. Perilaku ini dapat dinonaktifkan dengan menyetel properti [DirectoryService/GenerateSshKeysForUsers](#) boolean ke `false` dalam konfigurasi cluster. Secara default, [DirectoryService/GenerateSshKeysForUsers](#) diatur ke `true`.

Jika AWS ParallelCluster aplikasi memerlukan SSH tanpa kata sandi di antara node cluster, pastikan bahwa kunci SSH diatur dengan benar di direktori home pengguna.

AWS Managed Microsoft AD kata sandi kedaluwarsa setelah 42 hari. Untuk informasi selengkapnya, lihat [Mengelola kebijakan kata sandi AWS Managed Microsoft AD](#) di Panduan AWS Directory Service Administrasi. Jika kata sandi Anda kedaluwarsa, kata sandi harus diatur ulang untuk memulihkan akses cluster. Untuk informasi selengkapnya, lihat [Cara mengatur ulang kata sandi pengguna dan kata sandi yang kedaluwarsa](#).

Note

Jika fitur integrasi AD tidak berfungsi seperti yang diharapkan, log SSSD dapat memberikan informasi diagnostik yang berguna untuk memecahkan masalah. Log ini terletak di `/var/log/sss` direktori pada node cluster. Secara default, mereka juga disimpan dalam grup CloudWatch log Amazon cluster.

Untuk informasi selengkapnya, lihat [Memecahkan masalah integrasi multi-pengguna dengan Active Directory](#).

Lowongan kerja Running MPI

Seperti yang disarankan di SchedMD, bootstrap MPI menggunakan metode Slurm bootstrapping MPI. Untuk informasi lebih lanjut, lihat dokumentasi [Slurm resmi atau dokumentasi](#) resmi untuk perpustakaan MPI Anda.

Misalnya, dalam [dokumentasi resmi IntelMpi](#), Anda mengetahui bahwa saat menjalankan pekerjaan StarCCM, Anda harus menetapkan Slurm sebagai orkestrator proses dengan mengeksport variabel lingkungan. `I_MPI_HYDRA_BOOTSTRAP=slurm`

Note

Masalah yang diketahui

Dalam kasus di mana aplikasi MPI Anda bergantung pada SSH sebagai mekanisme untuk menelurkan pekerjaan MPI, dimungkinkan untuk menimbulkan [bug yang diketahui di Slurm](#) yang menyebabkan resolusi yang salah dari nama pengguna direktori menjadi “tidak ada”. Konfigurasi aplikasi Anda untuk digunakan Slurm sebagai metode bootstrap MPI atau lihat [Masalah yang diketahui dengan resolusi nama pengguna](#) di bagian Pemecahan Masalah untuk detail lebih lanjut dan kemungkinan solusi.

Contoh AWS Managed Microsoft AD konfigurasi kluster LDAP (S)

AWS ParallelCluster mendukung beberapa akses pengguna dengan mengintegrasikan dengan melalui Lightweight Directory Access Protocol (LDAP), atau LDAP melalui TLS/SSL (LDAPS). AWS Directory Service

Contoh berikut menunjukkan cara membuat konfigurasi cluster untuk diintegrasikan dengan AWS Managed Microsoft AD over LDAP (S).

AWS Managed Microsoft AD melalui LDAPS dengan verifikasi sertifikat

Anda dapat menggunakan contoh ini untuk mengintegrasikan kluster Anda dengan AWS Managed Microsoft AD over LDAPS, dengan verifikasi sertifikat.

Definisi khusus untuk AWS Managed Microsoft AD over LDAPS dengan konfigurasi sertifikat:

- [DirectoryService/LdapTlsReqCert](#) harus diatur ke hard (default) untuk LDAPS dengan verifikasi sertifikat.
- [DirectoryService/LdapTlsCaCert](#) harus menentukan jalur ke sertifikat otoritas (CA) Anda.

Sertifikat CA adalah bundel sertifikat yang berisi sertifikat dari seluruh rantai CA yang mengeluarkan sertifikat untuk pengontrol domain AD.

Sertifikat dan sertifikat CA Anda harus diinstal pada node cluster.

- Nama host pengontrol harus ditentukan untuk [DirectoryService/DomainAddr](#), bukan alamat IP.
- [DirectoryService/DomainReadOnlyUsers](#) sintaks harus sebagai berikut:

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Contoh file konfigurasi cluster untuk menggunakan AD over LDAPS:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
  CustomActions:
    OnNodeConfigured:
      Script: s3://aws-parallelcluster/scripts/pcluster-dub-msad-ldaps.post.sh
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ComputeResources:
      - Name: t2micro
```

```

    InstanceType: t2.micro
    MinCount: 1
    MaxCount: 10
Networking:
  SubnetIds:
    - subnet-abcdef01234567890
Iam:
  AdditionalIamPolicies:
    - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
CustomActions:
  OnNodeConfigured:
    Script: s3://aws-parallelcluster-pcluster/scripts/pcluster-dub-msad-
ldaps.post.sh
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://win-abcdef01234567890.corp.example.com,ldaps://win-
abcdef01234567890.corp.example.com
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsCaCert: /etc/ldap/cacerts/corp.example.com.bundleca.cer
  LdapTlsReqCert: hard

```

Tambahkan sertifikat dan konfigurasi pengontrol domain dalam skrip pasca instal:

```

*#!/bin/bash*
set -e

AD_CERTIFICATE_S3_URI="s3://corp.example.com/bundle/corp.example.com.bundleca.cer"
AD_CERTIFICATE_LOCAL="/etc/ldap/cacerts/corp.example.com.bundleca.cer"

AD_HOSTNAME_1="win-abcdef01234567890.corp.example.com"
AD_IP_1="192.0.2.254"

AD_HOSTNAME_2="win-abcdef01234567890.corp.example.com"
AD_IP_2="203.0.113.225"

# Download CA certificate
mkdir -p $(dirname "${AD_CERTIFICATE_LOCAL}")
aws s3 cp "${AD_CERTIFICATE_S3_URI}" "${AD_CERTIFICATE_LOCAL}"
chmod 644 "${AD_CERTIFICATE_LOCAL}"

# Configure domain controllers reachability

```

```
echo "${AD_IP_1} ${AD_HOSTNAME_1}" >> /etc/hosts
echo "${AD_IP_2} ${AD_HOSTNAME_2}" >> /etc/hosts
```

Anda dapat mengambil nama host pengontrol domain dari instance yang digabungkan ke domain seperti yang ditunjukkan pada contoh berikut.

Dari contoh Windows

```
$ nslookup 192.0.2.254
```

```
Server: corp.example.com
Address: 192.0.2.254

Name: win-abcdef01234567890.corp.example.com
Address: 192.0.2.254
```

Dari contoh Linux

```
$ nslookup 192.0.2.254
```

```
192.0.2.254.in-addr.arpa name = corp.example.com
192.0.2.254.in-addr.arpa name = win-abcdef01234567890.corp.example.com
```

AWS Managed Microsoft AD melalui LDAPS tanpa verifikasi sertifikat

Anda dapat menggunakan contoh ini untuk mengintegrasikan kluster Anda dengan AWS Managed Microsoft AD over LDAPS, tanpa verifikasi sertifikat.

Definisi khusus untuk AWS Managed Microsoft AD over LDAPS tanpa konfigurasi verifikasi sertifikat:

- [DirectoryService/LdapTlsReqCert](#) harus diatur ke `never`.
- Baik nama host pengontrol atau alamat IP dapat ditentukan untuk [DirectoryService/DomainAddr](#).
- [DirectoryService/DomainReadOnlyUser](#) sintaks harus sebagai berikut:

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Contoh file konfigurasi cluster untuk menggunakan AWS Managed Microsoft AD melalui LDAPS tanpa verifikasi sertifikat:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://203.0.113.225,ldaps://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsReqCert: never
```

Praktik terbaik

Praktik terbaik: pemilihan jenis instance node kepala

Meskipun head node tidak menjalankan pekerjaan, fungsinya dan ukurannya sangat penting untuk kinerja keseluruhan cluster. Ketika Anda memilih jenis instans yang akan digunakan untuk node kepala Anda, pertimbangkan karakteristik berikut:

Ukuran cluster: Node kepala mengatur logika penskalaan cluster dan bertanggung jawab untuk melampirkan node baru ke scheduler. Untuk skala atas dan ke bawah cluster yang memiliki sejumlah besar node, memberikan node kepala beberapa kapasitas komputasi tambahan.

Sistem file bersama: Saat Anda menggunakan sistem file bersama, pilih jenis instans dengan bandwidth jaringan yang cukup, dan bandwidth Amazon EBS yang cukup, untuk menangani alur kerja Anda. Pastikan bahwa node kepala mampu mengekspos direktori server NFS yang cukup untuk cluster dan menangani artefak yang perlu dibagi antara node komputasi dan node kepala.

Praktik terbaik: kinerja jaringan

Kinerja jaringan sangat penting untuk aplikasi komputasi kinerja tinggi (HPC). Tanpa kinerja jaringan yang andal, aplikasi ini tidak dapat berfungsi seperti yang diharapkan. Untuk mengoptimalkan kinerja jaringan, pertimbangkan praktik terbaik berikut.

- Grup penempatan: Jika Anda menggunakan Slurm, pertimbangkan untuk mengonfigurasi setiap Slurm antrean untuk menggunakan grup penempatan klaster. Grup penempatan klaster adalah pengelompokan logis instans di dalam Availability Zone. Untuk informasi selengkapnya tentang, lihat [Grup Penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux. Anda dapat menentukan [PlacementGroup](#) di [Networking](#) bagian antrian, setiap sumber daya komputasi ditetapkan ke grup penempatan antrian. Saat menentukan [PlacementGroup](#) di [Networking](#) bagian sumber daya komputasi, sumber daya komputasi tertentu tersebut ditetapkan ke grup penempatan tersebut. Spesifikasi grup penempatan sumber daya komputasi menimpa spesifikasi antrian untuk sumber daya komputasi. Untuk informasi lebih lanjut, lihat [SlurmQueuesNetworking/SlurmQueues/PlacementGroup](#) dan [ComputeResources/Networking/PlacementGroup](#).

```
Networking:
  PlacementGroup:
    Enabled: true
    Id: your-placement-group-name
```

Atau, AWS ParallelCluster buat grup penempatan untuk Anda.

```
Networking:
  PlacementGroup:
    Enabled: true
```


Dimulai dengan AWS ParallelCluster versi 3.3.0, pembuatan dan manajemen grup penempatan dimodifikasi. Bila Anda menentukan grup penempatan yang akan diaktifkan, `tanpaname` atau `Id`, dalam antrian, setiap sumber daya komputasi ditetapkan grup penempatan terkelolanya sendiri, bukan satu grup dikelola untuk seluruh antrian. Ini membantu mengurangi kesalahan kapasitas yang tidak mencukupi. Jika Anda perlu memiliki satu grup penempatan untuk seluruh antrian, Anda dapat menggunakan grup penempatan bernama.

[SlurmQueues/Networking/PlacementGroup/Name](#) ditambahkan sebagai alternatif yang disukai untuk [SlurmQueues/Networking/PlacementGroup/Id](#).

Untuk informasi selengkapnya, lihat [Networking](#).

- Jaringan yang disempurnakan: Pertimbangkan untuk memilih jenis instans yang mendukung jaringan yang disempurnakan. Rekomendasi ini berlaku untuk semua [instance generasi saat ini](#). Untuk informasi selengkapnya tentang, lihat [Jaringan yang disempurnakan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Adaptor Kain Elastis: Untuk mendukung tingkat tinggi instans yang dapat diskalakan ke komunikasi instans, pertimbangkan untuk memilih antarmuka jaringan EFA untuk jaringan Anda. Perangkat keras bypass sistem operasi (OS) meningkatkan instans untuk instans komunikasi instans dengan elastisitas sesuai permintaan dan fleksibilitas AWS Cloud. Anda dapat mengkonfigurasi setiap Slurm antrian [ComputeResource](#) untuk digunakan [Efa](#). Untuk informasi selengkapnya tentang menggunakan EFA AWS ParallelCluster, lihat [Elastic Fabric Adapter](#).

```
ComputeResources:  
  - Name: your-compute-resource-name  
    Efa:  
      Enabled: true
```

Untuk informasi selengkapnya tentang EFA, lihat [Elastic Fabric Adapter](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

- Bandwidth instans: Bandwidth menskalakan dengan ukuran instans. Untuk informasi tentang jenis instans yang [dioptimalkan untuk Amazon EBS](#), lihat [Instans](#) yang Dioptimalkan [Amazon EBS dan jenis volume](#) Amazon EC2 untuk Instans Linux.

Praktik terbaik: peringatan anggaran

Untuk mengelola biaya sumber daya AWS ParallelCluster, sebaiknya gunakan AWS Budgets tindakan untuk membuat anggaran. Anda juga dapat membuat peringatan ambang batas anggaran yang ditentukan untuk AWS sumber daya yang dipilih. Untuk informasi selengkapnya, lihat [Mengonfigurasi tindakan anggaran](#) di Panduan AWS Budgets Pengguna. Demikian pula, Anda juga dapat menggunakan Amazon CloudWatch untuk membuat alarm penagihan. Untuk informasi selengkapnya tentang, lihat [Membuat alarm penagihan untuk memantau estimasi AWS biaya Anda](#).

Praktik terbaik: memindahkan klaster ke versi AWS ParallelCluster minor atau patch baru

Saat ini setiap versi AWS ParallelCluster minor mandiri bersama dengan `pccluster` CLI-nya. Untuk memindahkan klaster ke versi minor atau patch baru, Anda harus membuat ulang klaster menggunakan CLI versi baru.

Untuk mengoptimalkan proses memindahkan klaster ke versi minor atau patch baru, sebaiknya lakukan hal berikut:

- Simpan data pribadi dalam volume eksternal yang dibuat di luar klaster, seperti Amazon EFS dan FSx for Lustre. Dengan melakukan ini, Anda dapat dengan mudah memindahkan data dari satu cluster ke yang lain di future.
- Buat sistem penyimpanan bersama menggunakan jenis berikut. Anda dapat membuat sistem ini menggunakan AWS CLI atau AWS Management Console.
 - [SharedStorage](#) / [EbsSettings](#) / [VolumeId](#)
 - [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)
 - [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)

Mendefinisikan sistem file atau volume dalam konfigurasi cluster sebagai sistem file yang ada atau volume. Dengan cara ini, mereka dipertahankan ketika Anda menghapus klaster dan dapat dilampirkan ke klaster baru.

Sebaiknya gunakan sistem file Amazon EFS atau FSx for Lustre. Kedua sistem ini dapat dilampirkan ke beberapa klaster pada waktu yang sama. Selain itu, Anda dapat melampirkan salah satu dari sistem ini ke klaster baru sebelum Anda menghapus klaster yang ada.

- Gunakan [tindakan bootstrap kustom](#) untuk menyesuaikan instans Anda daripada menggunakan AMI kustom. Jika sebaliknya, Anda menggunakan AMI kustom, maka Anda perlu menghapus dan membuat ulang AMI tersebut untuk setiap rilis versi baru.
- Kami menyarankan Anda menerapkan rekomendasi sebelumnya dalam urutan berikut:
 1. Perbarui konfigurasi kluster yang ada untuk menggunakan definisi sistem file yang ada.
 2. Verifikasi `pcCluster` versi dan perbarui jika diperlukan.
 3. Membuat dan menguji cluster baru. Saat Anda menguji kluster baru, periksa yang berikut ini:
 - Pastikan bahwa data Anda tersedia di kluster baru.
 - Pastikan aplikasi Anda bekerja di kluster baru.
 4. Setelah kluster baru Anda sepenuhnya diuji dan operasional dan Anda tidak lagi membutuhkan cluster yang ada, menghapusnya.

Pindah dari AWS ParallelCluster 2.x ke 3.x

Kustom Bootstrap Tindakan

Dengan AWS ParallelCluster 3, Anda dapat menentukan skrip tindakan bootstrap kustom yang berbeda untuk node kepala dan menghitung node menggunakan `OnNodeStart` (`pre_install` dalam AWS ParallelCluster versi 2) dan `OnNodeConfigured` (`post_install` dalam AWS ParallelCluster versi 2) parameter di [HeadNode](#) dan [Scheduling/0](#) bagian `SlurmQueues`. Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

Script tindakan bootstrap kustom yang dikembangkan untuk AWS ParallelCluster 2 harus disesuaikan untuk digunakan dalam AWS ParallelCluster 3:

- Kami tidak merekomendasikan menggunakan `/etc/parallelcluster/cfnconfig` dan `cfn_node_type` untuk membedakan antara head dan compute node. Sebagai gantinya, kami menyarankan Anda menentukan dua skrip yang berbeda di [HeadNode](#) dan [Scheduling/SlurmQueues](#).
- Jika Anda lebih memilih untuk melanjutkan loading `/etc/parallelcluster/cfnconfig` untuk digunakan dalam script tindakan bootstrap Anda, perhatikan nilai berubah dari "MasterServer" untuk "HeadNode" (lihat: [Bahasa inklusif](#)). `cfn_node_type`
- Pada AWS ParallelCluster 2, argumen masukan pertama untuk skrip aksi bootstrap adalah URL S3 ke skrip dan dicadangkan. Dalam AWS ParallelCluster 3, hanya argumen yang dikonfigurasi dalam konfigurasi yang diteruskan ke skrip.

⚠ Warning

Menggunakan variabel internal yang disediakan melalui `/etc/parallelcluster/cfnconfig` file tidak didukung secara resmi. File ini mungkin dihapus sebagai bagian dari rilis mendatang.

AWS ParallelCluster 2.x dan 3.x menggunakan sintaks file konfigurasi yang berbeda

AWS ParallelCluster konfigurasi 3.x menggunakan sintaks YAKL. Referensi lengkap dapat ditemukan di [File konfigurasi](#).

Selain membutuhkan format file YAKL, sejumlah bagian konfigurasi, pengaturan, dan nilai parameter telah diperbarui di AWS ParallelCluster 3.x. Pada bagian ini, kami mencatat perubahan kunci pada AWS ParallelCluster konfigurasi bersama dengan side-by-side contoh yang menggambarkan perbedaan ini di setiap versi AWS ParallelCluster.

Contoh konfigurasi antrian penjadwal ganda dengan hyperthreading diaktifkan dan dinonaktifkan

AWS ParallelCluster 2:

```
[cluster default]
queue_settings = ht-enabled, ht-disabled
...

[queue ht-enabled]
compute_resource_settings = ht-enabled-i1
disable_hyperthreading = false

[queue ht-disabled]
compute_resource_settings = ht-disabled-i1
disable_hyperthreading = true

[compute_resource ht-enabled-i1]
instance_type = c5n.18xlarge
[compute_resource ht-disabled-i1]
instance_type = c5.xlarge
```

AWS ParallelCluster 3:

```

...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: ht-enabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-enabled-i1
          DisableSimultaneousMultithreading: true
          InstanceType: c5n.18xlarge
    - Name: ht-disabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-disabled-i1
          DisableSimultaneousMultithreading: false
          InstanceType: c5.xlarge

```

Contoh FSx baru untuk konfigurasi file-system Lustre

AWS ParallelCluster2:

```

[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
storage_capacity = 1200
imported_file_chunk_size = 1024
import_path = s3://bucket
export_path = s3://bucket/export_dir
weekly_maintenance_start_time = 3:02:30
deployment_type = PERSISTENT_1
data_compression_type = LZ4

```

AWS ParallelCluster3:

```

...

```

```

SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      ImportedFileChunkSize: 1024
      ImportPath: s3://bucket
      ExportPath: s3://bucket/export_dir
      WeeklyMaintenanceStartTime: "3:02:30"
      DeploymentType: PERSISTENT_1
      DataCompressionType: LZ4

```

Contoh konfigurasi cluster pemasangan FSx yang ada untuk sistem file Lustre

AWS ParallelCluster2:

```

[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
fsx_fs_id = fsx_fs_id

```

AWS ParallelCluster3:

```

...
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      FileSystemId: fsx_fs_id

```

Contoh cluster dengan tumpukan perangkat lunak Intel HPC Platform Specification

AWS ParallelCluster2:

```

[cluster default]
enable_intel_hpc_platform = true
...

```

AWS ParallelCluster3:

```
...
AdditionalPackages:
  IntelSoftware:
    IntelHpcPlatform: true
```

Catatan:

- Pemasangan perangkat lunak Intel HPC Platform Specification tunduk pada syarat dan ketentuan [Perjanjian Lisensi Pengguna Akhir Intel](#) yang berlaku.

Contoh konfigurasi IAM kustom termasuk: profil instans, peran instans, kebijakan tambahan untuk instans, dan peran untuk fungsi lambda yang terkait dengan klaster

AWS ParallelCluster2:

```
[cluster default]
additional_iam_policies = arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess,arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
ec2_iam_role = ec2_iam_role
iam_lambda_role = lambda_iam_role
...
```

AWS ParallelCluster3:

```
...
Iam:
  Roles:
    CustomLambdaResources: lambda_iam_role
HeadNode:
  ...
  Iam:
    InstanceRole: ec2_iam_role
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ...
      Iam:
        InstanceProfile: iam_instance_profile
    - Name: queue2
```

```
...
Iam:
  AdditionalIamPolicies:
    - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
    - Policy: arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
```

Catatan:

- Untuk AWS ParallelCluster 2, pengaturan IAM diterapkan ke semua contoh klaster dan tidak `additional_iam_policies` dapat digunakan bersamaan dengan `ec2_iam_role`
- Untuk AWS ParallelCluster 3, Anda dapat memiliki pengaturan IAM yang berbeda untuk node head dan komputasi dan bahkan menentukan pengaturan IAM yang berbeda untuk setiap antrian komputasi.
- Untuk AWS ParallelCluster 3, Anda dapat menggunakan profil instans IAM sebagai alternatif untuk peran IAM. `InstanceProfile`, `InstanceRole` atau tidak `AdditionalIamPolicies` dapat dikonfigurasi bersama.

Contoh tindakan bootstrap kustom

AWS ParallelCluster2:

```
[cluster default]
s3_read_resource = arn:aws:s3:::bucket_name/*
pre_install = s3://bucket_name/scripts/pre_install.sh
pre_install_args = 'R curl wget'
post_install = s3://bucket_name/scripts/post_install.sh
post_install_args = "R curl wget"
...
```

AWS ParallelCluster3:

```
...
HeadNode:
  ...
  CustomActions:
    OnNodeStart:
      Script: s3://bucket_name/scripts/pre_install.sh
      Args:
        - R
        - curl
        - wget
```



```

OnNodeConfigured:
  Script: s3://bucket_name/scripts/post_install.sh
  Args: ['R', 'curl', 'wget']
Iam:
  S3Access:
    - BucketName: bucket_name
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
  CustomActions:
    OnNodeStart:
      Script: s3://bucket_name/scripts/pre_install.sh
      Args: ['R', 'curl', 'wget']
    OnNodeConfigured:
      Script: s3://bucket_name/scripts/post_install.sh
      Args: ['R', 'curl', 'wget']
Iam:
  S3Access:
    - BucketName: bucket_name

```

Contoh kluster dengan akses baca dan tulis ke sumber daya bucket S3

AWS ParallelCluster2:

```

[cluster default]
s3_read_resource = arn:aws:s3:::bucket/read_only/*
s3_read_write_resource = arn:aws:s3:::bucket/read_and_write/*
...

```

AWS ParallelCluster3:

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: bucket_name
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: bucket_name
        KeyName: read_and_write/

```

```
    EnableWriteAccess: True
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
  Iam:
    S3Access:
      - BucketName: bucket_name
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: bucket_name
        KeyName: read_and_write/
        EnableWriteAccess: True
```

Bahasa inklusif

AWS ParallelCluster3 menggunakan kata-kata “head node” di tempat-tempat di mana “master” digunakan dalam AWS ParallelCluster 2. Ini termasuk yang berikut:

- Variabel yang diekspor di lingkungan AWS Batch kerja berubah: dari MASTER_IP kePCLUSTER_HEAD_NODE_IP.
- Semua AWS CloudFormation output berubah dari Master* keHeadNode*.
- Semua NodeType dan tag berubah dari Master keHeadNode.

Dukungan Scheduler

AWS ParallelCluster3.x tidak mendukung Son of Grid Engine (SGE) dan penjadwal Torsi.

AWS BatchPerintahawsbhosts,awsbkill,awsbout,awsbqueues,awsbstat, dan awsbsub didistribusikan sebagai paket `aws-parallelcluster-awsbatch-cli` PyPI terpisah. Paket ini diinstal oleh AWS ParallelCluster pada node kepala. Anda masih dapat menggunakan AWS Batch perintah ini dari node kepala cluster. Namun, jika Anda ingin menggunakan AWS Batch perintah dari lokasi selain node kepala, Anda harus menginstal paket `aws-parallelcluster-awsbatch-cli` PyPI terlebih dahulu.

CLI AWS ParallelCluster

Antarmuka baris AWS ParallelCluster perintah (CLI) telah diubah. Sintaks baru dijelaskan dalam [AWS ParallelClusterPerintah CLI](#). Format output untuk CLI adalah string [JSON](#).

Mengkonfigurasi klaster baru

`pcluster configure`Perintah ini mencakup parameter yang berbeda dalam AWS ParallelCluster 3 dibandingkan dengan AWS ParallelCluster 2. Untuk informasi selengkapnya, lihat [pcluster configure](#).

Perhatikan juga bahwa sintaks file konfigurasi telah berubah dari AWS ParallelCluster 2. Untuk referensi lengkap pengaturan konfigurasi klaster, lihat [File konfigurasi cluster](#).

Membuat klaster baru

AWS ParallelCluster `pcluster create`Perintah 2 telah digantikan oleh [pcluster create-cluster](#)perintah.

Perhatikan perilaku default di AWS ParallelCluster 2.x, tanpa `-nw` opsi, adalah menunggu peristiwa pembuatan cluster, sementara perintah AWS ParallelCluster 3.x kembali segera. Kemajuan pembuatan cluster dapat dipantau menggunakan [pcluster describe-cluster](#)

File konfigurasi AWS ParallelCluster 3 berisi definisi klaster tunggal, sehingga `-t` parameter tidak diperlukan lagi.

Berikut ini adalah contoh file konfigurasi.

```
# AWS ParallelCluster v2
$ pcluster create \
  -r REGION \
  -c V2_CONFIG_FILE \
  -nw \
  -t CLUSTER_TEMPLATE \
  CLUSTER_NAME

# AWS ParallelCluster v3
$ pcluster create-cluster \
  --region REGION \
  --cluster-configuration V3_CONFIG_FILE \
  --cluster-name CLUSTER_NAME
```

Listing cluster

Perintah `pcluster list` AWS ParallelCluster 2.x harus diganti dengan [pcluster list-clusters](#)perintah.

Catatan: Anda perlu AWS ParallelCluster v2 CLI untuk daftar cluster dibuat dengan versi 2.x. AWS ParallelCluster Lihat [Instal AWS ParallelCluster di lingkungan virtual \(disarankan\)](#) cara menginstal beberapa versi AWS ParallelCluster menggunakan lingkungan virtual.

```
# AWS ParallelCluster v2
$ pcluster list -r REGION

# AWS ParallelCluster v3
$ pcluster list-clusters --region REGION
```

Memulai dan Menghentikan klaster

Perintah `pcluster start` dan `pcluster stop` AWS ParallelCluster 2.x harus diganti dengan [pcluster update-compute-fleet](#) perintah.

Memulai armada komputasi:

```
# AWS ParallelCluster v2
$ pcluster start \
  -r REGION \
  CLUSTER_NAME

# AWS ParallelCluster v3 - Slurm fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status START_REQUESTED

# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status ENABLED
```

Menghentikan armada komputasi:

```
# AWS ParallelCluster v2
$ pcluster stop \
  -r REGION \
  CLUSTER_NAME
```

```
# AWS ParallelCluster v3 - Slurm fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status STOP_REQUESTED

# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status DISABLED
```

Menghubungkan ke klaster

Perintah `pcluster ssh` AWS ParallelCluster 2.x memiliki nama parameter yang berbeda di AWS ParallelCluster 3.x. Lihat [pcluster ssh](#).

Menghubungkan ke cluster:

```
# AWS ParallelCluster v2
$ pcluster ssh \
  -r REGION \
  CLUSTER_NAME \
  -i ~/.ssh/id_rsa

# AWS ParallelCluster v3
$ pcluster ssh \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  -i ~/.ssh/id_rsa
```

Pembaruan konfigurasi IMDS

Dimulai dengan versi 3.0.0, AWS ParallelCluster memperkenalkan dukungan untuk membatasi akses ke IMDS node kepala (dan kredensi profil instance) ke subset superusers, secara default. Untuk informasi selengkapnya, lihat [Imdsproperti](#).

Wilayah yang Didukung untuk AWS ParallelCluster

AWS ParallelClusterversi 3 tersedia sebagai berikutWilayah AWS:

Nama Wilayah	Wilayah
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	as-barat-2
Afrika (Cape Town)	af-selatan-1
Asia Pasifik (Hong Kong)	ap-timur-1
Asia Pasifik (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-sentral-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-barat-2
Eropa (Milan)	eu-selatan-1
Eropa (Paris)	eu-west-3
Europe (Stockholm)	eu-utara-1

Nama Wilayah	Wilayah
Timur Tengah (Bahrain)	me-selatan-1
Amerika Selatan (São Paulo)	sa-east-1
AWS GovCloud (AS-Timur)	us-gov-east-1
AWS GovCloud (AS-Barat)	us-gov-west-1
Israel (Tel Aviv)	il-sentral-1

Menggunakan AWS ParallelCluster

Topik

- [AWS ParallelClusterUI](#)
- [AWS LambdaKonfigurasi VPC diAWS ParallelCluster](#)
- [AWS Identity and Access Management izin di AWS ParallelCluster](#)
- [Konfigurasi jaringan](#)
- [Node masuk](#)
- [Tindakan bootstrap kustom](#)
- [Bekerja dengan Amazon S3](#)
- [Mengerjakan Instans ITSpot](#)
- [Penjadwal didukung oleh AWS ParallelCluster](#)
- [Penyimpanan bersama](#)
- [AWS ParallelClustersumber daya dan penandaan](#)
- [Pemantauan AWS ParallelCluster dan log](#)
- [AWS CloudFormation sumber daya khusus](#)
- [Elastic Fabric Adapter](#)
- [Aktifkan Intel MPI](#)
- [API AWS ParallelCluster](#)
- [Connect ke node kepala melalui NICE DCV](#)
- [Menggunakan pcluster update-cluster](#)
- [AWS ParallelCluster Kustomisasi AMI](#)
- [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#)
- [Penambalan AMI dan penggantian instans EC2](#)
- [Sistem Operasi\)](#)

AWS ParallelClusterUI

AWS ParallelClusterUI adalah antarmuka pengguna berbasis web yang berfungsi sebagai dasbor untuk membuat, memantau, dan mengelola cluster. Anda menginstal dan mengaksesAWS

ParallelCluster UI di perangkat AndaAkun AWS. AWS ParallelClusterUI ditambahkan denganAWS ParallelCluster versi 3.5.0.

Untuk menginstalAWS ParallelCluster UI dan memulai, lihat[Menginstal AWS ParallelCluster UI](#) dan[Konfigurasi dan buat cluster dengan AWS ParallelCluster UI](#).

The screenshot displays the AWS ParallelCluster UI interface. At the top, the user is logged in as 'user@domain.com' in the 'eu-west-1' region. The main navigation menu includes 'Clusters', 'Images', and 'Users'. The 'Clusters' section is active, showing a list of clusters with columns for Name, Status, and Version. Two clusters are listed: 'hpc-cluster-1' with a status of 'CREATE COMPLETE' and version '3.5.0', and 'hpc-cluster-2' with a status of 'DELETE IN PROGRESS' and version '3.5.0'. Below the list, the details for 'hpc-cluster-1' are shown, including tabs for 'Details', 'Instances', 'Storage', 'Job scheduling', and 'Stack events'. The 'Details' tab is selected, showing properties such as 'Cluster configuration', 'SSH command', 'EC2 Instance Connect', 'Cluster status', 'Compute fleet status', 'Version', 'Region', 'Created time', and 'Latest update time'.

Name	Status	Version
hpc-cluster-1	CREATE COMPLETE	3.5.0
hpc-cluster-2	DELETE IN PROGRESS	3.5.0

Cluster: hpc-cluster-1

Properties

Cluster configuration VIEW	Cluster status CREATE COMPLETE	Region eu-west-1
SSH command Info ssh ec2-user@54.78.245.22	Compute fleet status RUNNING	Created time March 10, 2023 at 09:39 (UTC+1:00)
EC2 Instance Connect Info mssh -r eu-west-1 ec2-user@i-0b14dc1a2f5dc048e	Version 3.5.0	Latest update time March 10, 2023 at 09:39 (UTC+1:00)

AWS ParallelClusterUI yang mendukung fitur-fitur berikut:

- Menampilkan berikut ini:
 - Daftar klaster yang telah Anda buatAkun AWS bersama AndaAWS ParallelCluster.
 - Status dan detail yang tersedia untuk klaster yang Anda cantumkan.
 - CloudFormation tumpukan acara danAWS ParallelCluster log yang dapat Anda gunakan untuk pemantauan.
 - Status pekerjaan yang berjalan pada cluster Anda.

- Daftar gambar kustom yang dapat Anda gunakan untuk membangun kluster.
- Daftar gambar resmi yang digunakan UI untuk membuat cluster.
- Daftar pengguna yang memiliki akses keAWS ParallelCluster UI. Anda dapat menambahkan dan menghapus pengguna.
- Memberikan step-by-step panduan untuk membuat dan mengedit (memperbarui) kluster dan memilih fitur kluster yang didukung untuk ditambahkan, diedit, atau dihapus. Bidang input yang tidak dapat diakses tidak dapat diubah untuk konfigurasi kluster yang sedang diedit. Anda memiliki opsi untuk melakukan validasi dry run pada konfigurasi kluster Anda sebelum penerapan kluster.
- Fitur link shell langsung untuk mengakses node kepala dalam tampilan Cluster. Pilih Tambahkan sesi SSM selama step-by-step panduan untuk menambahkan akses shell langsung, dan kebijakan Inti Instans Terkelola SSM di node kepala.

Pertimbangkan hal berikut saat menggunakanAWS ParallelCluster UI:

- Anda hanya dapat membuat dan mengedit kluster atau membuat gambar denganAWS ParallelCluster versi yang sama yang digunakan untuk membuatAWS ParallelCluster UI. Versi sebelumnya cluster atau gambar hanya dapat dilihat. Jika Anda mengelola beberapa versi kluster dan gambar, sebaiknya buat instanceAWS ParallelCluster UI untuk mendukung setiap versi.
- AWS ParallelClusterUI dirancang untuk mencerminkan fungsionalitas`pccluster` CLI. Ada beberapa perbedaan. Jika Anda menyelaraskan dengan step-by-step panduan, maka Anda menggunakan semua fitur yang didukung. Sebelum penyebaran, Anda memiliki opsi untuk mengedit cluster atau konfigurasi gambar secara manual. Jika Anda melakukan ini, kami sarankan Anda memvalidasi konfigurasi dengan memilih Dry run untuk memverifikasi bahwa pengeditan Anda sepenuhnya didukung.

Note

AWS ParallelClusterUI tidak mendukungAWS Batch.

AWS LambdaKonfigurasi VPC diAWS ParallelCluster

AWS ParallelClusterdigunakanAWS Lambda untuk melakukan operasi selama siklus hidup cluster. Sebuah [AWS Lambdafungsi selalu berjalan di VPC](#) yang dimiliki oleh layanan Lambda. Fungsi

Lambda ini juga dapat terhubung ke subnet pribadi di virtual private cloud (VPC) untuk mengakses sumber daya pribadi.

Note

Fungsi Lambda tidak dapat terhubung langsung ke VPC dengan tenancy instans khusus. Untuk terhubung ke sumber daya dalam VPC khusus, sambungkan VPC kedua dengan penyewaan default yang dapat terhubung ke VPC khusus. Untuk informasi selengkapnya, lihat [Instans Khusus](#) di Panduan Pengguna EC2 untuk Instans Linux dan [Bagaimana cara menghubungkan fungsi Lambda ke VPC khusus?](#) dari PusatAWS Pengetahuan.

Fungsi Lambda yang dibuat olehAWS ParallelCluster dapat dihubungkan ke VPC pribadi. Fungsi Lambda ini perlu diaksesLayanan AWS. Anda dapat menyediakan akses melalui internet atau endpoint VPC dengan menggunakan metode berikut.

- Akses internet

Untuk mengakses internet danLayanan AWS, fungsi Lambda memerlukan penerjemahan alamat jaringan (NAT). Rutekan lalu lintas keluar dari subnet pribadi Anda ke [gateway NAT](#) di subnet publik.

- Titik akhir VPC

Beberapa layanan AWS menawarkan [VPC endpoint](#). Anda dapat menggunakan VPC endpoint untuk terhubungLayanan AWS ke VPC yang tidak memiliki akses internet. Untuk melihat daftar endpointAWS ParallelCluster VPC, lihat [Networking](#).

Note

Setiap kombinasi subnet dan grup keamanan harus menyediakan akses untukLayanan AWS menggunakan salah satu metode ini. Subnet dan grup keamanan harus berada di VPC yang sama.

Untuk informasi selengkapnya, lihat [titik akhir VPC](#) di Panduan Pengguna Amazon Virtual Private Cloud dan [akses Internet dan layanan untuk fungsi yang terhubung dengan VPC](#) di PanduanAWS Lambda Pengembang.

Untuk mengonfigurasi penggunaan fungsi Lambda dan VPC, lihat [DeploymentSettings/LambdaFunctionsVpcConfig](#) untuk kluster atau [DeploymentSettings/LambdaFunctionsVpcConfig](#) untuk gambar.

AWS Identity and Access Management izin di AWS ParallelCluster

AWS ParallelCluster menggunakan izin IAM untuk mengontrol akses ke sumber daya saat membuat dan mengelola cluster.

Untuk membuat dan mengelola cluster di AWS akun, AWS ParallelCluster memerlukan izin di dua tingkat:

- Izin yang diperlukan `pcluster` pengguna untuk menjalankan perintah `pcluster` CLI untuk membuat dan mengelola cluster.
- Izin yang diperlukan sumber daya cluster untuk melakukan tindakan kluster.

AWS ParallelCluster menggunakan [profil dan peran instans EC2](#) untuk memberikan izin sumber daya cluster. Untuk mengelola izin sumber daya cluster, AWS ParallelCluster juga memerlukan izin ke sumber daya IAM. Untuk informasi selengkapnya, lihat [AWS ParallelCluster kebijakan contoh pengguna untuk mengelola sumber daya IAM](#).

`pcluster` pengguna memerlukan izin IAM untuk menggunakan `pcluster` CLI untuk membuat dan mengelola cluster dan sumber dayanya. Izin ini disertakan dalam kebijakan IAM yang dapat ditambahkan ke pengguna atau peran. Untuk informasi selengkapnya tentang peran IAM, lihat [Membuat peran pengguna](#) di Panduan AWS Identity and Access Management Pengguna.

Anda juga bisa menggunakan [AWS ParallelCluster parameter konfigurasi untuk mengelola izin IAM](#).

Bagian berikut berisi izin yang diperlukan dengan contoh.

Untuk menggunakan contoh kebijakan, ganti `<REGION><AWS ACCOUNT ID>`, dan string serupa dengan nilai yang sesuai.

Contoh kebijakan berikut mencakup Amazon Resource Names (ARN) untuk sumber daya. Jika Anda bekerja di partisi AWS GovCloud (US) atau AWS China, ARN harus diubah. Secara khusus, mereka harus diubah dari “arn:aws” menjadi “arn:aws-us-gov” untuk AWS GovCloud (US) partisi atau “arn:aws-cn” untuk partisi China. Untuk informasi selengkapnya, lihat [Nama Sumber Daya Amazon \(ARN\) di AWS GovCloud \(US\) Wilayah](#) di Panduan AWS GovCloud (US) Pengguna dan [ARN untuk AWS layanan di Tiongkok dalam Memulai AWS layanan di Tiongkok](#).

Anda dapat melacak perubahan pada contoh kebijakan dalam [AWS ParallelCluster dokumentasi pada GitHub](#).

Topik

- [AWS ParallelCluster Peran instans EC2](#)
- [AWS ParallelCluster contoh kebijakan pcluster pengguna](#)
- [AWS ParallelCluster kebijakan contoh pengguna untuk mengelola sumber daya IAM](#)
- [AWS ParallelCluster parameter konfigurasi untuk mengelola izin IAM](#)

AWS ParallelCluster Peran instans EC2

Saat Anda membuat klaster dengan pengaturan konfigurasi default, AWS ParallelCluster gunakan [profil instans](#) EC2 untuk secara otomatis membuat [peran instans](#) EC2 cluster default yang menyediakan izin yang diperlukan untuk membuat dan mengelola cluster dan sumber dayanya.

Alternatif untuk menggunakan peran AWS ParallelCluster instance default

Sebagai pengganti peran AWS ParallelCluster instans default, Anda dapat menggunakan pengaturan konfigurasi `InstanceRole` cluster untuk menentukan peran IAM Anda sendiri yang ada untuk EC2. Untuk informasi selengkapnya, lihat [AWS ParallelCluster parameter konfigurasi untuk mengelola izin IAM](#). Biasanya, Anda menentukan peran IAM yang ada untuk sepenuhnya mengontrol izin yang diberikan kepada EC2.

Jika maksud Anda adalah menambahkan kebijakan tambahan ke peran instans default, sebaiknya Anda meneruskan kebijakan IAM tambahan dengan menggunakan setelan [AdditionalIamPolicies](#) konfigurasi, bukan setelan [InstanceProfile](#) atau [InstanceRole](#) setelan. Anda dapat memperbarui `AdditionalIamPolicies` saat memperbarui klaster, namun, Anda tidak dapat memperbarui `InstanceRole` saat memperbarui klaster.

AWS ParallelCluster contoh kebijakan **pcluster** pengguna

Contoh berikut menunjukkan kebijakan pengguna yang diperlukan untuk membuat dan mengelola AWS ParallelCluster dan sumber dayanya dengan menggunakan `pcluster` CLI. Anda dapat melampirkan kebijakan ke pengguna atau peran.

Topik

- [Kebijakan AWS ParallelCluster pcluster pengguna dasar](#)

- [Kebijakan AWS ParallelCluster pcluster pengguna tambahan saat menggunakan AWS Batch scheduler](#)
- [Kebijakan AWS ParallelCluster pcluster pengguna tambahan saat menggunakan Amazon FSx for Lustre](#)
- [AWS ParallelCluster kebijakan pcluster pengguna build image](#)

Kebijakan AWS ParallelCluster **pcluster** pengguna dasar

Kebijakan berikut menunjukkan izin yang diperlukan untuk menjalankan AWS ParallelCluster `pcluster` perintah.

Tindakan terakhir yang tercantum dalam kebijakan disertakan untuk memberikan validasi rahasia apa pun yang ditentukan dalam konfigurasi kluster. Misalnya, AWS Secrets Manager rahasia digunakan untuk mengkonfigurasi [DirectoryService](#) integrasi. Dalam hal ini, sebuah cluster dibuat hanya jika ada rahasia yang valid di [PasswordSecretArn](#). Jika tindakan ini dihilangkan, validasi rahasia dilewati. Untuk meningkatkan postur keamanan Anda, sebaiknya Anda mencatat pernyataan kebijakan ini dengan hanya menambahkan rahasia yang ditentukan dalam konfigurasi kluster Anda.

Note

Jika sistem file Amazon EFS yang ada adalah satu-satunya sistem file yang digunakan di kluster Anda, Anda dapat mencatat contoh pernyataan kebijakan Amazon EFS ke sistem file tertentu yang direferensikan [Bagian SharedStorage](#) dalam file konfigurasi cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Read"
    },
    {
      "Action": [
        "ec2:AllocateAddress",
```

```

        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateFleet",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateNetworkInterface",
        "ec2:CreatePlacementGroup",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2>DeleteLaunchTemplate",
        "ec2>DeleteNetworkInterface",
        "ec2>DeletePlacementGroup",
        "ec2>DeleteSecurityGroup",
        "ec2>DeleteVolume",
        "ec2:DisassociateAddress",
        "ec2:ModifyLaunchTemplate",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifyVolume",
        "ec2:ModifyVolumeAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RunInstances",
        "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Write"
},
{
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTagsOfResource",
        "dynamodb:CreateTable",
        "dynamodb>DeleteTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:Query",
        "dynamodb:TagResource"
    ]
}

```

```
    ],
    "Resource": "arn:aws:dynamodb:*:<AWS ACCOUNT ID>:table/parallelcluster-*",
    "Effect": "Allow",
    "Sid": "DynamoDB"
  },
  {
    "Action": [
      "route53:ChangeResourceRecordSets",
      "route53:ChangeTagsForResource",
      "route53:CreateHostedZone",
      "route53>DeleteHostedZone",
      "route53:GetChange",
      "route53:GetHostedZone",
      "route53:ListResourceRecordSets",
      "route53:ListQueryLoggingConfigs"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
  },
  {
    "Action": [
      "cloudformation:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudFormation"
  },
  {
    "Action": [
      "cloudwatch:PutDashboard",
      "cloudwatch:ListDashboards",
      "cloudwatch>DeleteDashboards",
      "cloudwatch:GetDashboard",
      "cloudwatch:PutMetricAlarm",
      "cloudwatch>DeleteAlarms",
      "cloudwatch:DescribeAlarms",
      "cloudwatch:PutCompositeAlarm"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatch"
  },
  {
```



```

    "Action": [
      "iam:GetRole",
      "iam:GetRolePolicy",
      "iam:GetPolicy",
      "iam:SimulatePrincipalPolicy",
      "iam:GetInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:policy/*",
      "arn:aws:iam::aws:policy/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRead"
  },
  {
    "Action": [
      "iam:CreateInstanceProfile",
      "iam>DeleteInstanceProfile",
      "iam:AddRoleToInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamInstanceProfile"
  },
  {
    "Condition": {
      "StringEqualsIfExists": {
        "iam:PassedToService": [
          "lambda.amazonaws.com",
          "ec2.amazonaws.com",
          "spotfleet.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ]
  }
}

```

```

    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda:DeleteFunction",
      "lambda:GetFunctionConfiguration",
      "lambda:GetFunction",
      "lambda:InvokeFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
      "lambda:UpdateFunctionConfiguration",
      "lambda:TagResource",
      "lambda:ListTags",
      "lambda:UntagResource"
    ],
    "Resource": [
      "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:parallelcluster-*",
      "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
  },
  {
    "Action": [
      "s3:*"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*",
      "arn:aws:s3:::aws-parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
  },
  {
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": "arn:aws:s3:::*-aws-parallelcluster*",
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
  }

```

```
    },
    {
      "Action": [
        "elasticfilesystem:*"
      ],
      "Resource": [
        "arn:aws:elasticfilesystem:*:<AWS ACCOUNT ID>:*"
      ],
      "Effect": "Allow",
      "Sid": "EFS"
    },
    {
      "Action": [
        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy",
        "logs:DescribeLogGroups",
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:CreateExportTask",
        "logs:DescribeLogStreams",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:PutMetricFilter",
        "logs>DeleteMetricFilter"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "CloudWatchLogs"
    },
    {
      "Action": [
        "resource-groups:ListGroupResources"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "ResourceGroupRead"
    },
    {
      "Sid": "AllowDescribingFileCache",
      "Effect": "Allow",
      "Action": [
```

```

        "fsx:DescribeFileCaches"
    ],
    "Resource": "*"
  },
  {
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET
NAME>",
    "Effect": "Allow"
  }
]
}

```

Kebijakan AWS ParallelCluster **pcluster** pengguna tambahan saat menggunakan AWS Batch scheduler

Jika Anda perlu membuat dan mengelola kluster dengan AWS Batch penjadwal, kebijakan tambahan berikut diperlukan.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEqualsIfExists": {
          "iam:PassedToService": [
            "ecs-tasks.amazonaws.com",
            "batch.amazonaws.com",
            "codebuild.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamPassRole"
    },
    {
      "Condition": {

```

```

        "StringEquals": {
            "iam:AWSServiceName": [
                "batch.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/
batch.amazonaws.com/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "codebuild:*"
    ],
    "Resource": "arn:aws:codebuild:*:<AWS ACCOUNT ID>:project/pcluster-*",
    "Effect": "Allow"
},
{
    "Action": [
        "ecr:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ECR"
},
{
    "Action": [
        "batch:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Batch"
},
{
    "Action": [
        "events:*"
    ],
    "Resource": "*",

```

```

    "Effect": "Allow",
    "Sid": "AmazonCloudWatchEvents"
  },
  {
    "Action": [
      "ecs:DescribeContainerInstances",
      "ecs:ListContainerInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ECS"
  }
]
}

```

Kebijakan AWS ParallelCluster **pcluster** pengguna tambahan saat menggunakan Amazon FSx for Lustre

Jika Anda perlu membuat dan mengelola kluster dengan Amazon FSx for Lustre, kebijakan tambahan berikut diperlukan.

Note

Jika sistem file Amazon FSx yang ada adalah satu-satunya sistem file yang digunakan dalam kluster Anda, Anda dapat mencatat contoh pernyataan kebijakan Amazon FSx ke sistem file tertentu yang direferensikan dalam file konfigurasi cluster. [Bagian SharedStorage](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "fsx.amazonaws.com",
            "s3.data-source.lustre.fsx.amazonaws.com"
          ]
        }
      },
      "Action": [

```

```

        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "fsx:*"
    ],
    "Resource": [
        "arn:aws:fsx:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "FSx"
},
{
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/*",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:Get*",
        "s3:List*",
        "s3:PutObject"
    ],
    "Resource": "arn:aws:s3:::<S3 NAME>",
    "Effect": "Allow"
}
]
}

```

AWS ParallelCluster kebijakan **pcluster** pengguna build image

Pengguna yang ingin membuat gambar EC2 kustom dengan AWS ParallelCluster harus memiliki set izin berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeImages",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeInstanceTypes",
        "ec2:DeregisterImage",
        "ec2>DeleteSnapshot"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2"
    },
    {
      "Action": [
        "iam:CreateInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/ParallelClusterImage*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IAM"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "lambda.amazonaws.com",
            "ec2.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ]
    }
  ]
}
```



```

    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IAMPassRole"
},
{
    "Action": [
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs>DeleteLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/imagebuilder/
ParallelClusterImage-*",
        "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/lambda/
ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "CloudWatch"
},
{
    "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation>CreateStack",
        "cloudformation>DeleteStack"
    ],
    "Resource": [
        "arn:aws:cloudformation:*:<AWS ACCOUNT ID>:stack/*"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormation"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda:GetFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda>DeleteFunction",
        "lambda:TagResource",

```

```

        "lambda:ListTags",
        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "imagebuilder:Get*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ImageBuilderGet"
},
{
    "Action": [
        "imagebuilder:CreateImage",
        "imagebuilder:TagResource",
        "imagebuilder:CreateImageRecipe",
        "imagebuilder:CreateComponent",
        "imagebuilder:CreateDistributionConfiguration",
        "imagebuilder:CreateInfrastructureConfiguration",
        "imagebuilder>DeleteImage",
        "imagebuilder>DeleteComponent",
        "imagebuilder>DeleteImageRecipe",
        "imagebuilder>DeleteInfrastructureConfiguration",
        "imagebuilder>DeleteDistributionConfiguration"
    ],
    "Resource": [
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image/parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:component/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:distribution-configuration/
parallelclusterimage-*",
        "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:infrastructure-configuration/
parallelclusterimage-*"
    ],
    "Effect": "Allow",
    "Sid": "ImageBuilder"
}

```

```
    },
    {
      "Action": [
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:ListBucketVersions"
      ],
      "Resource": [
        "arn:aws:s3:::parallelcluster-*"
      ],
      "Effect": "Allow",
      "Sid": "S3Bucket"
    },
    {
      "Action": [
        "sns:GetTopicAttributes",
        "sns:TagResource",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Publish",
        "SNS:DeleteTopic",
        "SNS:Unsubscribe"
      ],
      "Resource": [
        "arn:aws:sns:*:<AWS ACCOUNT ID>:ParallelClusterImage-*"
      ],
      "Effect": "Allow",
      "Sid": "SNS"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::parallelcluster-*/*"
      ],
      "Effect": "Allow",
      "Sid": "S3Objects"
    },
  ],
  {
```

```
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "imagebuilder.amazonaws.com"
      }
    }
  ]
}
```

AWS ParallelCluster kebijakan contoh pengguna untuk mengelola sumber daya IAM

Saat menggunakan AWS ParallelCluster untuk membuat kluster atau AMI kustom, kebijakan IAM harus disediakan yang berisi izin untuk memberikan kumpulan izin yang diperlukan ke komponen. AWS ParallelCluster Sumber daya IAM ini dapat dibuat secara otomatis oleh AWS ParallelCluster atau disediakan sebagai input saat membuat cluster atau gambar khusus.

Anda dapat menggunakan mode berikut untuk memberi AWS ParallelCluster pengguna izin yang diperlukan untuk mengakses sumber daya IAM dengan menggunakan kebijakan IAM tambahan dalam konfigurasi.

Topik

- [Mode akses IAM istimewa](#)
- [Mode akses IAM terbatas](#)
- [PermissionsBoundarymodus](#)

Mode akses IAM istimewa

Dengan mode ini, AWS ParallelCluster secara otomatis membuat semua sumber daya IAM yang diperlukan. Kebijakan IAM ini dicakup untuk mengaktifkan akses ke sumber daya cluster saja.

Untuk mengaktifkan mode akses IAM Privileged, tambahkan kebijakan berikut ke peran pengguna.

Note

Jika Anda mengonfigurasi [AdditionalPolicies](#) parameter [HeadNodeIam/SchedulingSlurmQueues/AdditionalPolicies](#) atau [Iam///](#), Anda harus memberi AWS ParallelCluster pengguna izin untuk melampirkan dan melepaskan kebijakan peran untuk setiap kebijakan tambahan seperti yang ditunjukkan dalam kebijakan berikut. Tambahkan ARN kebijakan tambahan ke kondisi untuk melampirkan dan melepaskan kebijakan peran.

Warning

Mode ini memungkinkan pengguna untuk memiliki hak Administrator IAM di Akun AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:TagRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRole"
    },
    {
      "Action": [
        "iam>CreateRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamCreateRole"
    }
  ]
}
```

```

    "Action": [
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
  },
  {
    "Condition": {
      "ArnLike": {
        "iam:PolicyARN": [
          "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
          "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
          "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
          "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
          "arn:aws:iam::aws:policy/AWSBatchFullAccess",
          "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
          "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",
          "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
          "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
          "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
          "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
          "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
        ]
      }
    },
    "Action": [
      "iam:AttachRolePolicy",
      "iam:DetachRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamPolicy"
  }
]
}

```

Mode akses IAM terbatas

Jika tidak ada kebijakan IAM tambahan yang diberikan kepada pengguna, peran IAM yang diperlukan oleh cluster atau pembuatan gambar kustom harus dibuat secara manual oleh administrator dan diteruskan sebagai bagian dari konfigurasi cluster.

Saat membuat cluster, parameter berikut diperlukan:

- [Iam / Roles / LambdaFunctionsRole](#)
- [HeadNode / Iam / InstanceRole](#) | [InstanceProfile](#)
- [Scheduling / SlurmQueues / Iam / InstanceRole](#) | [InstanceProfile](#)

Saat membuat gambar khusus, parameter berikut diperlukan:

- [Build / Iam / InstanceRole](#) | [InstanceProfile](#)
- [Build / Iam / CleanupLambdaRole](#)

Peran IAM yang diteruskan sebagai bagian dari parameter yang tercantum di atas harus dibuat pada awalan `/parallelcluster/` jalur. Jika ini tidak memungkinkan, kebijakan pengguna perlu diperbarui untuk memberikan `iam:PassRole` izin pada peran kustom tertentu, seperti pada contoh berikut.

```
{
  "Condition": {
    "StringEqualsIfExists": {
      "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "lambda.amazonaws.com",
        "ec2.amazonaws.com",
        "spotfleet.amazonaws.com",
        "batch.amazonaws.com",
        "codebuild.amazonaws.com"
      ]
    }
  },
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    <list all custom IAM roles>
  ]
}
```

```
],  
  "Effect": "Allow",  
  "Sid": "IamPassRole"  
}
```

Warning

Saat ini mode ini tidak memungkinkan pengelolaan AWS Batch cluster karena tidak semua peran IAM dapat diteruskan dalam konfigurasi cluster.

PermissionsBoundarymodus

Mode ini mendelegasikan AWS ParallelCluster pembuatan peran IAM yang terikat pada batas izin IAM yang dikonfigurasi. Untuk informasi selengkapnya tentang batas izin IAM, lihat [Batas izin untuk entitas IAM](#) di Panduan Pengguna IAM.

Kebijakan berikut perlu ditambahkan ke peran pengguna.

Dalam kebijakan tersebut, ganti `< permissions-boundary-arn >` dengan ARN kebijakan IAM yang akan diberlakukan sebagai batas izin.

Warning

Jika Anda mengonfigurasi [AdditionalPolicies](#) parameter [HeadNodeIam//AdditionalPolicies](#) atau [SchedulingSlurmQueues/Iam//](#), Anda harus memberikan izin kepada pengguna untuk melampirkan dan melepaskan kebijakan peran untuk setiap kebijakan tambahan seperti yang ditunjukkan dalam kebijakan berikut. Tambahkan ARN kebijakan tambahan ke kondisi untuk melampirkan dan melepaskan kebijakan peran.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": [  
        "iam:CreateServiceLinkedRole",  
        "iam>DeleteRole",  
        "iam:TagRole"  
      ]  
    }  
  ]  
}
```



```
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          <permissions-boundary-arn>
        ]
      }
    },
    "Action": [
      "iam:CreateRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamCreateRole"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
          <permissions-boundary-arn>
        ]
      }
    },
    "Action": [
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": [
```

```

        <permissions-boundary-arn>
    ]
},
"ArnLike": {
    "iam:PolicyARN": [
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
        "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
        "arn:aws:iam::aws:policy/AWSBatchFullAccess",
        "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
        "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
        "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
        "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
    ]
}
},
"Action": [
    "iam:AttachRolePolicy",
    "iam:DetachRolePolicy"
],
"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
"Effect": "Allow",
"Sid": "IamPolicy"
}
]
}

```

Saat mode ini diaktifkan, Anda harus menentukan batas izin ARN di [Iam](#) parameter konfigurasi [PermissionsBoundary](#)/saat membuat atau memperbarui cluster dan parameter [Iam/PermissionBoundary](#)/saat membuat gambar khusus. [Build](#)

AWS ParallelCluster parameter konfigurasi untuk mengelola izin IAM

AWS ParallelCluster mengekspos serangkaian opsi konfigurasi untuk menyesuaikan dan mengelola izin dan peran IAM yang digunakan dalam kluster atau selama proses pembuatan AMI kustom.

Topik

- [Konfigurasi cluster](#)
- [Konfigurasi Gambar Kustom](#)

Konfigurasi cluster

Topik

- [Peran IAM simpul kepala](#)
- [Akses Amazon S3](#)
- [Kebijakan IAM tambahan](#)
- [AWS Lambda peran fungsi](#)
- [Hitung peran IAM node](#)
- [Batas izin](#)

Peran IAM simpul kepala

[HeadNode](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Dengan opsi ini, Anda mengganti peran IAM default yang ditetapkan ke node kepala kluster. Untuk detail tambahan, silakan merujuk ke [InstanceProfile](#) referensi.

Berikut adalah kumpulan kebijakan minimal yang akan digunakan sebagai bagian dari peran ini ketika penjadwal adalah Slurm:

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [Membuat peran IAM dan pengguna untuk digunakan dengan CloudWatch agen](#) di Panduan CloudWatch Pengguna Amazon.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola AWS Systems Manager](#) di Panduan AWS Systems Manager Pengguna.
- Kebijakan IAM tambahan:

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

```

        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.<REGION>/*",
        "arn:aws:s3:::parallelcluster-* -v1-do-not-delete/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:BatchGetItem"
    ],
    "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/
parallelcluster-*",
    "Effect": "Allow"
},
{
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/parallelcluster:node-type": "Compute"
        }
    },
    "Action": "ec2:TerminateInstances",
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:RunInstances",
        "ec2:CreateFleet"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [

```

```
        "ec2.amazonaws.com"
    ]
  },
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeInstanceStatus",
    "ec2:DescribeVolumes",
    "ec2:DescribeInstanceAttribute",
    "ec2:DescribeCapacityReservations"
  ],
  "Resource": "*",
  "Effect": "Allow"
},
{
  "Action": [
    "ec2:CreateTags",
    "ec2:AttachVolume"
  ],
  "Resource": [
    "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
    "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "cloudformation:DescribeStacks",
    "cloudformation:DescribeStackResource",
    "cloudformation:SignalResource"
  ],
  "Resource": "*",
  "Effect": "Allow"
},
}
```

```

    {
      "Action": [
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT
ID>:secret:<SECRET_ID>",
      "Effect": "Allow"
    }
  ]
}

```

Perhatikan bahwa jika [Scheduling//SlurmQueues/Iam/InstanceRole](#) digunakan untuk mengganti peran IAM komputasi, kebijakan simpul kepala yang dilaporkan di atas perlu menyertakan peran tersebut di Resource bagian izin. `iam:PassRole`

Berikut adalah kumpulan kebijakan minimal yang akan digunakan sebagai bagian dari peran ini saat penjadwal adalah AWS Batch:

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [Membuat peran IAM dan pengguna untuk digunakan dengan CloudWatch agen](#) di Panduan CloudWatch Pengguna Amazon.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola AWS Systems Manager](#) di Panduan AWS Systems Manager Pengguna.
- Kebijakan IAM tambahan:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [

```

```

        "arn:aws:s3:::parallelcluster-* -v1-do-not-delete/*"
    ],
    "Effect": "Allow"
},
{
    "Action": "s3:GetObject",
    "Resource": [
        "arn:aws:s3:::dcv-license.<REGION>/*",
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
    ],
    "Effect": "Allow"
},
{
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "batch.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "batch:DescribeJobQueues",
        "batch:DescribeJobs",
        "batch:ListJobs",
        "batch:DescribeComputeEnvironments"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "batch:SubmitJob",
        "batch:TerminateJob",
        "logs:GetLogEvents",

```

```

        "ecs:ListContainerInstances",
        "ecs:DescribeContainerInstances",
    ],
    "Resource": [
        "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-
stream:PclusterJobDefinition*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/AWSBatch-
PclusterComputeEnviron*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/AWSBatch-Pcluster*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/
PclusterJobQueue*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-definition/
PclusterJobDefinition*:*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:CreateTags",
        "ec2:AttachVolume"
    ],
    "Resource": [
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStacks",
        "cloudformation:SignalResource"
    ],

```



```
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Action": "secretsmanager:GetSecretValue",
        "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT
ID>:secret:<SECRET_ID>",
        "Effect": "Allow"
    }
]
}
```

Akses Amazon S3

[HeadNode/Iam/S3Access](#) atau [Scheduling/SlurmQueues/S3Access](#)

Di bagian konfigurasi ini, Anda dapat menyesuaikan akses Amazon S3 dengan memberikan kebijakan Amazon S3 tambahan ke peran IAM yang terkait dengan node kepala atau node komputasi kluster saat peran tersebut dibuat oleh AWS ParallelCluster. Untuk informasi selengkapnya, lihat dokumentasi referensi untuk setiap parameter konfigurasi.

Parameter ini hanya dapat digunakan ketika pengguna dikonfigurasi dengan [Mode akses IAM istimewa](#) atau [PermissionsBoundary](#) modus.

Kebijakan IAM tambahan

[HeadNode/Iam/AdditionalIamPolicies](#) atau [SlurmQueues/Iam/AdditionalIamPolicies](#)

Gunakan opsi ini untuk melampirkan kebijakan IAM terkelola tambahan ke peran IAM yang terkait dengan node kepala atau node komputasi kluster saat peran tersebut dibuat oleh AWS ParallelCluster.

Warning

Untuk menggunakan opsi ini, pastikan [AWS ParallelCluster pengguna](#) diberikan `iam:AttachRolePolicy` dan `iam:DetachRolePolicy` izin untuk kebijakan IAM yang perlu dilampirkan.

AWS Lambda peran fungsi

[Iam](#) / [Roles](#) / [LambdaFunctionsRole](#)

Opsi ini mengesampingkan peran yang dilampirkan ke semua AWS Lambda fungsi yang digunakan selama proses pembuatan cluster. AWS Lambda perlu dikonfigurasi sebagai kepala sekolah diizinkan untuk mengambil peran.

Note

Jika [DeploymentSettings/LambdaFunctionsVpcConfig](#) disetel, [LambdaFunctionsRole](#) harus menyertakan [izin AWS Lambda peran](#) untuk mengatur konfigurasi VPC.

Berikut adalah seperangkat kebijakan minimal yang akan digunakan sebagai bagian dari peran ini:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "arn:aws:route53::hostedzone/*",
      "Effect": "Allow"
    },
    {
      "Action": ["logs:CreateLogStream", "logs:PutLogEvents"],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/lambda/
pcluster-*"
    },
    {
      "Action": "ec2:DescribeInstances",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "ec2:TerminateInstances",
      "Condition": {
```

```

    "StringEquals": {
      "ec2:ResourceTag/parallelcluster:node-type": "Compute"
    }
  },
  "Effect": "Allow",
  "Resource": "*"
},
{
  "Action": [
    "s3:DeleteObject",
    "s3:DeleteObjectVersion",
    "s3:ListBucket",
    "s3:ListBucketVersions"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:s3:::parallelcluster-* -v1-do-not-delete",
    "arn:aws:s3:::parallelcluster-* -v1-do-not-delete/*"
  ]
}
]
}

```

Hitung peran IAM node

[Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Opsi ini memungkinkan untuk mengganti peran IAM yang ditetapkan ke node komputasi cluster. Untuk informasi selengkapnya, lihat [InstanceProfile](#).

Berikut adalah seperangkat kebijakan minimal yang akan digunakan sebagai bagian dari peran ini:

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [Membuat peran IAM dan pengguna untuk digunakan dengan CloudWatch agen](#) di Panduan CloudWatch Pengguna Amazon.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola AWS Systems Manager](#) di Panduan AWS Systems Manager Pengguna.
- Kebijakan IAM tambahan:

```

{
  "Version": "2012-10-17",

```

```

    "Statement": [
      {
        "Action": [
          "dynamodb:Query",
          "dynamodb:UpdateItem",
          "dynamodb:PutItem",
          "dynamodb:GetItem"
        ],
        "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/
parallelcluster-*",
        "Effect": "Allow"
      },
      {
        "Action": "s3:GetObject",
        "Resource": [
          "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
        ],
        "Effect": "Allow"
      },
      {
        "Action": "ec2:DescribeInstanceAttribute",
        "Resource": "*",
        "Effect": "Allow"
      },
      {
        "Action": "cloudformation:DescribeStackResource",
        "Resource": [
          "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*" ],
        "Effect": "Allow"
      }
    ]
  }
}

```

Batas izin

[Iam / PermissionsBoundary](#)

Parameter ini memaksa AWS ParallelCluster untuk melampirkan kebijakan IAM yang diberikan sebagai a PermissionsBoundary ke semua peran IAM yang dibuat sebagai bagian dari penerapan cluster.

Lihat [PermissionsBoundarymodus](#) daftar kebijakan yang diperlukan oleh pengguna saat setelah ini ditentukan.

Konfigurasi Gambar Kustom

Topik

- [Peran instans untuk EC2 Image Builder](#)
- [AWS Lambda peran pembersihan](#)
- [Kebijakan IAM tambahan](#)
- [Batas izin](#)

Peran instans untuk EC2 Image Builder

[Build](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Dengan opsi ini, Anda mengganti peran IAM yang ditetapkan ke instans EC2 yang diluncurkan oleh EC2 Image Builder untuk membuat AMI kustom.

Berikut adalah seperangkat kebijakan minimal yang akan digunakan sebagai bagian dari peran ini:

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola AWS Systems Manager](#) di Panduan AWS Systems Manager Pengguna.
- `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder`kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [EC2InstanceProfileForImageBuilderkebijakan](#) di Panduan Pengguna Image Builder.
- Kebijakan IAM tambahan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateTags",
        "ec2:ModifyImageAttribute"
      ],
      "Resource": "arn:aws:ec2:<REGION>::image/*",
      "Effect": "Allow"
    }
  ]
}
```

```
]
}
```

AWS Lambda peran pembersihan

[Build](#) / [Iam](#) / [CleanupLambdaRole](#)

Opsi ini mengesampingkan peran yang dilampirkan ke semua AWS Lambda fungsi yang digunakan selama proses pembuatan gambar kustom. AWS Lambda perlu dikonfigurasi sebagai kepala sekolah diizinkan untuk mengambil peran.

Note

Jika [DeploymentSettings/LambdaFunctionsVpcConfig](#) disetel, `CleanupLambdaRole` harus menyertakan [izin AWS Lambda peran](#) untuk mengatur konfigurasi VPC.

Berikut adalah seperangkat kebijakan minimal yang akan digunakan sebagai bagian dari peran ini:

- `arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` kebijakan IAM terkelola. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola untuk fitur Lambda](#) di Panduan AWS Lambda Pengembang.
- Kebijakan IAM tambahan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:DetachRolePolicy",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy"
      ],
      "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam>DeleteInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
```

```

    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/
parallelcluster/*",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteInfrastructureConfiguration",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT
ID>:infrastructure-configuration/parallelclusterimage-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder:DeleteComponent"
    ],
    "Resource": [
      "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:component/
parallelclusterimage-*/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteImageRecipe",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteDistributionConfiguration",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:distribution-
configuration/parallelclusterimage-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder:DeleteImage",
      "imagebuilder:GetImage",
      "imagebuilder:CancelImageCreation"
    ],
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image/
parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "cloudformation:DeleteStack",

```

```

    "Resource": "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*",
    "Effect": "Allow"
  },
  {
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:<REGION>::image/*",
    "Effect": "Allow"
  },
  {
    "Action": "tag:TagResources",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:DeleteFunction",
      "lambda:RemovePermission"
    ],
    "Resource": "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:ParallelClusterImage-*",
    "Effect": "Allow"
  },
  {
    "Action": "logs:DeleteLogGroup",
    "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/lambda/ParallelClusterImage-*:*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "SNS:GetTopicAttributes",
      "SNS:DeleteTopic",
      "SNS:GetSubscriptionAttributes",
      "SNS:Unsubscribe"
    ],
    "Resource": "arn:aws:sns:<REGION>:<AWS ACCOUNT ID>:ParallelClusterImage-*",
    "Effect": "Allow"
  }
]
}

```


Kebijakan IAM tambahan

[Build](#) / [Iam](#) / [AdditionalIamPolicies](#)

Anda menggunakan opsi ini untuk melampirkan kebijakan IAM terkelola tambahan ke peran yang terkait dengan instans EC2 yang digunakan oleh EC2 Image Builder untuk menghasilkan AMI kustom.

Warning

Untuk menggunakan opsi ini, pastikan [AWS ParallelCluster pengguna](#) diberikan `iam:AttachRolePolicy` dan `iam:DetachRolePolicy` izin untuk kebijakan IAM yang perlu dilampirkan.

Batas izin

[Build](#) / [Iam](#) / [PermissionsBoundary](#)

Parameter ini memaksa AWS ParallelCluster untuk melampirkan kebijakan IAM yang diberikan sebagai a `PermissionsBoundary` ke semua peran IAM yang dibuat sebagai bagian dari build AMI kustom.

Lihat [PermissionsBoundarymodus](#) daftar kebijakan yang diperlukan untuk menggunakan fungsionalitas tersebut.

Konfigurasi jaringan

AWS ParallelCluster menggunakan Amazon Virtual Private Cloud (VPC) untuk jaringan. VPC menyediakan platform jaringan yang fleksibel dan dapat dikonfigurasi di mana Anda dapat menerapkan cluster.

VPC harus memiliki `DNS Resolution = yes`, `DNS Hostnames = yes` dan opsi DHCP dengan nama domain yang benar untuk Wilayah. Default DHCP Option Set sudah menentukan `AmazonProvidedDNS` yang diperlukan. Jika menentukan lebih dari satu server nama domain, lihat [set opsi DHCP](#) di Panduan Pengguna Amazon VPC.

AWS ParallelCluster mendukung konfigurasi tingkat tinggi berikut:

- Satu subnet untuk node head dan compute.
- Dua subnet, dengan node kepala dalam satu subnet publik, dan menghitung node dalam subnet pribadi. Subnet dapat berupa yang baru atau yang sudah ada.

Semua konfigurasi ini dapat beroperasi dengan atau tanpa alamat IP publik. AWS ParallelCluster juga dapat digunakan untuk menggunakan proxy HTTP untuk semua AWS permintaan. Kombinasi konfigurasi ini menghasilkan banyak skenario penyebaran. Misalnya, Anda dapat mengkonfigurasi subnet publik tunggal dengan semua akses melalui internet. Atau, Anda dapat mengkonfigurasi jaringan yang sepenuhnya pribadi menggunakan AWS Direct Connect dan proxy HTTP untuk semua lalu lintas.

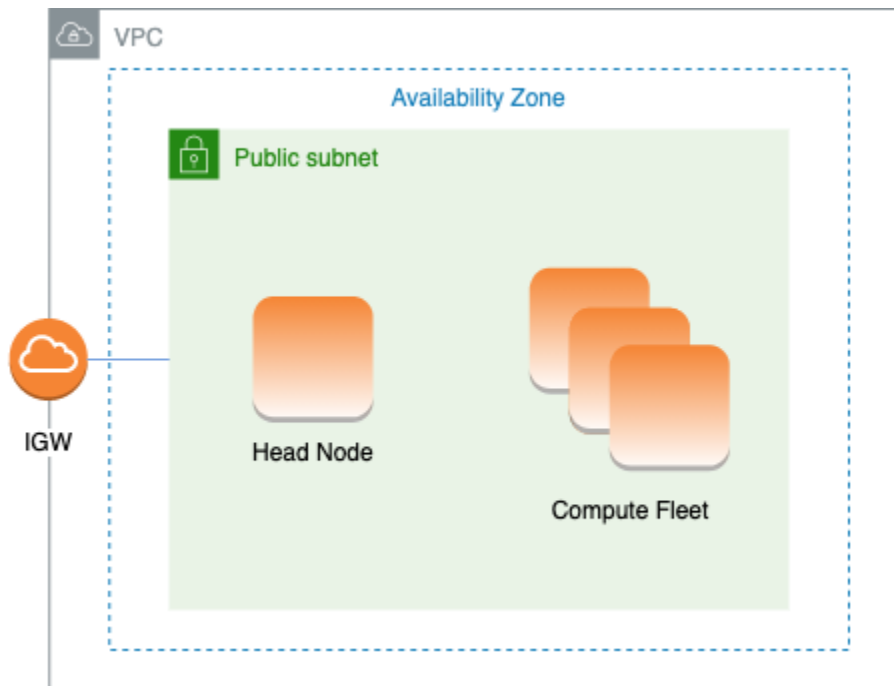
Mulai dari AWS ParallelCluster 3.0.0 adalah mungkin untuk mengkonfigurasi yang berbeda `SecurityGroups`, `AdditionalSecurityGroups` dan `PlacementGroup` pengaturan untuk setiap antrian. Untuk informasi lebih lanjut, lihat [HeadNode/Networking](#) dan [SlurmQueues/Networking](#) dan [AwsBatchQueues/Networking](#).

Untuk ilustrasi beberapa skenario jaringan, lihat diagram arsitektur berikut.

Topik

- [AWS ParallelCluster di subnet publik](#)
- [AWS ParallelCluster menggunakan dua subnet](#)
- [AWS ParallelCluster dalam satu subnet pribadi yang terhubung menggunakan AWS Direct Connect](#)
- [AWS ParallelCluster dengan AWS Batch scheduler](#)
- [AWS ParallelCluster dalam subnet tunggal tanpa akses internet](#)

AWS ParallelCluster di subnet publik



Konfigurasi untuk arsitektur ini memerlukan pengaturan berikut:

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
      Networking:
        SubnetIds:
          - subnet-12345678 # subnet with internet gateway
        #AssignPublicIp: true
```

Dalam konfigurasi ini, semua contoh cluster harus diberi IP publik untuk mendapatkan akses internet. Untuk melakukannya, lakukan hal berikut:

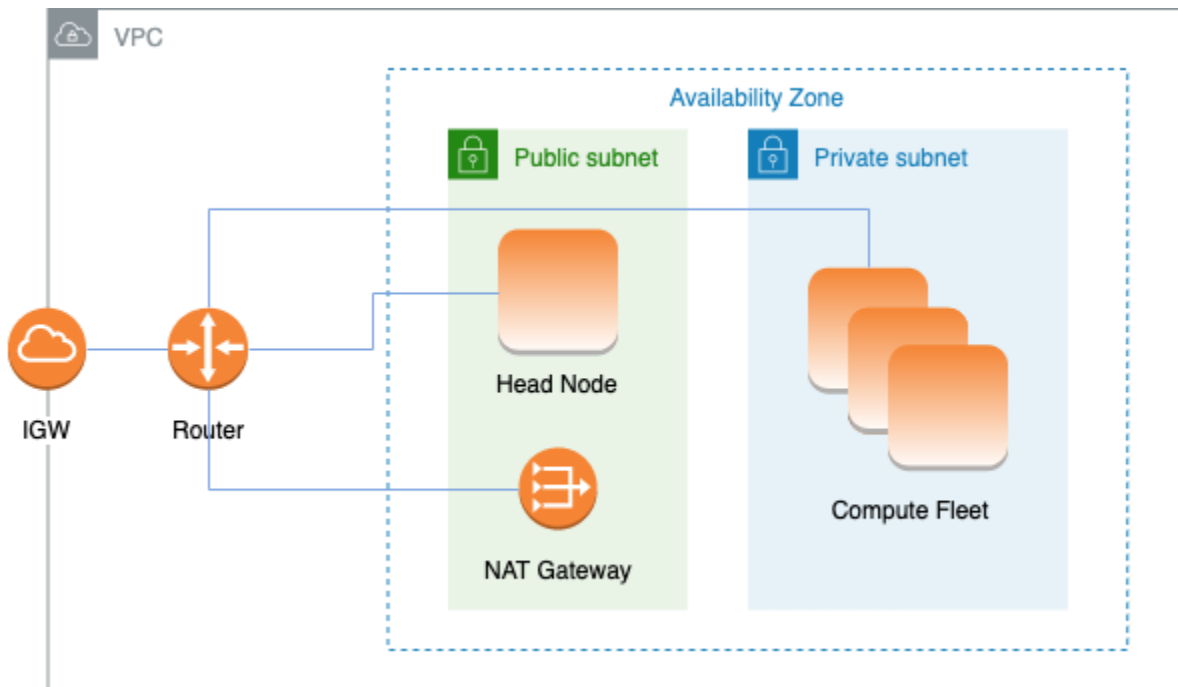
- Pastikan node kepala diberi alamat IP publik dengan menyalakan pengaturan “Aktifkan alamat IPv4 publik secara otomatis” untuk subnet yang digunakan

dalam [HeadNodeNetworking//SubnetId](#) atau dengan menetapkan IP elastis di [HeadNode/Networking/ElasticIp](#).

- Pastikan node komputasi diberi alamat IP publik dengan menyalakan pengaturan “Aktifkan alamat IPv4 publik secara otomatis” untuk subnet yang digunakan dalam [Scheduling//SlurmQueuesNetworking/SubnetIds](#) atau dengan pengaturan [AssignPublicIp: true](#) in [Scheduling/SlurmQueues/Networking](#).
- Jika Anda menentukan jenis p4d instance, atau jenis instance lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan ke node kepala, Anda harus mengatur [HeadNode/Networking/ElasticIp](#) `true` untuk menyediakan akses publik. AWSIP publik hanya dapat ditugaskan ke instance yang diluncurkan dengan satu antarmuka jaringan. Untuk kasus ini, sebaiknya gunakan [gateway NAT](#) untuk menyediakan akses publik ke node komputasi kluster. Untuk informasi selengkapnya tentang alamat IP, lihat [Menugaskan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Anda tidak dapat menentukan jenis p4d atau hp6id instance, atau jenis instance lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan untuk menghitung node karena IPAWS publik hanya dapat ditetapkan ke instance yang diluncurkan dengan satu antarmuka jaringan. Untuk informasi selengkapnya tentang alamat IP, lihat [Menugaskan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya, lihat [Mengaktifkan akses internet](#) di Panduan Pengguna Amazon VPC.

AWS ParallelCluster menggunakan dua subnet



Konfigurasi untuk menggunakan subnet pribadi yang ada untuk instance komputasi memerlukan pengaturan berikut:

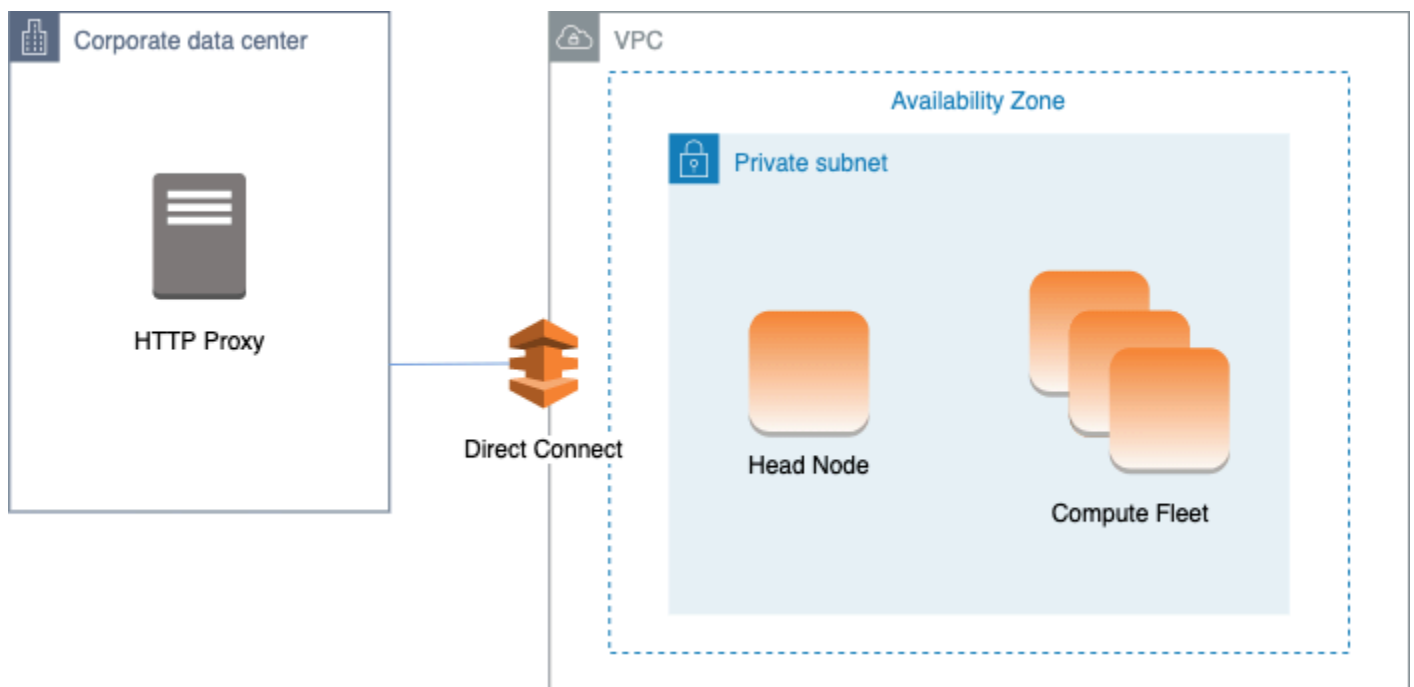
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
      Networking:
        SubnetIds:
          - subnet-23456789 # subnet with NAT gateway
        #AssignPublicIp: false
```

Dalam konfigurasi ini, hanya node kepala cluster yang diperlukan untuk memiliki IP publik yang ditetapkan. Anda dapat mencapai ini dengan menyalakan pengaturan “Aktifkan alamat IPv4 publik secara otomatis” untuk subnet yang digunakan dalam [HeadNode/Networking/SubnetId](#) atau dengan menetapkan IP Elastis di [HeadNode/Networking/ElasticIp](#).

Jika Anda mendefinisikan jenis contoh p4d atau jenis contoh lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan ke node kepala, Anda harus mengatur `HeadNode/Networking/ElasticIp>true` untuk menyediakan akses publik. AWSIP publik hanya dapat ditugaskan ke instance yang diluncurkan dengan satu antarmuka jaringan. Untuk informasi selengkapnya tentang alamat IP, lihat [Menugaskan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Konfigurasi ini memerlukan [gateway NAT](#) atau proxy internal di subnet yang digunakan untuk antrian, untuk memberikan akses internet ke instance komputasi.

AWS ParallelCluster dalam satu subnet pribadi yang terhubung menggunakan AWS Direct Connect



Konfigurasi untuk arsitektur ini memerlukan pengaturan berikut:

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-34567890 # subnet with proxy
  Proxy:
    HttpProxyAddress: http://proxy-address:port
  Ssh:
    KeyName: ec2-key-name
```

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
  Networking:
    SubnetIds:
      - subnet-34567890 # subnet with proxy
    AssignPublicIp: false
    Proxy:
      HttpProxyAddress: http://proxy-address:port
```

Kapan [Scheduling/SlurmQueues/Networking/AssignPublicIp](#) diatur ke `false`, subnet harus diatur dengan benar untuk menggunakan Proxy untuk semua lalu lintas. Akses web diperlukan untuk node head dan compute.

AWS ParallelCluster dengan AWS Batch scheduler

Saat Anda menggunakan `awsbatch` sebagai tipe penjadwal, AWS ParallelCluster buat lingkungan komputasi AWS Batch terkelola. AWS Batch Lingkungan mengelola instans Elastic Container Service (Amazon ECS). Instans ini diluncurkan di subnet yang dikonfigurasi dalam [SubnetIds](#) parameter [AwsBatchQueuesNetworking](#). AWS Batch Agar dapat berfungsi dengan benar, instans kontainer Amazon ECS membutuhkan akses jaringan eksternal untuk berkomunikasi dengan titik akhir layanan Amazon ECS. Ini diterjemahkan ke dalam skenario berikut:

- ID Subnet yang ditentukan untuk antrian menggunakan [gateway NAT](#) untuk mengakses internet. Direkomendasikan pendekatan ini.
- Instance yang diluncurkan di subnet antrian memiliki alamat IP publik dan dapat menjangkau internet melalui Internet Gateway.

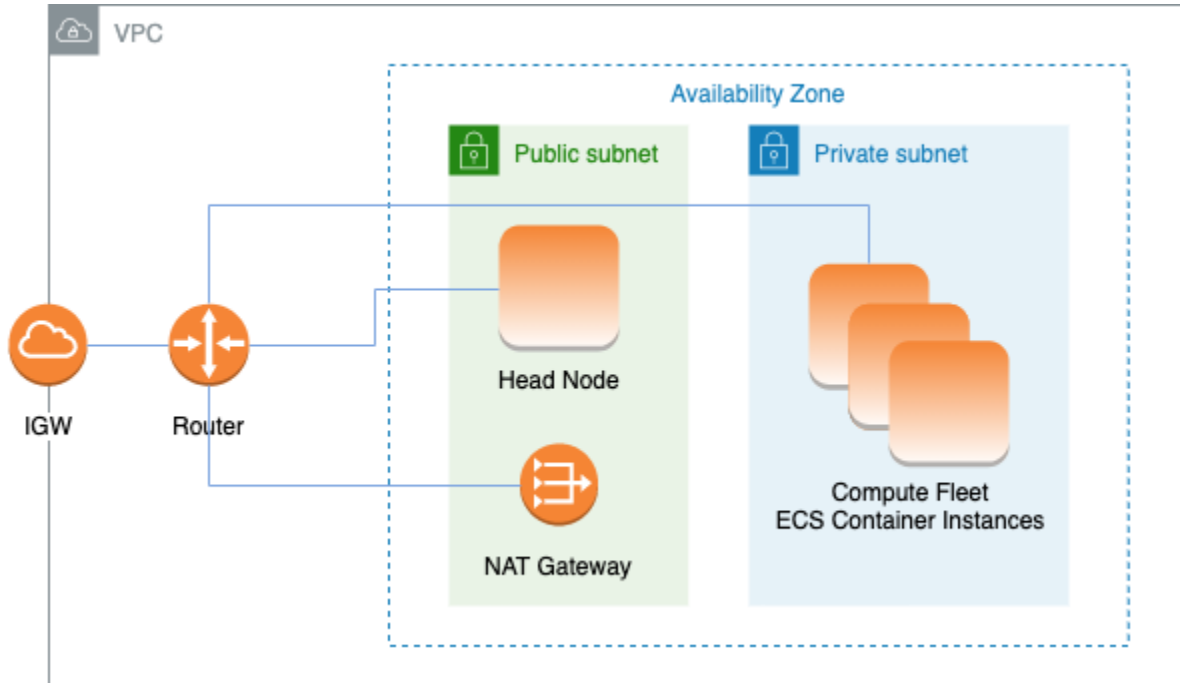
Selain itu, jika Anda tertarik dengan pekerjaan parallel multi-node (dari [AWS Batch dokumen](#)):

AWS Batch pekerjaan parallel multi-node menggunakan mode `awsvpc` jaringan Amazon ECS. Hal ini memberi tugas parallel multisimpul Anda properti jaringan yang sama dengan instans Amazon EC2. Setiap kontainer tugas paralel multisimpul mendapat antarmuka jaringan elastis sendiri, alamat IP privat utama, dan nama host DNS internal. Antarmuka jaringan dibuat di subnet Amazon VPC yang sama dengan sumber daya komputasi host-nya. Grup keamanan yang diterapkan pada sumber daya komputasi Anda juga diterapkan pada antarmuka jaringan.

Saat menggunakan Amazon ECS Task Networking, mode `awsvpc` jaringan tidak menyediakan antarmuka jaringan elastis dengan alamat IP publik untuk tugas yang menggunakan jenis peluncuran

Amazon EC2. Untuk mengakses internet, tugas yang menggunakan jenis peluncuran Amazon EC2 harus diluncurkan di subnet privat yang dikonfigurasi untuk menggunakan gateway NAT.

Anda harus mengkonfigurasi [gateway NAT](#) untuk mengaktifkan cluster menjalankan pekerjaan parallel multi-node.



Semua konfigurasi dan pertimbangan sebelumnya juga berlaku. AWS Batch Berikut ini adalah contoh konfigurasi AWS Batch jaringan.

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway, NAT gateway or proxy
    #ElasticIp: true | false | eip-12345678
    #Proxy:
      #HttpProxyAddress: http://proxy-address:port
  Ssh:
    KeyName: ec2-key-name
  Scheduling:
    Scheduler: awsbatch
  AwsBatchQueues:
    - ...
      Networking:
        SubnetIds:
```



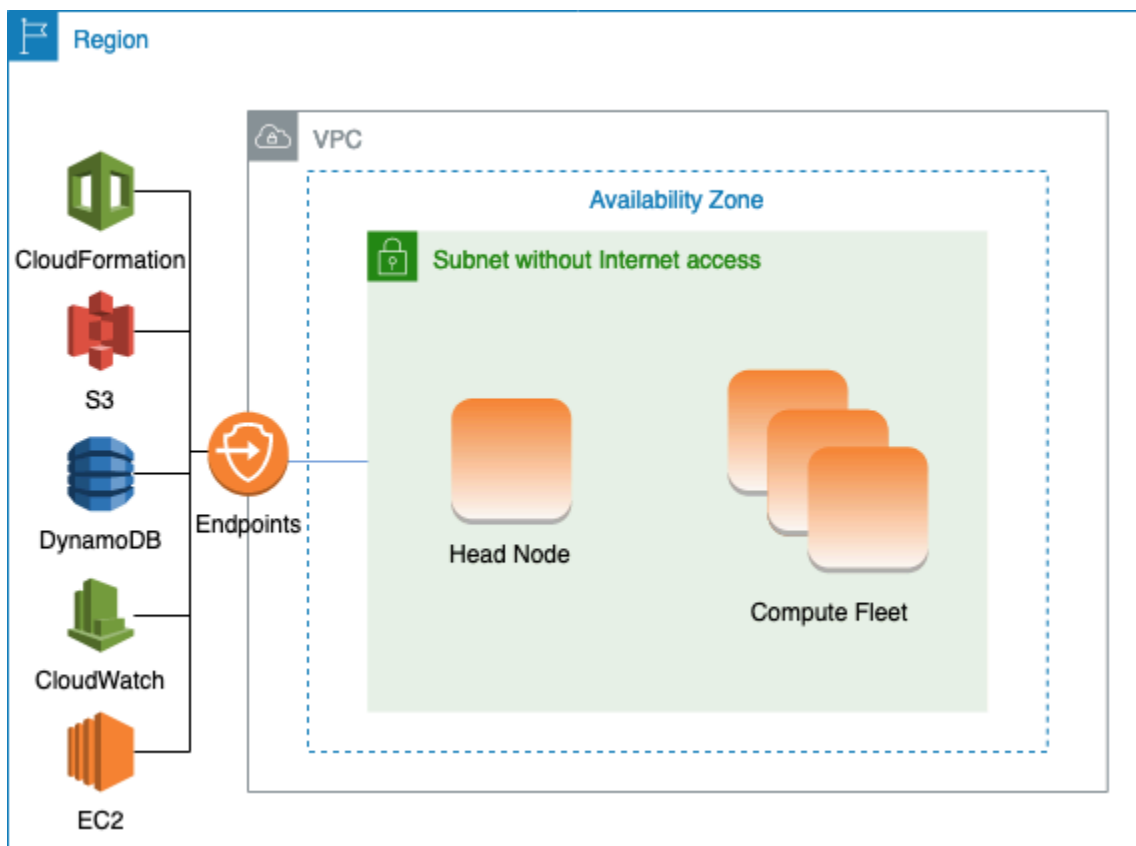
```
- subnet-23456789 # subnet with internet gateway, NAT gateway or proxy
#AssignPublicIp: true | false
```

Di [Networking](#) bagian [Scheduling/AwsBatchQueues/](#), [SubnetIds](#) adalah tipe daftar tetapi, saat ini, hanya satu subnet yang didukung.

Untuk informasi lain, lihat topik berikut:

- [AWS Batch lingkungan komputasi terkelola](#)
- [AWS Batch Tugas paralel paralel paralel multisimpul](#)
- [Jaringan tugas Amazon ECS dengan mode jaringan awsvpc](#)

AWS ParallelCluster dalam subnet tunggal tanpa akses internet



Subnet tanpa akses internet tidak mengizinkan koneksi masuk atau keluar ke internet. AWS ParallelCluster Konfigurasi ini dapat membantu pelanggan yang bersangkutan dengan keamanan lebih meningkatkan keamanan AWS ParallelCluster sumber daya mereka. AWS ParallelCluster node dibangun dari AWS ParallelCluster AMI yang mencakup semua perangkat lunak yang diperlukan

untuk menjalankan cluster tanpa akses internet. Dengan cara ini, AWS ParallelCluster dapat membuat dan mengelola cluster dengan node yang tidak memiliki akses internet.

Pada bagian ini, Anda mempelajari tentang cara mengonfigurasi kluster. Anda juga belajar tentang batasan dalam menjalankan cluster tanpa akses internet.

Mengkonfigurasi titik akhir VPC

Untuk memastikan berfungsinya cluster, node cluster harus dapat berinteraksi dengan sejumlah AWS Layanan.

Buat dan konfigurasi [endpoint VPC](#) berikut sehingga node kluster dapat berinteraksi dengan AWS Layanan, tanpa akses internet:

Commercial and AWS GovCloud (US) partitions

Layanan	Nama layanan	Tipe
Amazon CloudWatch	com.amazonaws. <i>wilayah-id</i> .logs	Antarmuka
AWS CloudFormation	com.amazonaws. <i>wilayah-id</i> .cloudformation	Antarmuka
Amazon EC2	com.amazonaws. <i>wilayah-id</i> .ec2	Antarmuka
Amazon S3	com.amazonaws. <i>wilayah-id</i> .s3	Gateway
Amazon DynamoDB	com.amazonaws. <i>wilayah-id</i> .dynamodb	Gateway
AWS Secrets Manager**	com.amazonaws. <i>wilayah-id</i> .secretsmanager	Antarmuka

China partition

Layanan	Nama layanan	Tipe
Amazon CloudWatch	com.amazonaws. <i>wilayah-id</i> .logs	Antarmuka
AWS CloudFormation	cn.com.amazonaws. <i>wilayah-id</i> .cloudformation	Antarmuka
Amazon EC2	cn.com.amazonaws. <i>wilayah-id</i> .ec2	Antarmuka
Amazon S3	com.amazonaws. <i>wilayah-id</i> .s3	Gateway
Amazon DynamoDB	com.amazonaws. <i>wilayah-id</i> .dynamodb	Gateway
AWS Secrets Manager**	com.amazonaws. <i>wilayah-id</i> .secretsmanager	Antarmuka

** Endpoint ini hanya diperlukan ketika [DirectoryService](#) diaktifkan, jika tidak maka opsional.

Semua instance di VPC harus memiliki grup keamanan yang tepat untuk berkomunikasi dengan titik akhir. Anda dapat melakukan ini dengan menambahkan grup keamanan ke [AdditionalSecurityGroupsAdditionalSecurityGroups](#) bawah [HeadNode](#) dan di bawah [SlurmQueues](#) konfigurasi. Misalnya, jika titik akhir VPC dibuat tanpa secara eksplisit menentukan grup keamanan, grup keamanan default dikaitkan dengan titik akhir. Dengan menambahkan grup keamanan default `AdditionalSecurityGroups`, Anda mengaktifkan komunikasi antara kluster dan titik akhir.

Note

Saat Anda menggunakan kebijakan IAM untuk membatasi akses ke titik akhir VPC, Anda harus menambahkan yang berikut ke titik akhir Amazon S3 VPC:

```
PolicyDocument:
```

```

Version: 2012-10-17
Statement:
  - Effect: Allow
    Principal: "*"
    Action:
      - "s3:PutObject"
    Resource:
      - !Sub "arn:${AWS::Partition}:s3::cloudformation-waitcondition-
        ${AWS::Region}/*"

```

Nonaktifkan Route 53 dan gunakan nama host EC2

Saat membuat cluster Slurm, AWS ParallelCluster buat zona host Route 53 pribadi yang digunakan untuk menyelesaikan nama-nama host node komputasi kustom, seperti `{queue_name}-{st|dy}-{compute_resource}-{N}`. Karena Route 53 tidak mendukung titik akhir VPC, fitur ini harus dinonaktifkan. Selain itu, AWS ParallelCluster harus dikonfigurasi untuk menggunakan nama host EC2 default, seperti `ip-1-2-3-4`. Terapkan pengaturan berikut ke konfigurasi klaster Anda:

```

...
Scheduling:
  ...
  SlurmSettings:
    Dns:
      DisableManagedDns: true
      UseEc2Hostnames: true

```

Warning

Untuk klaster yang dibuat dengan [SlurmSettingsDns//DisableManagedDns](#) dan [UseEc2Hostnames](#) diatur ke `true`, `SlurmNodeName` tidak diselesaikan oleh DNS. Gunakan `SlurmNodeHostName` sebagai gantinya.

Note

Catatan ini tidak relevan dimulai dengan AWS ParallelCluster versi 3.3.0.
Untuk versi AWS ParallelCluster yang didukung sebelum 3.3.0:

Ketika `UseEc2Hostnames` diatur ke `true`, file konfigurasi Slurm diatur dengan `prolog` dan `epilog` script AWS ParallelCluster:

- `prolog` berjalan untuk menambahkan info node ke `/etc/hosts` node komputasi ketika setiap pekerjaan dialokasikan.
- `epilog` berjalan untuk membersihkan isi yang ditulis oleh `prolog`.

Untuk menambahkan kustom `prolog` atau `epilog` skrip, menambahkannya ke `/opt/slurm/etc/pcluster/prolog.d/` atau `/opt/slurm/etc/pcluster/epilog.d/` folder masing-masing.

Konfigurasi cluster

Pelajari cara mengkonfigurasi klaster Anda agar berjalan di subnet tanpa koneksi ke internet.

Konfigurasi untuk arsitektur ini memerlukan pengaturan berikut:

```
# Note that all values are only provided as examples
...
HeadNode:
  ...
  Networking:
    SubnetId: subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
    endpoints
    AdditionalSecurityGroups:
      - sg-abcdef01234567890 # optional, the security group that enables the
      communication between the cluster and the VPC endpoints
  Scheduling:
    Scheduler: slurm # Cluster in a subnet without internet access is supported only when
    the scheduler is Slurm.
    SlurmSettings:
      Dns:
        DisableManagedDns: true
        UseEc2Hostnames: true
    SlurmQueues:
      - ...
      Networking:
        SubnetIds:
          - subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
          endpoints attached
```

AdditionalSecurityGroups:

```
- sg-1abcdef01234567890 # optional, the security group that enables the communication between the cluster and the VPC endpoints
```

- [SubnetId\(s\)](#): Subnet tanpa akses internet.

Untuk mengaktifkan komunikasi antara AWS ParallelCluster dan AWS Layanan, VPC subnet harus memiliki titik akhir VPC yang terpasang. Sebelum Anda membuat klaster Anda, verifikasi bahwa [alamat IPv4 publik auto-assign dinonaktifkan](#) di subnet untuk memastikan bahwa `pc1uster` perintah memiliki akses ke klaster.

- [AdditionalSecurityGroups](#): Grup keamanan yang memungkinkan komunikasi antara cluster dan titik akhir VPC.

Opsional:

- Jika titik akhir VPC dibuat tanpa secara eksplisit menentukan grup keamanan, maka terkait dengan grup keamanan default VPC. Oleh karena itu, berikan grup keamanan default ke `AdditionalSecurityGroups`.
- Jika grup keamanan khusus digunakan saat membuat klaster dan/atau titik akhir VPC, `AdditionalSecurityGroups` tidak perlu selama grup keamanan khusus mengaktifkan komunikasi antara klaster dan titik akhir VPC.
- [Scheduler](#): Cluster scheduler.

`slurm` adalah satu-satunya nilai yang valid. Hanya penjadwal `slurm` yang mendukung cluster di subnet tanpa akses internet.

- [SlurmSettings](#): Pengaturan `slurm`.

Lihat bagian sebelumnya Nonaktifkan Route53 dan gunakan nama host EC2.

Keterbatasan

- Menghubungkan ke node kepala melalui SSH atau NICE DCV: Saat menghubungkan ke cluster, pastikan klien koneksi dapat mencapai node kepala cluster melalui alamat IP pribadinya. Jika klien tidak berada dalam VPC yang sama dengan head node, gunakan instance proxy di subnet publik VPC. Persyaratan ini berlaku untuk koneksi SSH dan DCV. IP publik dari node kepala tidak dapat diakses jika subnet tidak memiliki akses internet. `dcv-connect` Perintah `pc1uster ssh` dan menggunakan IP publik jika ada atau IP pribadi. Sebelum Anda membuat klaster Anda, verifikasi

bahwa [alamat IPv4 publik auto-assign dinonaktifkan](#) di subnet untuk memastikan bahwa `pcluster` perintah memiliki akses ke klaster.

Contoh berikut menunjukkan bagaimana Anda dapat terhubung ke sesi DCV yang berjalan di node kepala klaster Anda. Anda terhubung melalui instans EC2. Instance berfungsi sebagai server NICE DCV untuk PC Anda dan sebagai klien untuk node kepala di subnet pribadi.

Connect melalui DCV melalui instance proxy di subnet publik:

1. Buat instans EC2 di subnet publik, yang berada di VPC yang sama dengan subnet cluster.
 2. Pastikan bahwa klien NICE DCV dan server diinstal pada instans EC2 Anda.
 3. Lampirkan KebijakanAWS ParallelCluster Pengguna ke instans EC2 proxy. Untuk informasi selengkapnya, lihat [AWS ParallelCluster contoh kebijakan pcluster pengguna](#).
 4. InstallAWS ParallelCluster pada instans EC2.
 5. Connect melalui DCV ke instans EC2 proxy.
 6. Gunakan `pcluster dcv-connect` perintah pada instance proxy untuk terhubung ke cluster di dalam subnet tanpa akses internet.
- Berinteraksi denganAWS layanan lain: Hanya layanan yang diwajibkan secara ketat oleh yangAWS ParallelCluster tercantum di atas. Jika klaster Anda harus berinteraksi dengan layanan lain, buat endpoint VPC yang sesuai.

Node masuk

Mulai dari versi 3.7.0,AWS ParallelClusteradministrator cluster dapat menyediakan node login yang dapat digunakan untuk menyediakan akses ke pengguna untuk menjalankan pekerjaan vs langsung mengakses node kepala cluster. Pengguna klaster dengan izin yang sesuai dapat menggunakan Active Directory atau kredensi ssh mereka untuk masuk, mengirimkan, dan mengelola pekerjaan mereka. Akibatnya, manajemen cluster dapat ditingkatkan dan kemungkinan menghabiskan sumber daya node kepala yang dibutuhkan oleh Slurm untuk mengelola cluster dapat diminimalkan. Pengguna yang masuk juga akan memiliki akses ke semua penyimpanan bersama cluster yang dipasang pada node login. Jika node login perlu dihentikan, pengguna akan diberi tahu terlebih dahulu melalui sesi shell aktif yang mereka gunakan.

Node login ditentukan sebagai pool di mana pool mendefinisikan sekelompok node login yang memiliki konfigurasi sumber daya yang sama. Semua node login di kolam dikonfigurasi untuk menjadi bagian dari[penyeimbang beban jaringan](#)yang memungkinkan pendistribusian sesi di seluruh

node login dengan cara round-robin. Implementasi saat ini memungkinkan untuk menentukan satu kumpulan node login yang berisi beberapa node Login.

Keamanan

Node login mewarisi `AllowedIPs` pengaturan [AllowedIps](#) dari simpul kepala. Dengan cara ini, administrator cluster dapat membatasi postur keamanan cluster dengan menentukan CIDR sumber atau daftar awalan dari mana koneksi SSH diizinkan.

Dalam implementasi ini, akses ke node kepala tidak dibatasi secara otomatis saat mengaktifkan node login. Jika diperlukan, administrator cluster dapat membatasi akses ini memperbarui konfigurasi ssh node head menggunakan perintah Linux standar. Ini juga dapat dicapai dengan menentukan Grup Keamanan khusus pada node kepala dengan menggunakan `AdditionalSecurityGroup` pengaturan di bagian simpul kepala ParallelCluster File YAMLUntuk menolak koneksi dari pengguna yang tidak sah.

Jaringan

Node login disediakan dengan alamat koneksi tunggal ke penyeimbang beban jaringan yang dikonfigurasi untuk kumpulan node login. Pengaturan konektivitas alamat didasarkan pada jenis subnet yang ditentukan dalam konfigurasi Pool node Login.

- Jika subnet bersifat pribadi, alamatnya akan bersifat pribadi dan, untuk memberikan akses ke node login, administrator cluster harus menyediakan host bastion.
- Jika subnet bersifat publik, alamatnya akan bersifat publik

Semua permintaan koneksi dikelola oleh Network Load Balancer menggunakan routing round-robin.

Penyimpanan

Semua penyimpanan bersama dikonfigurasi pada cluster menggunakan ParallelCluster termasuk penyimpanan terkelola akan dipasang pada semua node login.

Ambil informasi node login

Untuk mengambil alamat koneksi tunggal yang disediakan untuk mengakses node login, administrator cluster dapat menjalankan [describe-cluster](#) perintah. Perintah ini juga akan memberikan informasi lebih lanjut tentang status node login.

Node login adalah tipe node baru yang didukung oleh ParallelCluster yang dapat ditentukan dengan [describe-cluster-instances](#) perintah saat menanyakan status tipe node tertentu.

Ketersediaan alamat koneksi tunggal ke kumpulan node Login tidak mencegah akses langsung ke node login tertentu. Namun, tidak disarankan untuk menggunakan koneksi langsung untuk menghindari peringatan dari klien ssh. Klien ssh menyimpan pengidentifikasi host secara lokal untuk setiap alamat target. Karena pengenalan host spesifik per kumpulan, penggunaan IP yang berbeda dan/atau alamat koneksi tunggal mungkin memiliki pengenalan host yang sama yang terkait dengan alamat target yang berbeda: ini dapat menyebabkan peringatan dari klien ssh karena pengidentifikasi host yang sama dikaitkan dengan beberapa target.

Properti Imds

Akses ke IMDS node login (dan kredensial profil instance) dibatasi untuk pengguna root, pengguna administratif cluster (`pc-cluster-admin` secara default) dan pengguna default khusus sistem operasi (`ec2-user` di Amazon Linux 2 dan RedHat, `ubuntu` di Ubuntu 18.04, `centos` di CentOS 7).

Untuk membatasi akses IMDS, AWS ParallelCluster mengelola rantai `iptables`.

Note

Setiap kustomisasi `iptables` atau `ip6tables` aturan dapat mengganggu mekanisme yang digunakan untuk membatasi akses IMDS pada node login. Lihat juga [Imds property setting](#).

Siklus hidup Node Login

Saat ini, tidak ada perintah khusus untuk menghentikan dan memulai node login di kolam. Untuk menghentikan node login di kolam, administrator cluster harus memperbarui konfigurasi cluster yang menentukan nol pada jumlah node login (`Count: 0`) dan kemudian jalankan [`pcluster.update-cluster-v3`](#) perintah.

Note

Pengguna yang masuk akan diberi tahu tentang penghentian instance tertentu dan tentang periode `gracetime` terkait. Selama periode `gracetime` tidak ada koneksi baru yang diizinkan kecuali yang dari [pengguna default cluster](#). Pesan yang ditampilkan dapat disesuaikan oleh administrator cluster dari node kepala atau dari node login mengedit file `/opt/parallelcluster/shared_login_nodes/loginmgtd_config.json`.

Untuk memulai kumpulan node login, administrator cluster harus mengembalikan yang sebelumnya `Count` nilai dalam konfigurasi cluster dan kemudian jalankan `update-cluster` perintah.

Izin yang diperlukan untuk menjalankan kumpulan node login

Untuk mengelola kumpulan node login, administrator cluster harus memiliki izin tambahan berikut:

```

- Action:
  - autoscaling:DeleteAutoScalingGroup
  - autoscaling:DeleteLifecycleHook
  - autoscaling:Describe*
  - autoscaling:PutLifecycleHook
  - autoscaling:UpdateAutoScalingGroup
  - elasticloadbalancing:CreateListener
  - elasticloadbalancing:CreateTargetGroup
  - elasticloadbalancing>DeleteListener
  - elasticloadbalancing>DeleteLoadBalancer
  - elasticloadbalancing>DeleteTargetGroup
  - elasticloadbalancing:Describe*
  - elasticloadbalancing:ModifyLoadBalancerAttributes
Resource: '*'
Condition:
  ForAllValues:StringEquals:
    aws:TagKeys: [ "parallelcluster:cluster-name" ]
- Action:
  - autoscaling:CreateAutoScalingGroup
  - elasticloadbalancing:AddTags
  - elasticloadbalancing:CreateLoadBalancer
Resource: '*'
Effect: Allow

```

Tindakan bootstrap kustom

Jika Anda menentukan pengaturan `OnNodeStart` konfigurasi `HeadNodeCustomActions`//, AWS ParallelCluster jalankan kode arbitrer segera setelah node dimulai. Jika Anda menentukan pengaturan `OnNodeConfigured` konfigurasi `HeadNodeCustomActions`//, AWS ParallelCluster jalankan kode setelah konfigurasi node selesai dengan benar.

Dimulai dengan AWS ParallelCluster versi 3.4.0, kode dapat dijalankan setelah pembaruan node kepala, jika Anda menentukan pengaturan `OnNodeUpdated` konfigurasi `HeadNodeCustomActions`//.

Dalam kebanyakan kasus, kode ini disimpan di Amazon Simple Storage Service (Amazon S3) dan diakses melalui koneksi HTTPS. Kode dijalankan sebagai `root` dan dapat dalam bahasa skrip apa pun yang didukung oleh OS cluster. Seringkali kode dalam Bash atau Python.

Note

Dimulai dengan AWS ParallelCluster versi 3.7.0, [ImdsSupport](#) pengaturan kluster [Imds/default](#) adalah `v2.0`

Saat Anda membuat cluster baru untuk meningkatkan ke versi 3.7.0 dan versi yang lebih baru, perbarui skrip tindakan bootstrap kustom Anda agar kompatibel dengan IMDSv2 atau atur [Imds/ImdsSupport](#) ke `v1.0` dalam file konfigurasi cluster Anda.

Warning

Anda bertanggung jawab untuk mengonfigurasi skrip dan argumen kustom seperti yang dijelaskan dalam model [tanggung jawab bersama](#). Verifikasi bahwa skrip dan argumen bootstrap kustom Anda berasal dari sumber yang Anda percayai memiliki akses penuh ke node cluster Anda.

Warning

AWS ParallelCluster tidak mendukung penggunaan variabel internal yang disediakan melalui `/etc/parallelcluster/cfnconfig` file. File ini mungkin akan dihapus sebagai bagian dari rilis future.

`OnNodeStart` tindakan dipanggil sebelum tindakan bootstrap penerapan node dimulai, seperti mengonfigurasi NAT, Amazon Elastic Block Store (Amazon EBS) atau penjadwal.

`OnNodeStart` Tindakan bootstrap mungkin termasuk memodifikasi penyimpanan, menambahkan pengguna tambahan, dan menambahkan paket.

Note

Jika Anda mengkonfigurasi [DirectoryService](#) dan [OnNodeStart](#) skrip [HeadNodeCustomActions](#)//untuk cluster Anda, AWS ParallelCluster mengkonfigurasi [DirectoryService](#) dan memulai ulang `ssd`, sebelum menjalankan skrip. `OnNodeStart`

`OnNodeConfigured` tindakan dipanggil setelah proses bootstrap node selesai.

`OnNodeConfigured` tindakan melayani tindakan terakhir yang terjadi sebelum sebuah instance dianggap sepenuhnya dikonfigurasi dan selesai. Beberapa `OnNodeConfigured` tindakan termasuk mengubah pengaturan penjadwal, memodifikasi penyimpanan, dan memodifikasi paket. Anda dapat meneruskan argumen ke skrip dengan menentukannya selama konfigurasi.

`OnNodeUpdated` tindakan dipanggil setelah pembaruan node kepala selesai dan penjadwal dan penyimpanan bersama diselaraskan dengan perubahan konfigurasi cluster terbaru.

Kapan `OnNodeStart` atau tindakan `OnNodeConfigured` kustom berhasil, keberhasilan ditunjukkan dengan kode keluar nol (0). Kode keluar lainnya menunjukkan instance bootstrap gagal.

Ketika tindakan `OnNodeUpdated` kustom berhasil, keberhasilan ditandai dengan kode keluar nol (0). Kode keluar lainnya menunjukkan pembaruan gagal.

Note

Jika Anda mengkonfigurasi [OnNodeUpdated](#), Anda harus mengembalikan `OnNodeUpdated` tindakan secara manual ke keadaan sebelumnya pada kegagalan pembaruan.

Jika tindakan `OnNodeUpdated` kustom gagal, pembaruan akan kembali ke status sebelumnya. Namun, `OnNodeUpdated` tindakan ini hanya dijalankan pada waktu pembaruan dan bukan pada waktu rollback stack.

Anda dapat menentukan skrip yang berbeda untuk node kepala dan untuk setiap antrian, di bagian [CustomActions](#) konfigurasi [HeadNode/CustomActions](#) dan [SchedulingSlurmQueues](#)/. [OnNodeUpdated](#) hanya dapat dikonfigurasi di HeadNode bagian.

Note

Sebelum AWS ParallelCluster versi 3.0, tidak mungkin untuk menentukan skrip yang berbeda untuk node head dan compute. Silakan merujuk ke [Pindah dari AWS ParallelCluster 2.x ke 3.x](#).

Topik

- [Konfigurasi](#)
- [Argumen](#)
- [Contoh cluster dengan tindakan bootstrap kustom](#)
- [Contoh memperbarui skrip bootstrap kustom untuk IMDSv2](#)
- [Contoh memperbarui konfigurasi untuk IMDSv1](#)

Konfigurasi

Pengaturan konfigurasi berikut digunakan untuk mendefinisikan

[HeadNode/CustomActions/OnNodeStart](#) & [OnNodeConfigured](#) & [OnNodeUpdated](#) dan [Scheduling/CustomActions/OnNodeStart](#) & [OnNodeConfigured](#) tindakan dan argumen.

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URL. This is run before any of the bootstrap scripts are run
      Script: s3://bucket-name/on-node-start.sh
      Args:
        - arg1
    OnNodeConfigured:
      # Script URL. This is run after all the bootstrap scripts are run
      Script: s3://bucket-name/on-node-configured.sh
      Args:
        - arg1
    OnNodeUpdated:
      # Script URL. This is run after the head node update is completed.
      Script: s3://bucket-name/on-node-updated.sh
      Args:
        - arg1
  # Bucket permissions
```

```
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false
Scheduling:
  Scheduler: slurm
  [...]
  SlurmQueues:
    - Name: queue1
      [...]
    CustomActions:
      OnNodeStart:
        Script: s3://bucket-name/on-node-start.sh
        Args:
          - arg1
      OnNodeConfigured:
        Script: s3://bucket-name/on-node-configured.sh
        Args:
          - arg1
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false
```

Menggunakan Sequence pengaturan (ditambahkan dalam AWS ParallelCluster versi 3.6.0):

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, before any of the bootstrap scripts are run.
      Sequence:
        - Script: s3://bucket-name/on-node-start1.sh
          Args:
            - arg1
        - Script: s3://bucket-name/on-node-start2.sh
          Args:
            - arg1
        [...]
    OnNodeConfigured:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, after all the bootstrap scripts are run.
```

```
Sequence:
  - Script: s3://bucket-name/on-node-configured1.sh
  Args:
    - arg1
  - Script: s3://bucket-name/on-node-configured2.sh
  Args:
    - arg1
  [...]
OnNodeUpdated:
  # Script URLs. The scripts are run in the same order as listed in the
  # configuration, after the head node update is completed.
  Sequence:
    - Script: s3://bucket-name/on-node-updated1.sh
    Args:
      - arg1
    - Script: s3://bucket-name/on-node-updated2.sh
    Args:
      - arg1
    [...]
# Bucket permissions
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false
Scheduling:
  Scheduler: slurm
  [...]
  SlurmQueues:
    - Name: queue1
    [...]
  CustomActions:
    OnNodeStart:
      # Script URLs. The scripts are run in the same order as listed in the
      # configuration, before any of the bootstrap scripts are run
      Sequence:
        - Script: s3://bucket-name/on-node-start1.sh
        Args:
          - arg1
        - Script: s3://bucket-name/on-node-start2.sh
        Args:
          - arg1
        [...]
    OnNodeConfigured:
```

```

    # Script URLs. The scripts are run in the same order as listed in the
    configuration, after all the bootstrap scripts are run
    Sequence:
      - Script: s3://bucket-name/on-node-configured1.sh
        Args:
          - arg1
      - Script: s3://bucket-name/on-node-configured2.sh
        Args:
          - arg1
      [...]
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false

```

SequencePengaturan ditambahkan dimulai dengan AWS ParallelCluster versi 3.6.0. Saat Anda menentukanSequence, Anda dapat membuat daftar beberapa skrip untuk tindakan kustom. AWS ParallelClusterterus mendukung konfigurasi tindakan khusus dengan satu skrip, tanpa menyertakanSequence.

AWS ParallelClustertidak mendukung termasuk skrip tunggal dan Sequence untuk tindakan kustom yang sama. Misalnya, AWS ParallelCluster gagal jika Anda menentukan konfigurasi berikut.

```

[...]
  CustomActions:
    OnNodeStart:
      # Script URL. This is run before any of the bootstrap scripts are run
      Script: s3://bucket-name/on-node-start.sh
        Args:
          - arg1
      # Script URLs. The scripts are run in the same order as listed in the
      configuration, before any of the bootstrap scripts are run.
      Sequence:
        - Script: s3://bucket-name/on-node-start1.sh
          Args:
            - arg1
        - Script: s3://bucket-name/on-node-start2.sh
          Args:
            - arg1
  [...]

```


Argumen

Note

Dalam AWS ParallelCluster 2.x \$1 argumennya adalah yang dicadangkan, untuk menyimpan URL skrip kustom. Jika Anda ingin menggunakan kembali skrip bootstrap khusus yang dibuat untuk AWS ParallelCluster 2.x dengan AWS ParallelCluster 3.x, Anda perlu menyesuaikannya dengan mempertimbangkan pergeseran argumen. Silakan merujuk ke [Pindah dari AWS ParallelCluster 2.x ke 3.x](#).

Contoh cluster dengan tindakan bootstrap kustom

Langkah-langkah berikut membuat skrip sederhana untuk dieksekusi setelah node dikonfigurasi, yang menginstal R, curl dan wget paket di node cluster.

1. Membuat skrip.

```
#!/bin/bash
echo "The script has $# arguments"
for arg in "$@"
do
    echo "arg: ${arg}"
done
yum -y install "${@:1}"
```

2. Unggah skrip dengan izin yang benar ke Amazon S3. Jika izin baca publik tidak sesuai untuk Anda, gunakan bagian [SlurmQueues](#) konfigurasi [HeadNodeIam/S3Access](#) dan [Scheduling/](#). Untuk informasi selengkapnya, lihat [Bekerja dengan Amazon S3](#).

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

Important

Jika skrip diedit di Windows, akhiran baris harus diubah dari CRLF ke LF sebelum skrip diunggah ke Amazon S3.

3. Perbarui AWS ParallelCluster konfigurasi untuk menyertakan OnNodeConfigured tindakan baru.

```
CustomActions:
OnNodeConfigured:
  Script: https://<bucket-name>.s3.<region>.amazonaws.com/myscript.sh
  Args:
    - "R"
    - "curl"
    - "wget"
```

Jika bucket tidak memiliki izin baca publik, gunakan s3 sebagai protokol URL.

```
CustomActions:
OnNodeConfigured:
  Script: s3://<bucket-name>/myscript.sh
  Args:
    - "R"
    - "curl"
    - "wget"
```

4. Luncurkan cluster.

```
$ pcluster create-cluster --cluster-name mycluster \
  --region <region> --cluster-configuration config-file.yaml
```

5. Verifikasi output.

- Jika Anda menambahkan tindakan kustom ke HeadNode konfigurasi, masuk ke node kepala dan periksa `cfn-init.log` file yang terletak di `/var/log/cfn-init.log` dengan menjalankan perintah berikut:

```
$ less /var/log/cfn-init.log
2021-09-03 10:43:54,588 [DEBUG] Command run
postinstall output: The script has 3 arguments
arg: R
arg: curl
arg: wget
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

- Jika Anda menambahkan tindakan kustom ke SlurmQueues pengaturan, periksa `cloud-init.log` lokasi di `/var/log/cloud-init.log` dalam node komputasi. Gunakan CloudWatch untuk melihat log ini.

Anda dapat melihat kedua log ini di CloudWatch konsol Amazon. Untuk informasi selengkapnya, lihat [Integrasi dengan Amazon CloudWatch Logs](#).

Contoh memperbarui skrip bootstrap kustom untuk IMDSv2

Dalam contoh berikut, kami memperbarui skrip tindakan bootstrap kustom yang digunakan dengan IMDSv1 untuk digunakan dengan IMDSv2. Skrip IMDSv1 mengambil metadata ID AMI instans EC2.

```
#!/bin/bash
AMI_ID=$(curl http://169.254.169.254/latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

Berikut ini menunjukkan skrip tindakan bootstrap kustom yang dimodifikasi agar kompatibel dengan IMDSv2.

```
#!/bin/bash
AMI_ID=$(TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
    && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

Untuk informasi selengkapnya, lihat [Mengambil metadata instans](#) di Panduan Pengguna EC2 untuk Instans Linux.

Contoh memperbarui konfigurasi untuk IMDSv1

Berikut ini adalah contoh konfigurasi cluster yang mendukung IMDSv1 saat menggunakan AWS ParallelCluster versi 3.7.0 dan yang lebih lama.

```
Region: us-east-1
Imds:
  ImdsSupport: v1.0
Image:
  Os: alinux2
HeadNode:
```

```

InstanceType: t2.micro
Networking:
  SubnetId: subnet-abcdef01234567890
Ssh
  KeyName: key-name
CustomActions:
  OnNodeConfigured:
    Script: Script-path
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CustomActions:
        OnNodeConfigured:
          Script: Script-path
      ComputeResources:
        - Name: t2micro
          Instances:
            - InstanceType: t2.micro
          MinCount: 11
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890

```

Bekerja dengan Amazon S3

Anda dapat mengonfigurasi AWS ParallelCluster akses ke Amazon S3 melalui [S3Access](#) parameter [HeadNodeIam/S3Access](#) dan [SchedulingSlurmQueues/- Name/Iam](#) dalam AWS ParallelCluster konfigurasi.

Contoh

Contoh berikut mengkonfigurasi akses read-only ke semua objek dalam *firstbucket/read_only/* dan membaca/menulis akses ke semua objek di *secondbucket/read_and_write/*.

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: firstbucket
        KeyName: read_only/*

```

```

    EnableWriteAccess: false
  - BucketName: secondbucket
    KeyName: read_and_write/*
    EnableWriteAccess: true
  ...

```

Contoh berikutnya mengkonfigurasi akses read-only ke semua objek dalam folder *read_only/* di bucket (*) di akun.

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: *
        KeyName: read_only/*
        EnableWriteAccess: false
  ...

```

Contoh terakhir mengkonfigurasi akses *read_only* ke semua bucket dan objek di akun.

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: *
  ...

```

Mengerjakan Instans ITSpot

AWS ParallelCluster menggunakan Instans Spot jika Anda telah menetapkan [SlurmQueues/CapacityType](#) atau [AwsBatchQueues/CapacityType](#) ke SPOT dalam file konfigurasi cluster. Instans Spot lebih hemat biaya daripada Instans Sesuai Permintaan, tetapi Instans Spot mungkin terganggu. Mungkin membantu memanfaatkan pemberitahuan gangguan Instans Spot, yang memberikan peringatan dua menit sebelum Amazon EC2 harus menghentikan atau menghentikan Instans Spot Anda. Untuk informasi selengkapnya, lihat [interupsi Instans Spot](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk mempelajari cara [AwsBatchQueues](#) bekerja dengan Instans Spot, lihat [Menghitung Sumber Daya](#) di Panduan AWS Batch Pengguna.

Penjadwal yang AWS ParallelCluster dikonfigurasi menetapkan pekerjaan untuk menghitung sumber daya dalam antrian dengan instans spot dengan cara yang sama menetapkan pekerjaan untuk menghitung sumber daya dalam antrian dengan instans sesuai permintaan.

Saat menggunakan Instans Spot, peran `AWSServiceRoleForEC2Spot` tertaut layanan harus ada di akun Anda. Untuk membuat peran ini di akun Anda menggunakan AWS CLI, jalankan perintah berikut:

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

Untuk informasi selengkapnya, lihat [peran tertaut layanan untuk permintaan Instans Spot](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Bagian berikut menjelaskan tiga skenario di mana Instans Spot dapat terganggu saat menggunakan [SlurmQueues](#).

Skenario 1: Instans Spot tanpa pekerjaan berjalan terganggu

Ketika gangguan ini terjadi, AWS ParallelCluster coba ganti instance jika antrian penjadwal memiliki pekerjaan tertunda yang memerlukan instance tambahan, atau jika jumlah instance aktif lebih rendah dari [SlurmQueues/ComputeResources/MinCount](#). Jika tidak AWS ParallelCluster dapat menyediakan instans baru, maka permintaan untuk instans baru akan diulang secara berkala.

Skenario 2: Instance Spot yang menjalankan pekerjaan node tunggal terganggu

Pekerjaan gagal dengan kode negara `NODE_FAIL`, dan pekerjaan yang `requeued` (kecuali `--no-requeue` ditentukan ketika pekerjaan diserahkan). Jika node adalah node statis, itu diganti. Jika node adalah node dinamis, node diakhiri dan reset. Untuk informasi lebih lanjut tentang `batch`, termasuk `--no-requeue` parameter, lihat [sbatch](#) di dokumentasi Slurm.

Skenario 3: Instance Spot yang menjalankan pekerjaan multi-node terganggu

Pekerjaan gagal dengan kode negara `NODE_FAIL`, dan pekerjaan yang `requeued` (kecuali ditentukan `--no-requeue` ketika pekerjaan diserahkan). Jika node adalah node statis, itu diganti. Jika node adalah node dinamis, node diakhiri dan reset. Node lain yang menjalankan

pekerjaan diakhiri mungkin dialokasikan untuk pekerjaan tertunda lainnya, atau diperkecil setelah [SlurmSettingsScaledownIdleTime](#) dikonfigurasi/waktu telah berlalu.

Untuk informasi lebih lanjut tentang Instans Spot, lihat [Instans Spot](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Penjadwal didukung oleh AWS ParallelCluster

Penjadwal didukung oleh AWS ParallelCluster

AWS ParallelCluster mendukung Slurm dan AWS Batch penjadwal, atur menggunakan pengaturan [Scheduler](#)

Topik

- [Slurm Workload Manager \(slurm\)](#)
- [AWS Batch \(awsbatch\)](#)

Slurm Workload Manager (**slurm**)

Untuk informasi tentang Slurm, lihat <https://slurm.schedmd.com>. Untuk unduhan, lihat <https://github.com/SchedMD/slurm/tags>. Untuk kode sumber, lihat <https://github.com/SchedMD/slurm>.

AWS ParallelCluster versi (s)	SlurmVersi yang didukung
3.9.0	23.11.4
3.8.0	23.02.7
3.7.2	23.02.6
3.7.1	23.02.5
3.7.0	23.02.4
3.6.0, 3.6.1	23.02.2
3.5.0, 3.5.1	22.05.8
3.4.0, 3.4.1	22.05.7

AWS ParallelCluster versi (s)	SlurmVersi yang didukung
3.3.0, 3.3.1	22.05.5
3.1.4, 3.1.5, 3.2.0, 3.2.1	21.08.8-2
3.1.2, 3.1.3	21.08.6
3.1.1	21.08.5
3.0.0	20.11.8

Topik

- [Konfigurasi beberapa antrian](#)
- [Slurmpanduan untuk beberapa mode antrian](#)
- [Slurmmodus cluster dilindungi](#)
- [Slurmcluster cepat kapasitas tidak mencukupi fail-over](#)
- [Slurmpenjadwalan berbasis memori](#)
- [Beberapa alokasi tipe instans dengan Slurm](#)
- [Penskalaan cluster untuk node dinamis](#)
- [Slurmakuntansi dengan AWS ParallelCluster](#)
- [Slurmkustomisasi konfigurasi](#)
- [Slurm prolog dan epilog](#)
- [Ukuran dan pembaruan kapasitas cluster](#)

Konfigurasi beberapa antrian

Konfigurasi beberapa antrian

Dengan AWS ParallelCluster versi 3, Anda dapat mengonfigurasi beberapa antrian dengan mengatur [Scheduler](#) ke `slurm` dan menentukan lebih dari satu antrian untuk [SlurmQueues](#) dalam file konfigurasi. Dalam mode ini, jenis instance yang berbeda hidup berdampingan dalam node komputasi yang ditentukan di [ComputeResources](#) bagian file konfigurasi. [ComputeResources](#) dengan jenis instance yang berbeda diskalakan ke atas atau ke bawah sesuai kebutuhan untuk [SlurmQueues](#)

Antrian cluster dan kuota sumber daya komputasi

Sumber daya	Kuota
Slurm queues	50 antrian per cluster
Compute resources	50 sumber daya komputasi per antrian 50 sumber daya komputasi per cluster

Hitungan Node

Setiap sumber daya komputasi dalam [ComputeResources](#) antrian harus memiliki unik [Name](#), [InstanceTypeMinCount](#), dan [MaxCount](#) [MinCount](#) dan [MaxCount](#) memiliki nilai default yang menentukan rentang instance untuk sumber daya komputasi [ComputeResources](#) untuk antrian. Anda juga dapat menentukan nilai Anda sendiri untuk [MinCount](#) dan [MaxCount](#). Setiap sumber daya komputasi di [ComputeResources](#) terdiri dari node statis bernomor dari 1 ke nilai [MinCount](#) dan node dinamis diberi nomor dari nilai [MinCount](#) ke nilai [MaxCount](#).

Contoh Konfigurasi

Berikut ini adalah contoh dari bagian [Penjadwalan](#) untuk file konfigurasi cluster. Dalam konfigurasi ini ada dua antrian bernama queue1 dan queue2 dan masing-masing antrian memiliki [ComputeResources](#) dengan yang ditentukan. [MaxCount](#)

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - Name: queue1
    ComputeResources:
    - InstanceType: c5.xlarge
      MaxCount: 5
      Name: c5xlarge
    - InstanceType: c4.xlarge
      MaxCount: 5
      Name: c4xlarge
  - Name: queue2
    ComputeResources:
    - InstanceType: c5.xlarge
      MaxCount: 5
      Name: c5xlarge
```

Nama host

Instance yang diluncurkan ke armada komputasi ditetapkan secara dinamis. Nama host dihasilkan untuk setiap node. Secara default AWS ParallelCluster akan menggunakan format nama host berikut:

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

- `$QUEUE` adalah nama antrian. Misalnya, jika [SlurmQueues](#) bagian memiliki entri dengan [Name](#) set ke "queue-name" maka "`$QUEUE`" adalah "queue-name".
- `$STATDYN` adalah `st` untuk node statis atau `dy` untuk node dinamis.
- `$COMPUTE_RESOURCE` adalah sumber daya [ComputeResources](#) komputasi yang sesuai dengan node ini. [Name](#)
- `$NODENUM` adalah jumlah node. `$NODENUM` adalah antara satu (1) dan nilai [MinCount](#) untuk node statis dan antara satu (1) dan [MaxCount](#) - [MinCount](#) untuk node dinamis.

Dari contoh file konfigurasi di atas, node tertentu dari queue1 dan sumber daya komputasi `c5xlarge` memiliki nama host: `queue1-dy-c5xlarge-1`

Nama host dan nama domain yang sepenuhnya memenuhi syarat (FQDN) dibuat menggunakan zona yang dihosting Amazon Route 53. FQDN adalah `$HOSTNAME.$CLUSTERNAME.pcluster`, di mana `$CLUSTERNAME` nama cluster.

Perhatikan bahwa format yang sama akan digunakan untuk nama node Slurm juga.

Pengguna dapat memilih untuk menggunakan nama host EC2 default dari instance yang memberi daya pada node komputasi alih-alih format nama host default yang digunakan oleh AWS ParallelCluster. Ini dapat dilakukan dengan mengatur [UseEc2Hostnames](#) parameter menjadi benar. Namun, nama node Slurm akan terus menggunakan format default AWS ParallelCluster.

Slurmpanduan untuk beberapa mode antrian

Di sini Anda dapat mempelajari bagaimana AWS ParallelCluster dan Slurm mengelola node antrian (partisi) dan bagaimana Anda dapat memantau status antrian dan node.

Gambaran Umum

Arsitektur penskalaan didasarkan pada Slurm [Panduan Penjadwalan Cloud](#) dan plugin hemat daya. Untuk informasi selengkapnya tentang plugin hemat daya, lihat [Panduan Penghematan Slurm Daya](#). Dalam arsitektur, sumber daya yang berpotensi tersedia untuk kluster biasanya telah ditentukan sebelumnya dalam Slurm konfigurasi sebagai node cloud.

Siklus hidup simpul awan

Sepanjang siklus hidupnya, node cloud memasukkan beberapa jika tidak semua status berikut: `POWER_SAVING`, `POWER_UP` (`pow_up`), `()`, dan `ALLOCATED POWER_DOWN` (`allocpow_dn`). Dalam beberapa kasus, node cloud mungkin memasuki `OFFLINE` status. Daftar berikut merinci beberapa aspek status ini dalam siklus hidup node cloud.

- Sebuah node dalam **POWER_SAVING** keadaan muncul dengan `~` akhiran (misalnya `idle~`) di `sinfo`. Dalam keadaan ini, tidak ada instance EC2 yang mendukung node. Namun, masih Slurm dapat mengalokasikan pekerjaan ke node.
- Sebuah simpul yang beralih ke **POWER_UP** status muncul dengan `#` akhiran (misalnya `idle#`) di `sinfo`. Sebuah node secara otomatis bertransisi ke `POWER_UP` keadaan, ketika Slurm mengalokasikan pekerjaan ke node dalam keadaan `POWER_SAVING`.

Atau, Anda dapat mentransisikan node ke `POWER_UP` status secara manual sebagai pengguna `su root` dengan perintah:

```
$ scontrol update nodename=nodename state=power_up
```

Pada tahap ini, `ResumeProgram` dipanggil, instans EC2 diluncurkan dan dikonfigurasi, dan transisi node ke status `POWER_UP`.

- Node yang saat ini tersedia untuk digunakan muncul tanpa akhiran (misalnya `idle`) di `sinfo`. Setelah node diatur dan telah bergabung dengan cluster, itu menjadi tersedia untuk menjalankan pekerjaan. Pada tahap ini, node dikonfigurasi dengan benar dan siap digunakan.

Sebagai aturan umum, kami merekomendasikan bahwa jumlah instans EC2 sama dengan jumlah node yang tersedia. Dalam kebanyakan kasus, node statis tersedia setelah cluster dibuat.

- Node yang bertransisi ke **POWER_DOWN** status muncul dengan `%` akhiran (misalnya `idle%`) di `sinfo`. Node dinamis secara otomatis memasuki `POWER_DOWN` status setelahnya [ScaledownIdleTime](#). Sebaliknya, node statis dalam banyak kasus tidak dimatikan. Namun, Anda dapat menempatkan node dalam `POWER_DOWN` keadaan secara manual sebagai pengguna `su root` dengan perintah:

```
$ scontrol update nodename=nodename state=down reason="manual draining"
```

Dalam keadaan ini, instance yang terkait dengan node dihentikan, dan node diatur kembali ke `POWER_SAVING` status dan tersedia untuk digunakan setelahnya. [ScaledownIdleTime](#)

[ScaledownIdleTime](#) Pengaturan disimpan ke SuspendTimeout pengaturan Slurm konfigurasi.

- Node yang offline muncul dengan * akhiran (misalnyadown*) di `sinfo`. Sebuah node akan offline jika Slurm controller tidak dapat menghubungi node atau jika node statis dinonaktifkan dan instance backing dihentikan.

Pertimbangkan status node yang ditunjukkan dalam `sinfo` contoh berikut.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4    idle~ efa-dy-efacompute1-[1-4]
efa        up    infinite   1    idle  efa-st-efacompute1-1
gpu        up    infinite   1    idle% gpu-dy-gpucompute1-1
gpu        up    infinite   9    idle~ gpu-dy-gpucompute1-[2-10]
ondemand   up    infinite   2    mix#  ondemand-dy-ondemandcompute1-[1-2]
ondemand   up    infinite  18    idle~ ondemand-dy-ondemandcompute1-
[3-10],ondemand-dy-ondemandcompute2-[1-10]
spot*      up    infinite  13    idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*      up    infinite   2    idle  spot-st-spotcompute2-[1-2]
```

efa-st-efacompute1-1 Node spot-st-spotcompute2-[1-2] dan sudah memiliki instance pendukung yang disiapkan dan tersedia untuk digunakan. ondemand-dy-ondemandcompute1-[1-2] Node berada dalam POWER_UP keadaan dan harus tersedia dalam beberapa menit. gpu-dy-gpucompute1-1 Node dalam POWER_DOWN keadaan, dan transisi ke POWER_SAVING keadaan setelah [ScaledownIdleTime](#) (default ke 10 menit).

Semua node lain dalam POWER_SAVING keadaan tanpa instance EC2 yang mendukungnya.

Bekerja dengan node yang tersedia

Node yang tersedia didukung oleh instance EC2. Secara default, nama node dapat digunakan untuk langsung SSH ke instance (misalnyassh efa-st-efacompute1-1). Alamat IP pribadi dari instance dapat diambil menggunakan perintah:

```
$ scontrol show nodes nodename
```

Periksa alamat IP di NodeAddr bidang yang dikembalikan.

Untuk node yang tidak tersedia, `NodeAdd` bidang tidak boleh mengarah ke instance EC2 yang sedang berjalan. Sebaliknya, itu harus sama dengan nama node.

Status pekerjaan dan pengajuan

Pekerjaan yang diajukan dalam banyak kasus segera dialokasikan ke node dalam sistem, atau ditempatkan di pending jika semua node dialokasikan.

Jika node yang dialokasikan untuk pekerjaan menyertakan node apa pun dalam suatu `POWER_SAVING` keadaan, pekerjaan dimulai dengan `CF`, atau `CONFIGURING` status. Pada saat ini, pekerjaan menunggu node di `POWER_SAVING` negara bagian untuk beralih ke negara `POWER_UP` bagian dan menjadi tersedia.

Setelah semua node yang dialokasikan untuk pekerjaan tersedia, pekerjaan memasuki status `RUNNING (R)`.

Secara default, semua pekerjaan dikirimkan ke antrian default (dikenal sebagai partisi di Slurm). Ini ditandai dengan * akhiran setelah nama antrian. Anda dapat memilih antrian menggunakan opsi pengiriman `-p` pekerjaan.

Semua node dikonfigurasi dengan fitur berikut, yang dapat digunakan dalam perintah pengiriman pekerjaan:

- Tipe instance (misalnya `c5.xlarge`)
- Tipe simpul (Ini adalah salah satu `dynamic` atau `static`.)

Anda dapat melihat fitur untuk node tertentu dengan menggunakan perintah:

```
$ scontrol show nodes nodename
```

Sebagai gantinya, periksa `AvailableFeatures` daftarnya.

Pertimbangkan status awal cluster, yang dapat Anda lihat dengan menjalankan `sinfo` perintah.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4  idle~ efa-dy-efacompute1-[1-4]
efa        up    infinite   1  idle  efa-st-efacompute1-1
gpu        up    infinite  10  idle~ gpu-dy-gpucompute1-[1-10]
ondemand  up    infinite  20  idle~ ondemand-dy-ondemandcompute1-
[1-10],ondemand-dy-ondemandcompute2-[1-10]
```

```
spot*      up    infinite    13  idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*      up    infinite    2   idle spot-st-spotcompute2-[1-2]
```

Perhatikan bahwa spot adalah antrian default. Hal ini ditunjukkan oleh * sufiks.

Kirim pekerjaan ke satu node statis dalam antrian default (spot).

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

Kirim pekerjaan ke satu node dinamis dalam EFA antrian.

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

Kirim pekerjaan ke delapan (8) c5.2xlarge node dan dua (2) t2.xlarge node dalam ondemand antrian.

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

Kirim pekerjaan ke satu node GPU dalam gpu antrian.

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

Pertimbangkan keadaan pekerjaan menggunakan squeue perintah.

```
$ squeue
JOBID PARTITION  NAME  USER  ST      TIME  NODES NODELIST(REASON)
  12  ondemand    wrap  ubuntu CF      0:36    10  ondemand-dy-ondemandcompute1-
[1-8],ondemand-dy-ondemandcompute2-[1-2]
  13      gpu      wrap  ubuntu CF      0:05     1  gpu-dy-gpucompute1-1
   7      spot    wrap  ubuntu R      2:48     1  spot-st-spotcompute2-1
   8      efa     wrap  ubuntu R      0:39     1  efa-dy-efacompute1-1
```

Pekerjaan 7 dan 8 (dalam spot dan efa antrian) sudah berjalan (R). Pekerjaan 12 dan 13 masih mengkonfigurasi (CF), mungkin menunggu instance tersedia.

```
# Nodes states corresponds to state of running jobs
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite    3  idle~ efa-dy-efacompute1-[2-4]
```

```

efa          up    infinite    1    mix  efa-dy-efacompute1-1
efa          up    infinite    1    idle efa-st-efacompute1-1
gpu          up    infinite    1    mix~ gpu-dy-gpucompute1-1
gpu          up    infinite    9    idle~ gpu-dy-gpucompute1-[2-10]
ondemand    up    infinite    10   mix#  ondemand-dy-ondemandcompute1-[1-8],ondemand-
dy-ondemandcompute2-[1-2]
ondemand    up    infinite    10   idle~  ondemand-dy-ondemandcompute1-[9-10],ondemand-
dy-ondemandcompute2-[3-10]
spot*       up    infinite    13   idle~  spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*       up    infinite    1    mix  spot-st-spotcompute2-1
spot*       up    infinite    1    idle spot-st-spotcompute2-2

```

Status dan fitur node

Dalam kebanyakan kasus, status node sepenuhnya dikelola oleh AWS ParallelCluster sesuai dengan proses spesifik dalam siklus hidup node cloud yang dijelaskan sebelumnya dalam topik ini.

Namun, AWS ParallelCluster juga menggantikan atau mengakhiri node yang tidak sehat di DOWN dan DRAINED status dan node yang memiliki instance dukungan yang tidak sehat. Untuk informasi selengkapnya, lihat [clustermgtd](#).

Status partisi

AWS ParallelCluster mendukung status partisi berikut. SlurmPartisi adalah antrian di AWS ParallelCluster.

- UP: Menunjukkan bahwa partisi dalam keadaan aktif. Ini adalah keadaan default partisi. Dalam keadaan ini, semua node di partisi aktif dan tersedia untuk digunakan.
- INACTIVE: Menunjukkan bahwa partisi dalam keadaan tidak aktif. Dalam keadaan ini, semua instance backing node dari partisi tidak aktif dihentikan. Instance baru tidak diluncurkan untuk node di partisi yang tidak aktif.

pcluster update-compute-fleet

- Menghentikan armada komputasi - Ketika perintah berikut dijalankan, semua partisi beralih ke INACTIVE negara bagian, dan AWS ParallelCluster proses menjaga partisi dalam keadaan. INACTIVE

```
$ pcluster update-compute-fleet --cluster-name testSlurm \
```

```
--region eu-west-1 --status STOP_REQUESTED
```

- Memulai armada komputasi - Ketika perintah berikut dijalankan, semua partisi awalnya beralih ke negara bagian UP. Namun, AWS ParallelCluster proses tidak menjaga partisi dalam UP keadaan. Anda perlu mengubah status partisi secara manual. Semua node statis menjadi tersedia setelah beberapa menit. Perhatikan bahwa menyetel partisi ke UP tidak menyalakan kapasitas dinamis apa pun.

```
$ pcluster update-compute-fleet --cluster-name testSlurm \
  --region eu-west-1 --status START_REQUESTED
```

Ketika `update-compute-fleet` dijalankan, Anda dapat memeriksa status cluster dengan menjalankan `pcluster describe-compute-fleet` perintah dan memeriksa `Status`. Berikut daftar kemungkinan status:

- `STOP_REQUESTED`: Permintaan armada stop compute dikirim ke cluster.
- `STOPPING`: `pcluster` Proses saat ini menghentikan armada komputasi.
- `STOPPED`: `pcluster` Proses menyelesaikan proses penghentian, semua partisi dalam `INACTIVE` keadaan, dan semua instance komputasi dihentikan.
- `START_REQUESTED`: Permintaan armada komputasi awal dikirim ke cluster.
- `STARTING`: `pcluster` Proses saat ini memulai cluster.
- `RUNNING`: `pcluster` Proses menyelesaikan proses awal, semua partisi dalam UP keadaan, dan node statis tersedia setelah beberapa menit.
- `PROTECTED`: Status ini menunjukkan bahwa beberapa partisi memiliki kegagalan bootstrap yang konsisten. Partisi yang terpengaruh tidak aktif. Silakan selidiki masalah ini dan kemudian jalankan `update-compute-fleet` untuk mengaktifkan kembali armada.

Kontrol antrian secara manual

Dalam beberapa kasus, Anda mungkin ingin memiliki beberapa kontrol manual atas node atau antrian (dikenal sebagai partisi diSlurm) dalam sebuah cluster. Anda dapat mengelola node dalam cluster melalui prosedur umum berikut menggunakan `scontrol` perintah.

- Nyalakan node dinamis dalam **POWER_SAVING** keadaan

Jalankan perintah sebagai pengguna `su root`:


```
$ scontrol update nodename=nodename state=power_up
```

Anda juga dapat mengirimkan `sleep 1` pekerjaan placeholder yang meminta sejumlah node dan kemudian mengandalkan Slurm untuk meningkatkan jumlah node yang diperlukan.

- Matikan node dinamis sebelumnya [ScaledownIdletime](#)

Kami menyarankan Anda mengatur node dinamis DOWN sebagai pengguna `su root` dengan perintah:

```
$ scontrol update nodename=nodename state=down reason="manually draining"
```

AWS ParallelCluster secara otomatis mengakhiri dan me-reset node dinamis yang jatuh.

Secara umum, kami tidak menyarankan Anda mengatur node untuk `POWER_DOWN` langsung menggunakan `scontrol update nodename=nodename state=power_down` perintah. Ini karena AWS ParallelCluster secara otomatis menangani proses power down.

- Nonaktifkan antrian (partisi) atau hentikan semua node statis di partisi tertentu

Tetapkan antrian tertentu untuk `INACTIVE` sebagai pengguna `su root` dengan perintah:

```
$ scontrol update partition=queuename state=inactive
```

Melakukan hal ini mengakhiri semua instance backing node di partisi.

- Aktifkan antrian (partisi)

Tetapkan antrian tertentu ke `UP` pengguna `su root` dengan perintah:

```
$ scontrol update partition=queuename state=up
```

Perilaku dan penyesuaian penskalaan

Berikut adalah contoh alur kerja penskalaan normal:

- Penjadwal menerima pekerjaan yang membutuhkan dua node.
- Penjadwal mentransisikan dua node ke `POWER_UP` status, dan memanggil `ResumeProgram` dengan nama node (misalnya `queue1-dy-spotcompute1-[1-2]`).

- ResumeProgrammeluncurkan dua instans EC2 dan menetapkan alamat IP pribadi dan nama host dari `queue1-dy-spotcompute1-[1-2]`, menunggu ResumeTimeout (periode default adalah 30 menit sebelum mengatur ulang node).
- Instans dikonfigurasi dan bergabung dengan cluster. Sebuah pekerjaan mulai berjalan pada instance.
- Pekerjaan selesai dan berhenti berjalan.
- Setelah konfigurasi SuspendTime telah berlalu (yang diatur ke [ScaledownIdleTime](#)), penjadwal menyetel instance ke status. POWER_SAVING Scheduler kemudian menetapkan `queue1-dy-spotcompute1-[1-2]` ke POWER_DOWN status dan memanggil SuspendProgram dengan nama node.
- SuspendProgramdisebut untuk dua node. Node tetap dalam POWER_DOWN keadaan, misalnya, dengan tetap idle% selama a SuspendTimeout (periode default adalah 120 detik (2 menit)). Setelah `clustermgtd` mendeteksi bahwa node dimatikan, itu mengakhiri instance dukungan. Kemudian, transisi `queue1-dy-spotcompute1-[1-2]` ke keadaan idle dan me-reset alamat IP pribadi dan nama host sehingga siap untuk power up untuk pekerjaan masa depan.

Jika ada yang salah dan instance untuk node tertentu tidak dapat diluncurkan karena alasan tertentu, maka hal berikut terjadi:

- Penjadwal menerima pekerjaan yang membutuhkan dua node.
- Penjadwal mentransisikan dua node cloud bursting ke POWER_UP status dan memanggil ResumeProgram dengan nama node, (misalnya). `queue1-dy-spotcompute1-[1-2]`
- ResumeProgrammeluncurkan hanya satu (1) instans EC2 dan mengonfigurasi `queue1-dy-spotcompute1-1`, dengan satu (1) instance `queue1-dy-spotcompute1-2`, gagal diluncurkan.
- `queue1-dy-spotcompute1-1` tidak terpengaruh dan online setelah mencapai POWER_UP negara bagian.
- `queue1-dy-spotcompute1-2` transisi ke POWER_DOWN status, dan pekerjaan diminta ulang secara otomatis karena Slurm mendeteksi kegagalan node.
- `queue1-dy-spotcompute1-2` menjadi tersedia setelah SuspendTimeout (defaultnya adalah 120 detik (2 menit)). Sementara itu, pekerjaan tersebut diminta ulang dan dapat mulai berjalan di node lain.
- Proses di atas berulang sampai pekerjaan dapat berjalan pada node yang tersedia tanpa terjadi kegagalan.

Ada dua parameter waktu yang dapat disesuaikan jika diperlukan:

- **ResumeTimeout**(defaultnya adalah 30 menit): ResumeTimeout mengontrol waktu Slurm menunggu sebelum mentransisikan node ke keadaan turun.
 - Mungkin berguna untuk memperpanjang ResumeTimeout jika proses instalasi pra/pasca Anda memakan waktu hampir selama itu.
 - ResumeTimeout juga merupakan waktu maksimum yang AWS ParallelCluster menunggu sebelum mengganti atau mengatur ulang node jika ada masalah. Menghitung node berhenti sendiri jika ada kesalahan yang terjadi selama peluncuran atau penyiapan. AWS ParallelCluster proses menggantikan node pada deteksi instance yang dihentikan.
- **SuspendTimeout**(defaultnya adalah 120 detik (2 menit)): SuspendTimeout mengontrol seberapa cepat node ditempatkan kembali ke sistem dan siap digunakan lagi.
 - Yang lebih pendek SuspendTimeout berarti node diatur ulang lebih cepat, dan Slurm dapat mencoba meluncurkan instance lebih sering.
 - Yang lebih lama SuspendTimeout berarti bahwa node yang gagal diatur ulang lebih lambat. Sementara itu, Slurm mencoba menggunakan node lain. Jika SuspendTimeout lebih dari beberapa menit, cobalah Slurm untuk menelusuri semua node dalam sistem. Yang lebih lama SuspendTimeout mungkin bermanfaat bagi sistem skala besar (lebih dari 1.000 node) untuk mengurangi stres Slurm ketika mencoba untuk sering mengantri ulang pekerjaan yang gagal.
 - Perhatikan bahwa SuspendTimeout tidak mengacu pada waktu AWS ParallelCluster menunggu untuk mengakhiri instance dukungan untuk sebuah node. Instance pendukung untuk POWER_DOWN node segera dihentikan. Proses penghentian biasanya selesai dalam beberapa menit. Namun, selama waktu ini, node tetap dalam POWER_DOWN status dan tidak tersedia untuk penggunaan penjadwal.

Log untuk arsitektur

Daftar berikut berisi log kunci. Nama aliran log yang digunakan dengan Amazon CloudWatch Logs memiliki format `{hostname}.{instance_id}.{logIdentifier}`, di mana *LogIdentifier* mengikuti nama log.

- ResumeProgram: `/var/log/parallelcluster/slurm_resume.log` (slurm_resume)
- SuspendProgram: `/var/log/parallelcluster/slurm_suspend.log` (slurm_suspend)
- clustermgtd: `/var/log/parallelcluster/clustermgtd.log` (clustermgtd)
- computemgtd: `/var/log/parallelcluster/computemgtd.log` (computemgtd)

- `slurmctld`: `/var/log/slurmctld.log` (`slurmctld`)
- `slurmd`: `/var/log/slurmd.log` (`slurmd`)

Masalah umum dan cara men-debug:

Node yang gagal diluncurkan, dinyalakan, atau bergabung dengan cluster

- Node dinamis:
 - Periksa `ResumeProgram` log untuk melihat apakah `ResumeProgram` dipanggil dengan node. Jika tidak, periksa `slurmctld` log untuk menentukan apakah Slurm mencoba memanggil `ResumeProgram` dengan node. Perhatikan bahwa izin yang salah `ResumeProgram` dapat menyebabkannya gagal secara diam-diam.
 - Jika `ResumeProgram` dipanggil, periksa untuk melihat apakah sebuah instance diluncurkan untuk node. Jika instance tidak diluncurkan, harus ada pesan kesalahan yang jelas mengapa instance gagal diluncurkan.
 - Jika sebuah instance diluncurkan, mungkin ada beberapa masalah selama proses bootstrap. Temukan alamat IP pribadi dan ID instance yang sesuai dari `ResumeProgram` log dan lihat log bootstrap yang sesuai untuk instance tertentu di CloudWatch Log.
- Node statis:
 - Periksa `clustermgtd` log untuk melihat apakah instance diluncurkan untuk node. Jika instance tidak diluncurkan, harus ada kesalahan yang jelas tentang mengapa instance gagal diluncurkan.
 - Jika sebuah instance diluncurkan, ada beberapa masalah dengan proses bootstrap. Temukan IP pribadi dan ID instance yang sesuai dari `clustermgtd` log dan lihat log bootstrap yang sesuai untuk instance tertentu di CloudWatch Log.

Node diganti atau dihentikan secara tak terduga, dan kegagalan node

- Node diganti/dihentikan secara tak terduga:
 - Dalam kebanyakan kasus, `clustermgtd` menangani semua tindakan pemeliharaan node. Untuk memeriksa apakah `clustermgtd` diganti atau dihentikan node, periksa `clustermgtd` log.
 - Jika `clustermgtd` diganti atau dihentikan node, harus ada pesan yang menunjukkan alasan tindakan. Jika alasannya terkait penjadwal (misalnya, `simpulnyaDOWN`), periksa `slurmctld` log untuk lebih jelasnya. Jika alasannya terkait EC2, gunakan alat seperti Amazon CloudWatch atau konsol AWS EC2, CLI, atau SDK, untuk memeriksa status atau log untuk instance itu. Misalnya,

Anda dapat memeriksa apakah instans memiliki acara terjadwal atau gagal pemeriksaan status kesehatan EC2.

- Jika `clustermgtd` tidak mengakhiri node, periksa apakah node `computemgtd` dihentikan atau apakah EC2 menghentikan instance untuk merebut kembali Instance Spot.
- Kegagalan simpul:
 - Dalam kebanyakan kasus, pekerjaan secara otomatis diminta ulang jika node gagal. Lihat di `slurmctld` log untuk melihat mengapa pekerjaan atau node gagal dan menilai situasi dari sana.

Kegagalan saat mengganti atau menghentikan instance, kegagalan saat mematikan node

- Secara umum, `clustermgtd` menangani semua tindakan penghentian instance yang diharapkan. Lihat di `clustermgtd` log untuk melihat mengapa gagal mengganti atau mengakhiri node.
- Untuk node dinamis gagal [ScaledownIdleTime](#), lihat di `SuspendProgram` log untuk melihat apakah `slurmctld` proses membuat panggilan dengan node tertentu sebagai argumen. Catatan `SuspendProgram` tidak benar-benar melakukan tindakan tertentu. Sebaliknya, itu hanya log ketika dipanggil. Semua penghentian dan `NodeAddr` penyetelan ulang instans diselesaikan oleh `clustermgtd`. `Slurmtransisi` node ke IDLE setelahnya `SuspendTimeout`.

Masalah lainnya:

- AWS ParallelCluster tidak membuat alokasi pekerjaan atau keputusan penskalaan. Itu hanya mencoba untuk meluncurkan, menghentikan, dan memelihara sumber daya sesuai dengan Slurm instruksi.

Untuk masalah terkait alokasi pekerjaan, alokasi node, dan keputusan penskalaan, lihat `slurmctld` log untuk kesalahan.

Slurmmodus cluster dilindungi

Ketika cluster berjalan dengan modus dilindungi diaktifkan, AWS ParallelCluster monitor dan trek menghitung kegagalan node bootstrap sebagai node komputasi sedang diluncurkan. Hal ini dilakukan untuk mendeteksi apakah kegagalan ini terjadi terus menerus.

Jika berikut ini terdeteksi dalam antrian (partisi), cluster memasuki status dilindungi:

1. Kegagalan bootstrap node komputasi berturut-turut terjadi terus menerus tanpa peluncuran node komputasi yang berhasil.

2. Jumlah kegagalan mencapai ambang batas yang telah ditentukan.

Setelah cluster memasuki status dilindungi, AWS ParallelCluster menonaktifkan antrian dengan kegagalan pada atau di atas ambang batas yang telah ditentukan.

Slurmmodus cluster dilindungi ditambahkan dalam AWS ParallelCluster versi 3.0.0.

Anda dapat menggunakan modus dilindungi untuk mengurangi waktu dan sumber daya yang dihabiskan untuk menghitung node bootstrap kegagalan bersepeda.

Parameter mode terlindungi

protected_failure_count

`protected_failure_count` menentukan jumlah kegagalan berturut-turut dalam antrian (partisi) yang mengaktifkan status cluster dilindungi.

`protected_failure_count` Defaultnya adalah 10 dan mode terlindungi diaktifkan.

Jika `protected_failure_count` lebih besar dari nol, modus dilindungi diaktifkan.

Jika `protected_failure_count` kurang dari atau sama dengan nol, mode dilindungi dinonaktifkan.

Anda dapat mengubah `protected_failure_count` nilai dengan menambahkan parameter dalam file `clustermgtd` konfigurasi yang terletak di `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` dalam file. HeadNode

Anda dapat memperbarui parameter ini kapan saja dan Anda tidak perlu menghentikan armada komputasi untuk melakukannya. Jika peluncuran berhasil dalam antrian sebelum jumlah kegagalan mencapainya `protected_failure_count`, jumlah kegagalan diatur ulang ke nol.

Memeriksa status klaster dalam status terlindungi

Ketika cluster dalam status dilindungi, Anda dapat memeriksa status armada komputasi dan status node.

Hitung status armada

Status armada komputasi ada dalam cluster yang berjalan PROTECTED dalam status terlindungi.

```
$ pcluster describe-compute-fleet --cluster-name <cluster-name> --region <region-id>
{
```

```
"status": "PROTECTED",
"lastStatusUpdateTime": "2022-04-22T00:31:24.000Z"
}
```

Status simpul

Untuk mempelajari antrian (partisi) yang memiliki kegagalan bootstrap yang telah mengaktifkan status terlindungi, masuk ke cluster dan jalankan perintah. `sinfo` Partisi dengan kegagalan bootstrap di atas `protected_failure_count` berada di INACTIVE negara bagian. Partisi tanpa kegagalan bootstrap di atas `protected_failure_count` berada di UP negara bagian dan bekerja seperti yang diharapkan.

PROTECTED status tidak berdampak menjalankan pekerjaan. Jika pekerjaan berjalan pada partisi dengan kegagalan bootstrap di atas `protected_failure_count`, partisi diatur ke INACTIVE setelah pekerjaan yang berjalan selesai.

Pertimbangkan negara simpul yang ditunjukkan dalam contoh berikut.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*  inact infinite 10 down% queue1-dy-c5xlarge-[1-10]
queue1*  inact infinite 3490 idle~ queue1-dy-c5xlarge-[11-3500]
queue2  up  infinite 10 idle~ queue2-dy-c5xlarge-[1-10]
```

Partisi `queue1` adalah INACTIVE karena 10 kegagalan bootstrap node komputasi berturut-turut terdeteksi.

Contoh di belakang node `queue1-dy-c5xlarge-[1-10]` diluncurkan tetapi gagal bergabung dengan kluster karena status tidak sehat.

Kluster berada dalam status terlindungi.

Partisi `queue2` tidak dipengaruhi oleh kegagalan bootstrap di `queue1`. Itu di UP negara bagian dan masih bisa menjalankan pekerjaan.

Cara menonaktifkan status yang dilindungi

Setelah kesalahan bootstrap telah diselesaikan, Anda dapat menjalankan perintah berikut untuk mengambil cluster dari status dilindungi.

```
$ pcluster update-compute-fleet --cluster-name <cluster-name> \
  --region <region-id> \
```

```
--status START_REQUESTED
```

Kegagalan bootstrap yang mengaktifkan status yang dilindungi

Kesalahan bootstrap yang mengaktifkan status dilindungi dibagi lagi menjadi tiga jenis berikut. Untuk mengidentifikasi jenis dan masalah, Anda dapat memeriksa apakah log AWS ParallelCluster yang dihasilkan. Jika log yang dihasilkan, Anda dapat memeriksa mereka untuk rincian kesalahan. Untuk informasi selengkapnya, lihat [Mengambil dan melestarikan log](#).

1. Kesalahan bootstrap yang menyebabkan sebuah instance untuk mengakhiri sendiri.

Sebuah instance gagal di awal proses bootstrap, seperti instance yang berakhir sendiri karena kesalahan dalam skrip [SlurmQueues](#)\ [CustomActions](#)\ [OnNodeStart](#)|\ [OnNodeConfigured](#)

Untuk node dinamis, cari kesalahan yang serupa dengan yang berikut ini:

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

Untuk node statis, lihat di `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) untuk kesalahan yang mirip dengan berikut ini:

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

2. Node **resume_timeout** atau **node_replacement_timeout** kedaluwarsa.

Sebuah instance tidak dapat bergabung dengan cluster dalam `resume_timeout` (untuk node dinamis) atau `node_replacement_timeout` (untuk node statis). Itu tidak mengakhiri diri sendiri sebelum batas waktu. Misalnya, jaringan tidak diatur dengan benar untuk cluster dan node diatur ke DOWN negara Slurm setelah batas waktu berakhir.

Untuk node dinamis, cari kesalahan yang serupa dengan yang berikut ini:

```
Node bootstrap error: Resume timeout expires for node
```

Untuk node statis, lihat di `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) untuk kesalahan yang mirip dengan berikut ini:

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

3. Node gagal pemeriksaan kesehatan.

Sebuah contoh di belakang node gagal pemeriksaan kesehatan EC2 atau dijadwalkan pemeriksaan kesehatan acara, dan node diperlakukan sebagai node kegagalan bootstrap. Dalam hal ini, contoh berakhir karena alasan di luar kendali. AWS ParallelCluster

Lihat di `clustermgtd log (/var/log/parallelcluster/clustermgtd)` untuk kesalahan yang mirip dengan berikut ini:

```
Node bootstrap error: Node %s failed during bootstrap when performing health check.
```

4. Compute node gagal Slurm pendaftaran.

Pendaftaran `slurmd` daemon dengan daemon Slurm kontrol (`slurmctld`) gagal dan menyebabkan negara node komputasi untuk berubah ke negara. `INVALID_REG` Node Slurm komputasi yang dikonfigurasi dengan salah dapat menyebabkan kesalahan ini, seperti node terkompulasi yang dikonfigurasi dengan kesalahan spesifikasi node [CustomSlurmSettings](#)komputasi.

Lihat di file `slurmctld log (/var/log/slurmctld.log)` pada node kepala, atau lihat di file `slurmd log (/var/log/slurmd.log)` dari node komputasi gagal untuk kesalahan yang mirip dengan berikut ini:

```
Setting node %s to INVALID with reason: ...
```

Cara men-debug mode terlindungi

Jika klaster Anda dalam status dilindungi, dan jika `clustermgtd log` AWS ParallelCluster yang dihasilkan dari `HeadNode` dan `cloud-init-output log` dari node komputasi bermasalah, maka Anda dapat memeriksa log untuk rincian kesalahan. Untuk informasi selengkapnya tentang cara mengambil log, lihat [Mengambil dan melestarikan log](#).

`clustermgtdlog (/var/log/parallelcluster/clustermgtd)` pada simpul kepala

Pesan log menunjukkan partisi mana yang memiliki kegagalan bootstrap dan jumlah kegagalan bootstrap yang sesuai.

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - INFO - Partitions bootstrap failure count: {'queue1': 2}, cluster will be set into protected mode if protected failure count reach threshold.
```

Dalam `clustermgtd` log, cari `Found the following bootstrap failure nodes` untuk menemukan node mana yang gagal bootstrap.

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - WARNING -  
Found the following bootstrap failure nodes: (x2) ['queue1-st-  
c5large-1(192.168.110.155)', 'broken-st-c5large-2(192.168.65.215)']
```

Di `clustermgtd` log, cari `Node bootstrap error` untuk menemukan alasan kegagalan.

```
[slurm_plugin.clustermgtd:_is_node_bootstrap_failure] - WARNING - Node bootstrap  
error:  
Node broken-st-c5large-2(192.168.65.215) is currently in replacement and no backing  
instance
```

cloud-init-outputlog (`/var/log/cloud-init-output.log`) pada node komputasi

Setelah mendapatkan alamat IP pribadi node kegagalan bootstrap di `clustermgtd` log, Anda dapat menemukan log node komputasi yang sesuai dengan masuk ke node komputasi atau dengan mengikuti panduan [Mengambil dan melestarikan log](#) untuk mengambil log. Dalam kebanyakan kasus, `/var/log/cloud-init-output` log dari node bermasalah menunjukkan langkah yang menyebabkan kegagalan bootstrap node komputasi.

Slurmcluster cepat kapasitas tidak mencukupi fail-over

Dimulai dengan AWS ParallelCluster versi 3.2.0, cluster berjalan dengan mode fail-over kapasitas yang tidak mencukupi yang diaktifkan secara default. Ini meminimalkan waktu yang dihabiskan untuk mencoba mengantri pekerjaan ketika kesalahan kapasitas EC2 tidak mencukupi terdeteksi. Ini sangat efektif ketika Anda mengkonfigurasi kluster Anda dengan beberapa jenis jenis instans.

EC2 mendeteksi kegagalan kapasitas yang tidak mencukupi:

- `InsufficientInstanceCapacity`
- `InsufficientHostCapacity`
- `InsufficientReservedInstanceCapacity`
- `MaxSpotInstanceCountExceeded`
- `SpotMaxPriceTooLow`: Diaktifkan jika harga permintaan Spot lebih rendah dari harga pemenuhan permintaan Spot minimum yang disyaratkan.
- `Unsupported`: Diaktifkan dengan menggunakan jenis instans yang tidak didukung dalam spesifik Wilayah AWS.

Dalam mode kegagalan kapasitas yang tidak mencukupi, jika kesalahan kapasitas yang tidak mencukupi terdeteksi saat pekerjaan ditugaskan ke a [SlurmQueues/compute_resource](#), AWS ParallelCluster lakukan hal berikut:

1. Ini menetapkan sumber daya komputasi ke negara cacat (DOWN) untuk jangka waktu yang telah ditetapkan.
2. Menggunakan `POWER_DOWN_FORCE` untuk membatalkan sumber daya komputasi gagal pekerjaan node dan untuk menangguhkan node gagal. Ini menetapkan node gagal untuk `IDLE` dan `POWER_DOWN (!)` negara, dan kemudian ke `POWERING_DOWN (%)`.
3. Ini requeue pekerjaan untuk sumber daya komputasi lain.

Node statis dan bertenaga dari sumber daya komputasi yang dinonaktifkan tidak terpengaruh. Pekerjaan dapat menyelesaikan pada node ini.

Siklus ini berulang sampai pekerjaan berhasil ditugaskan ke node sumber daya komputasi atau node. Untuk informasi tentang keadaan simpul, lihat [Slurmpanduan untuk beberapa mode antrian](#).

Jika tidak ada sumber daya komputasi yang ditemukan untuk menjalankan pekerjaan, pekerjaan diatur ke `PENDING` negara sampai periode waktu yang telah ditentukan berlalu. Dalam hal ini, Anda dapat memodifikasi periode waktu yang telah ditentukan seperti yang dijelaskan di bagian berikut.

Parameter batas waktu kapasitas tidak mencukupi

`insufficient_capacity_timeout`

`insufficient_capacity_timeout` menentukan periode waktu (dalam detik) bahwa sumber daya komputasi disimpan dalam keadaan cacat (down) ketika kesalahan kapasitas tidak mencukupi terdeteksi.

Secara default, `insufficient_capacity_timeout` diaktifkan.

Default `insufficient_capacity_timeout` adalah 600 detik (10 menit).

Jika `insufficient_capacity_timeout` nilainya kurang dari atau sama dengan nol, mode kegagalan kapasitas yang tidak mencukupi akan dinonaktifkan.

Anda dapat mengubah `insufficient_capacity_timeout` nilai dengan menambahkan parameter dalam file `clustermgtd` konfigurasi yang terletak di `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` dalam file `HeadNode`.

Parameter dapat diperbarui kapan saja tanpa menghentikan armada komputasi.

Misalnya:

- `insufficient_capacity_timeout=600`:

Jika kesalahan kapasitas tidak mencukupi terdeteksi, sumber daya komputasi diatur ke disabled (DOWN). Setelah 10 menit, node gagal diatur ke `idle~` (POWER_SAVING) negara.

- `insufficient_capacity_timeout=60`:

Jika kesalahan kapasitas tidak mencukupi terdeteksi, sumber daya komputasi dalam cacat (DOWN). Setelah 1 menit, node gagal diatur ke `idle~` negara.

- `insufficient_capacity_timeout=0`:

Mode kegagalan kapasitas yang tidak mencukupi dengan cepat dinonaktifkan. Sumber daya komputasi tidak dinonaktifkan.

Note

Mungkin ada penundaan hingga satu menit antara waktu ketika node gagal dengan kesalahan kapasitas yang tidak mencukupi dan waktu ketika daemon manajemen cluster mendeteksi kegagalan node. Ini karena daemon manajemen kluster memeriksa kegagalan kapasitas node yang tidak mencukupi dan menetapkan sumber daya komputasi ke down status pada interval satu menit.

Status mode fail-over kapasitas yang tidak mencukupi

Ketika cluster dalam mode fail-over kapasitas yang tidak mencukupi, Anda dapat memeriksa status dan status simpul.

Node negara

Ketika pekerjaan diserahkan ke node dinamis sumber daya komputasi dan kesalahan kapasitas yang tidak mencukupi terdeteksi, node ditempatkan di `down#` negara dengan alasan.

```
(Code:InsufficientInstanceCapacity)Failure when resuming nodes.
```

Kemudian dimatikan node (node dalam `idle~` keadaan) diatur `down~` dengan alasan.

```
(Code:InsufficientInstanceCapacity)Temporarily disabling node due to insufficient capacity.
```

Pekerjaan ini requeued ke sumber daya komputasi lainnya dalam antrian.

Node statis sumber daya komputasi dan node yang UP tidak terpengaruh oleh mode fail-over kapasitas yang tidak mencukupi.

Pertimbangkan keadaan simpul yang ditunjukkan dalam contoh berikut.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   30    idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up      infinite   30    idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

Kami mengirimkan pekerjaan ke queue1 yang membutuhkan satu node.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   1     down# queue1-dy-c-1-1
queue1*   up      infinite   15    idle~ queue1-dy-c-2-[1-15]
queue1*   up      infinite   14    down~ queue1-dy-c-1-[2-15]
queue2    up      infinite   30    idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

Nodequeue1-dy-c-1-1 diluncurkan untuk menjalankan pekerjaan. Namun, instance gagal diluncurkan karena kesalahan kapasitas yang tidak mencukupi. Nodequeue1-dy-c-1-1 diatur ktdown. Node dinamis yang dimatikan dalam sumber daya komputasi (queue2-dy-c-1) diatur ktdown.

Anda dapat memeriksa alasan simpul dengan `scontrol show nodes`.

```
$ scontrol show nodes queue1-dy-c-1-1
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Failure when resuming nodes
[root@2022-03-10T22:17:50]

$ scontrol show nodes queue1-dy-c-1-2
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
```

```
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Temporarily disabling node due to
insufficient capacity [root@2022-03-10T22:17:50]
```

Pekerjaan diantrekan ke jenis instance lain dalam sumber daya komputasi antrian.

Setelah `insufficient_capacity_timeout` berlalu, node dalam sumber daya komputasi diatur ulang ke `idle` status.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

Setelah `insufficient_capacity_timeout` berlalu dan node dalam sumber daya komputasi diatur ulang ke `idle` status, Slurm scheduler memberikan node prioritas yang lebih rendah.

Penjadwal terus memilih node dari sumber daya komputasi antrian lainnya dengan bobot yang lebih tinggi kecuali salah satu dari berikut ini terjadi:

- Persyaratan pengiriman pekerjaan sesuai dengan sumber daya komputasi yang dipulihkan.
- Tidak ada sumber daya komputasi lain yang tersedia karena mereka pada kapasitas.
- `slurmctld` dimulai ulang.
- Armada AWS ParallelCluster komputasi dihentikan dan mulai power down dan power up semua node.

log log Terkait

Log yang terkait dengan kesalahan kapasitas yang tidak mencukupi dan mode fail-over kapasitas yang tidak mencukupi dapat ditemukan di `SlurmResumeClustermgtd` log dan masuk ke head node.

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

Pesan kesalahan saat node gagal diluncurkan karena kapasitas tidak mencukupi.

```
[slurm_plugin.instance_manager:_launch_ec2_instances] - ERROR - Failed RunInstances
request: dcd0c252-90d4-44a7-9c79-ef740f7ecd87
[slurm_plugin.instance_manager:add_instances_for_nodes] - ERROR - Encountered
exception when launching instances for nodes (x1) ['queue1-dy-c-1-1']: An error
occurred
```

```
(InsufficientInstanceCapacity) when calling the RunInstances operation (reached max
retries: 1): We currently do not have sufficient p4d.24xlarge capacity in the
Availability Zone you requested (us-west-2b). Our system will be working on
provisioning additional capacity. You can currently get p4d.24xlarge capacity by
not
specifying an Availability Zone in your request or choosing us-west-2a, us-west-2c.
```

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

Menghitung sumber daya c-1 dalam antrian dinonaktifkan karena kapasitas tidak mencukupi.

```
[slurm_plugin.clustermgtd:_reset_timeout_expired_compute_resources] - INFO - The
following compute resources are in down state
due to insufficient capacity: {'queue1': {'c-1':
ComputeResourceFailureEvent(timestamp=datetime.datetime(2022, 4, 14, 23, 0, 4,
769380, tzinfo=datetime.timezone.utc),
error_code='InsufficientInstanceCapacity')}}}, compute resources are reset after
insufficient capacity timeout (600 seconds) expired
```

Setelah batas waktu kapasitas tidak mencukupi berakhir, sumber daya komputasi diatur ulang, node dalam sumber daya komputasi diatur ke `idle`.

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the
following compute resources because insufficient capacity
timeout expired: {'queue1': ['c-1']}
```

Slurmpenjadwalan berbasis memori

Dimulai dengan versi 3.2.0, AWS ParallelCluster mendukung penjadwalan Slurm berbasis memori dengan parameter [SlurmSettingsEnableMemoryBasedScheduling](#) konfigurasi/cluster.

Note

[Dimulai dengan AWS ParallelCluster versi 3.7.0, EnableMemoryBasedScheduling dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instans di Instans.](#)

Untuk AWS ParallelCluster versi 3.2.0 hingga 3.6. x, tidak `EnableMemoryBasedScheduling` dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instance di [Instance](#).

⚠ Warning

Bila Anda menentukan beberapa jenis instance dalam sumber daya komputasi Slurm antrian dengan `EnableMemoryBasedScheduling` diaktifkan, `RealMemory` nilainya adalah jumlah minimum memori yang tersedia untuk semua jenis instans. Ini dapat menyebabkan sejumlah besar memori yang tidak digunakan jika Anda menentukan jenis instance dengan kapasitas memori yang sangat berbeda.

Dengan `EnableMemoryBasedScheduling: true`, Slurm penjadwal melacak jumlah memori yang dibutuhkan setiap pekerjaan pada setiap node. Kemudian, Slurm penjadwal menggunakan informasi ini untuk menjadwalkan beberapa pekerjaan pada node komputasi yang sama. Jumlah total memori yang dibutuhkan pekerjaan pada node tidak bisa lebih besar dari memori node yang tersedia. Penjadwal mencegah pekerjaan menggunakan lebih banyak memori daripada yang diminta saat pekerjaan dikirimkan.

Dengan `EnableMemoryBasedScheduling: false`, pekerjaan mungkin bersaing untuk mendapatkan memori pada node bersama dan menyebabkan kegagalan pekerjaan dan `out-of-memory` peristiwa.

⚠ Warning

Slurmmenggunakan kekuatan 2 notasi untuk labelnya, seperti MB atau GB. Baca label ini sebagai MiB dan GiB, masing-masing.

Slurm konfigurasi dan penjadwalan berbasis memori

Dengan `EnableMemoryBasedScheduling: true`, Slurm atur parameter Slurm konfigurasi berikut:

- [`SelectTypeParameters=CR_CPU_Memory`](#) di `slurm.conf`. Opsi ini mengonfigurasi memori node menjadi sumber daya habis pakai di Slurm
- [`ConstrainRAMSpace=yes`](#) di `Slurmgroup.conf`. Dengan opsi ini, akses pekerjaan ke memori terbatas pada jumlah memori yang diminta pekerjaan saat dikirimkan.

Note

Beberapa parameter Slurm konfigurasi lainnya dapat memengaruhi perilaku Slurm penjadwal dan pengelola sumber daya saat kedua opsi ini disetel. Untuk informasi selengkapnya, lihat Dokumentasi [Slurm](#).

Slurmpenjadwal dan penjadwalan berbasis memori

EnableMemoryBasedScheduling: false(default)

Secara default, `EnableMemoryBasedScheduling` diatur ke `false`. Ketika `false`, Slurm tidak termasuk memori sebagai sumber daya dalam algoritma penjadwalan dan tidak melacak memori yang digunakan pekerjaan. Pengguna dapat menentukan `--mem MEM_PER_NODE` opsi untuk mengatur jumlah minimum memori per node yang dibutuhkan pekerjaan. Ini memaksa penjadwal untuk memilih node dengan `RealMemory` nilai setidaknya `MEM_PER_NODE` saat menjadwalkan pekerjaan.

Misalnya, anggaplah bahwa pengguna mengirimkan dua pekerjaan dengan `--mem=5GB`. Jika sumber daya yang diminta seperti CPU atau GPU tersedia, pekerjaan dapat berjalan pada saat yang sama pada node dengan memori 8 GiB. Kedua pekerjaan tidak dijadwalkan pada node komputasi dengan kurang dari 5 `RealMemory` GiB.

Warning

Saat penjadwalan berbasis memori dinonaktifkan, Slurm tidak melacak jumlah memori yang digunakan pekerjaan. Pekerjaan yang berjalan pada node yang sama mungkin bersaing untuk sumber daya memori dan menyebabkan pekerjaan lain gagal.

Ketika penjadwalan berbasis memori dinonaktifkan, kami menyarankan agar pengguna tidak menentukan atau opsi. [--mem-per-cpu--mem-per-gpu](#) Opsi ini dapat menyebabkan perilaku yang berbeda dari apa yang dijelaskan dalam Dokumentasi [Slurm](#).

EnableMemoryBasedScheduling: true

Ketika `EnableMemoryBasedScheduling` disetel ke `true`, Slurm melacak penggunaan memori setiap pekerjaan dan mencegah pekerjaan menggunakan lebih banyak memori daripada yang diminta dengan opsi `--mem` pengiriman.

Menggunakan contoh sebelumnya, pengguna mengirimkan dua pekerjaan dengan `--mem=5GB`. Pekerjaan tidak dapat berjalan pada saat yang sama pada node dengan memori 8 GiB. Ini karena jumlah total memori yang dibutuhkan lebih besar daripada memori yang tersedia di node.

Dengan penjadwalan berbasis memori diaktifkan, `--mem-per-cpu` dan `--mem-per-gpu` berperilaku konsisten dengan apa yang dijelaskan dalam dokumentasi. Slurm Misalnya, pekerjaan diserahkan `--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB`. Dalam hal ini, Slurm berikan pekerjaan total 4 GiB untuk setiap node.

Warning

Ketika penjadwalan berbasis memori diaktifkan, kami menyarankan agar pengguna menyertakan `--mem` spesifikasi saat mengirimkan pekerjaan. Dengan Slurm konfigurasi default yang disertakan AWS ParallelCluster, jika tidak ada opsi memori yang disertakan (`--mem`, `--mem-per-cpu`, atau `--mem-per-gpu`), Slurm menetapkan seluruh memori node yang dialokasikan ke pekerjaan, bahkan jika itu hanya meminta sebagian dari sumber daya lain, seperti CPU atau GPU. Ini secara efektif mencegah berbagi node sampai pekerjaan selesai karena tidak ada memori yang tersedia untuk pekerjaan lain. Ini terjadi karena Slurm menyetel memori per node untuk pekerjaan `DefMemPerNode` ketika tidak ada spesifikasi memori yang disediakan pada waktu pengiriman pekerjaan. Nilai default untuk parameter ini adalah 0 dan menentukan akses tak terbatas ke memori node.

Jika beberapa jenis sumber daya komputasi dengan jumlah memori yang berbeda tersedia dalam antrian yang sama, pekerjaan yang dikirimkan tanpa opsi memori mungkin diberikan jumlah memori yang berbeda pada node yang berbeda. Ini tergantung pada node mana yang disediakan oleh penjadwal untuk pekerjaan itu. Pengguna dapat menentukan nilai kustom untuk opsi, seperti `DefMemPerNode` atau `DefMemPerCPU`, pada tingkat cluster atau partisi dalam file Slurm konfigurasi untuk mencegah perilaku ini.

Slurm `RealMemory` dan AWS ParallelCluster `SchedulableMemory`

Dengan Slurm konfigurasi yang disertakan AWS ParallelCluster, Slurm menafsirkan `RealMemory` sebagai jumlah memori per node yang tersedia untuk pekerjaan. Dimulai dengan versi 3.2.0, secara default, AWS ParallelCluster disetel `RealMemory` ke 95 persen memori yang tercantum dalam Jenis [Instans Amazon EC2](#) dan dikembalikan oleh Amazon EC2 API. [DescribeInstanceTypes](#)

Saat penjadwalan berbasis memori dinonaktifkan, Slurm penjadwal menggunakan `RealMemory` untuk memfilter node saat pengguna mengirimkan pekerjaan dengan yang ditentukan. `--mem`

Ketika penjadwalan berbasis memori diaktifkan, Slurm scheduler menafsirkan `RealMemory` sebagai jumlah maksimum memori yang tersedia untuk pekerjaan yang berjalan pada node komputasi.

Pengaturan default mungkin tidak optimal untuk semua jenis instans:

- Pengaturan ini mungkin lebih tinggi dari jumlah memori yang sebenarnya dapat diakses oleh node. Hal ini dapat terjadi ketika node komputasi adalah tipe instance kecil.
- Pengaturan ini mungkin lebih rendah dari jumlah memori yang sebenarnya dapat diakses oleh node. Hal ini dapat terjadi ketika node komputasi adalah tipe instance besar dan dapat menyebabkan sejumlah besar memori yang tidak digunakan.

Anda dapat menggunakan [SlurmQueues/ComputeResources/SchedulableMemory](#) untuk menyempurnakan nilai yang `RealMemory` dikonfigurasi oleh AWS ParallelCluster untuk node komputasi. Untuk mengganti default, tentukan nilai kustom `SchedulableMemory` khusus untuk konfigurasi kluster Anda.

Untuk memeriksa memori aktual node komputasi yang tersedia, jalankan `/opt/slurm/sbin/slurmd -C` perintah pada node. Perintah ini mengembalikan konfigurasi perangkat keras node, termasuk [RealMemory](#) nilainya. Untuk informasi selengkapnya, lihat [slurmd -C](#).

Pastikan bahwa proses sistem operasi node komputasi memiliki memori yang cukup. Untuk melakukan ini, batasi memori yang tersedia untuk pekerjaan dengan menetapkan `SchedulableMemory` nilai lebih rendah dari `RealMemory` nilai yang dikembalikan `slurmd -C` perintah.

Beberapa alokasi tipe instans dengan Slurm

Dimulai dengan AWS ParallelCluster versi 3.3.0, Anda dapat mengonfigurasi kluster Anda untuk mengalokasikan dari kumpulan sumber daya komputasi dari jenis instans yang ditentukan. Alokasi dapat didasarkan pada strategi biaya rendah armada EC2 atau kapasitas optimal.

Kumpulan jenis instance yang ditentukan ini harus memiliki jumlah vCPU yang sama atau, jika multithreading dinonaktifkan, jumlah inti yang sama. Selain itu, rangkaian jenis instance ini harus memiliki jumlah akselerator yang sama dari produsen yang sama. Jika `Efa/Enabled` diatur ke `true`, instance harus didukung EFA. Untuk informasi dan persyaratan selengkapnya, lihat [Scheduling/SlurmQueues/AllocationStrategy](#) dan [ComputeResources/Instances](#).

Anda dapat mengatur [AllocationStrategy](#) kepada `lowest-price` atau `capacity-optimized` tergantung pada Anda [CapacityType](#) konfigurasi.

Di [Instances](#), Anda dapat mengkonfigurasi satu set jenis instance.

Note

Dimulai dengan AWS ParallelCluster versi 3.7.0, `EnableMemoryBasedScheduling` dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instans di [Contoh](#).

Untuk AWS ParallelCluster versi 3.2.0 hingga 3.6.x, `EnableMemoryBasedScheduling` tidak dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instans di [Contoh](#).

Contoh berikut menunjukkan bagaimana Anda dapat melakukan kueri jenis instans untuk vCPU, dukungan EFA, dan arsitektur.

Permintaan `InstanceTypes` dengan 96 vCPU dan arsitektur `x86_64`.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-vcpus,Values=96" "Name=processor-info.supported-
  architecture,Values=x86_64" \
  --query "sort_by(InstanceTypes[*].
  {InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
  &InstanceType})" \
  --output table
```

Permintaan `InstanceTypes` dengan 64 core, dukungan EFA, dan arsitektur `arm64`.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-cores,Values=64" "Name=processor-
  info.supported-architecture,Values=arm64" "Name=network-info.efa-
  supported,Values=true" --query "sort_by(InstanceTypes[*].
  {InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
  &InstanceType})" \
  --output table
```

Contoh cuplikan konfigurasi cluster berikutnya menunjukkan bagaimana Anda dapat menggunakan `InstanceType` dan `AllocationStrategy` properti.

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue-1
```

```
CapacityType: ONDEMAND
AllocationStrategy: lowest-price
...
ComputeResources:
  - Name: computeresource1
    Instances:
      - InstanceType: r6g.2xlarge
      - InstanceType: m6g.2xlarge
      - InstanceType: c6g.2xlarge
    MinCount: 0
    MaxCount: 500
  - Name: computeresource2
    Instances:
      - InstanceType: m6g.12xlarge
      - InstanceType: x2gd.12xlarge
    MinCount: 0
    MaxCount: 500
...
```

Penskalaan cluster untuk node dinamis

ParallelCluster mendukung Slurm metode untuk menskalakan cluster secara dinamis dengan menggunakan plugin Slurm penghemat daya. Untuk informasi selengkapnya, lihat [Panduan Penjadwalan Cloud dan Panduan Hemat Slurm Daya](#) di Slurm dokumentasi.

Dimulai dengan ParallelCluster versi 3.8.0, ParallelCluster menggunakan resume tingkat Pekerjaan atau penskalaan tingkat pekerjaan sebagai strategi alokasi node dinamis default untuk menskalakan cluster: ParallelCluster meningkatkan skala cluster berdasarkan persyaratan setiap pekerjaan, jumlah node yang dialokasikan untuk pekerjaan, dan node mana yang perlu dilanjutkan. ParallelCluster mendapatkan informasi ini dari variabel lingkungan SLURM_RESUME_FILE.

Penskalaan untuk node dinamis adalah proses dua langkah, yang melibatkan peluncuran instans EC2 dan penetapan instans EC2 yang diluncurkan ke node Slurm. Masing-masing dari dua langkah ini dapat dilakukan dengan menggunakan logika upaya terbaik all-or-nothing atau terbaik.

Untuk peluncuran instans EC2:

- all-or-nothing memanggil peluncuran EC2 API dengan target minimum sama dengan total kapasitas target
- upaya terbaik memanggil peluncuran EC2 API dengan target minimum sama dengan 1 dan total kapasitas target sama dengan kapasitas yang diminta

Untuk penugasan instance EC2 ke node Slurm:

- all-or-nothingmenetapkan instance EC2 ke node Slurm hanya jika memungkinkan untuk menetapkan instance EC2 ke setiap node yang diminta
- upaya terbaik menetapkan instans EC2 ke node Slurm bahkan jika semua node yang diminta tidak tercakup oleh kapasitas instans EC2

Kombinasi yang mungkin dari strategi di atas diterjemahkan ke dalam strategi ParallelCluster peluncuran.

Example

<caption>The available ParallelCluster strategi peluncuran that can be set into the [ScalingStrategy](#) cluster configuration to be used with penskalaan tingkat pekerjaan are:</caption>

all-or-nothingpenskalaan:

Strategi ini melibatkan AWS ParallelCluster memulai panggilan API instans peluncuran Amazon EC2 untuk setiap pekerjaan, yang mengharuskan semua instance yang diperlukan agar node komputasi yang diminta berhasil diluncurkan. Ini memastikan bahwa klaster hanya menskalakan ketika kapasitas yang diperlukan per pekerjaan tersedia, menghindari instance idle yang tersisa di akhir proses penskalaan.

Strategi ini menggunakan all-or-nothinglogika untuk peluncuran instans EC2 untuk setiap pekerjaan plus dan all-or-nothinglogika untuk penugasan instans EC2 ke node Slurm.

Grup strategi meluncurkan permintaan ke dalam batch, satu untuk setiap sumber daya komputasi yang diminta dan masing-masing hingga 500 node. Untuk permintaan yang mencakup beberapa sumber daya komputasi atau melebihi 500 node, ParallelCluster secara berurutan memproses beberapa batch.

Kegagalan kumpulan sumber daya tunggal mana pun mengakibatkan penghentian semua kapasitas yang tidak digunakan terkait, memastikan bahwa tidak ada instance idle yang tersisa di akhir proses penskalaan.

Batasan

- Waktu yang dibutuhkan untuk penskalaan berbanding lurus dengan jumlah pekerjaan yang diajukan per pelaksanaan program resume Slurm.

- Operasi penskalaan dibatasi oleh batas akun RunInstances sumber daya, ditetapkan pada 1000 instance secara default. Batasan ini sesuai dengan AWS kebijakan pembatasan API EC2, untuk detail selengkapnya lihat dokumentasi pelambatan API [AWS EC2](#)
- Saat Anda mengirimkan pekerjaan dalam sumber daya komputasi dengan satu jenis instans, dalam antrian yang mencakup beberapa Availability Zone, panggilan API peluncuran all-or-nothingEC2 hanya berhasil jika semua kapasitas dapat disediakan dalam satu Availability Zone.
- Saat Anda mengirimkan pekerjaan di sumber daya komputasi dengan beberapa jenis instans, dalam antrian dengan satu Availability Zone, panggilan API peluncuran all-or-nothingEC2 hanya berhasil jika semua kapasitas dapat disediakan oleh satu jenis instans.
- Saat Anda mengirimkan pekerjaan di sumber daya komputasi dengan beberapa jenis instans, dalam antrian yang mencakup beberapa Availability Zone, panggilan API peluncuran all-or-nothingEC2 tidak didukung dan ParallelCluster melakukan penskalaan upaya terbaik sebagai gantinya.

greedy-all-or-nothingpenskalaan:

Varian all-or-nothing strategi ini masih memastikan bahwa kluster hanya menskalakan ketika kapasitas yang diperlukan per pekerjaan tersedia, menghindari instans idle di akhir proses penskalaan, tetapi ini melibatkan ParallelCluster memulai panggilan API instans peluncuran Amazon EC2 yang bertujuan untuk kapasitas target minimum 1, mencoba memaksimalkan jumlah node yang diluncurkan hingga kapasitas yang diminta. Strategi ini menggunakan logika upaya terbaik untuk peluncuran instans EC2 untuk semua pekerjaan ditambah all-or-nothinglogika untuk penugasan instans EC2 ke node Slurm untuk setiap pekerjaan.

Grup strategi meluncurkan permintaan ke dalam batch, satu untuk setiap sumber daya komputasi yang diminta dan masing-masing hingga 500 node. Untuk permintaan yang mencakup beberapa sumber daya komputasi atau melebihi 500 node, Parallelcluster secara berurutan memproses beberapa batch.

Ini memastikan bahwa tidak ada instance idle yang tersisa di akhir proses penskalaan, dengan memaksimalkan throughput dengan biaya penskalaan berlebih sementara selama proses penskalaan.

Batasan

- Penskalaan berlebih sementara dimungkinkan, yang menyebabkan biaya tambahan untuk instance yang beralih ke status berjalan sebelum penyelesaian penskalaan.

- Batas instans yang sama seperti dalam all-or-nothing strategi berlaku, tunduk AWS pada batas akun RunInstances sumber daya.

penskalaan upaya terbaik:

Strategi ini memanggil panggilan API instans peluncuran EC2 dengan menargetkan kapasitas minimum 1 dan bertujuan untuk mencapai total kapasitas yang diminta dengan biaya meninggalkan instans idle setelah eksekusi proses penskalaan jika tidak semua kapasitas yang diminta tersedia. Strategi ini menggunakan logika upaya terbaik untuk peluncuran instans EC2 untuk semua pekerjaan ditambah logika upaya terbaik untuk penugasan instans Amazon EC2 ke node Slurm untuk setiap pekerjaan.

Grup strategi meluncurkan permintaan ke dalam batch, satu untuk setiap sumber daya komputasi yang diminta dan masing-masing hingga 500 node. Untuk permintaan yang mencakup beberapa sumber daya komputasi atau melebihi 500 node, ParallelCluster secara berurutan memproses beberapa batch.

Strategi ini memungkinkan penskalaan jauh melampaui batas 1000 instans default atas beberapa eksekusi proses penskalaan, dengan biaya memiliki instans idle di seluruh proses penskalaan yang berbeda.

Batasan

- Kemungkinan instance idle running di akhir proses penskalaan, untuk kasus ketika tidak mungkin mengalokasikan semua node yang diminta oleh pekerjaan.

Berikut ini adalah contoh yang menunjukkan bagaimana penskalaan node dinamis berperilaku menggunakan strategi ParallelCluster peluncuran yang berbeda. Misalkan Anda telah mengirimkan dua pekerjaan yang meminta masing-masing 20 node, dengan total 40 node dari jenis yang sama, tetapi hanya ada 30 instans EC2 yang tersedia untuk menutupi kapasitas yang diminta pada EC2.

all-or-nothing penskalaan:

- Untuk pekerjaan pertama, API instans peluncuran all-or-nothing EC2 dipanggil, meminta 20 instans. Panggilan yang berhasil menghasilkan peluncuran 20 instance
- all-or-nothing penugasan 20 instance yang diluncurkan ke node Slurm untuk pekerjaan pertama berhasil

- API instance peluncuran all-or-nothing EC2 lainnya dipanggil, meminta 20 instance untuk pekerjaan kedua. Panggilan tidak berhasil, karena hanya ada kapasitas untuk 10 contoh lainnya. Tidak ada instance yang diluncurkan saat ini

greedy-all-or-nothing penskalaan:

- API instans peluncuran EC2 upaya terbaik dipanggil, meminta 40 instans, yang merupakan total kapasitas yang diminta oleh semua pekerjaan. Ini menghasilkan peluncuran 30 instance
- all-or-nothing Penugasan 20 instance yang diluncurkan ke node Slurm untuk pekerjaan pertama berhasil
- all-or-nothing Penugasan lain dari instance yang diluncurkan yang tersisa ke node Slurm untuk pekerjaan kedua dicoba, tetapi karena hanya ada 10 instance yang tersedia dari total 20 yang diminta oleh pekerjaan, penugasan tidak berhasil
- 10 instans yang diluncurkan yang tidak ditetapkan dihentikan

penskalaan upaya terbaik:

- API instans peluncuran EC2 upaya terbaik dipanggil, meminta 40 instans, yang merupakan total kapasitas yang diminta oleh semua pekerjaan. Ini menghasilkan peluncuran 30 instance.
- Penugasan upaya terbaik dari 20 instance yang diluncurkan ke node Slurm untuk pekerjaan pertama berhasil.
- Penugasan upaya terbaik lainnya dari 10 instance yang diluncurkan yang tersisa ke node Slurm untuk pekerjaan kedua berhasil, bahkan jika total kapasitas yang diminta adalah 20. Tetapi karena pekerjaan itu meminta 20 node, dan dimungkinkan untuk menetapkan instans EC2 hanya 10 dari mereka, pekerjaan tidak dapat dimulai dan instance dibiarkan berjalan menganggur, sampai kapasitas yang cukup ditemukan untuk memulai 10 instance yang hilang pada panggilan selanjutnya dari proses penskalaan, atau penjadwal menjadwalkan pekerjaan di node komputasi lain yang sudah berjalan.

Slurm strategi alokasi node dinamis dalam versi 3.7.x

ParallelCluster menggunakan 2 jenis strategi alokasi node dinamis untuk menskalakan cluster:

- Alokasi berdasarkan informasi node yang diminta yang tersedia:
 - Semua node melanjutkan atau penskalaan daftar simpul:

ParallelCluster meningkatkan skala cluster hanya berdasarkan Slurm nama daftar node yang diminta saat Slurm `ResumeProgram` dijalankan. Ini mengalokasikan sumber daya komputasi ke node hanya dengan nama node. Daftar nama node dapat mencakup beberapa pekerjaan.

- Resume tingkat pekerjaan atau penskalaan tingkat pekerjaan:

ParallelCluster skala cluster berdasarkan persyaratan setiap pekerjaan, jumlah node saat ini yang dialokasikan untuk pekerjaan, dan node mana yang perlu dilanjutkan. ParallelCluster mendapatkan informasi ini dari variabel `SLURM_RESUME_FILE` lingkungan.

- Alokasi dengan strategi peluncuran EC2:

- Penskalaan upaya terbaik:

ParallelCluster meningkatkan skala cluster dengan menggunakan panggilan API instans peluncuran EC2 dengan kapasitas target minimum sama dengan 1, untuk meluncurkan beberapa, tetapi tidak harus semua instance diperlukan untuk mendukung node yang diminta.

- Il-or-nothingPenskalaan:

ParallelCluster meningkatkan skala cluster dengan menggunakan panggilan API instance peluncuran EC2 yang hanya berhasil jika semua instance yang diperlukan untuk mendukung node yang diminta diluncurkan. Dalam hal ini, ia memanggil API instans peluncuran EC2 dengan kapasitas target minimum sama dengan total kapasitas yang diminta.

Secara default, ParallelCluster menggunakan penskalaan daftar simpul dengan strategi peluncuran EC2 upaya terbaik untuk meluncurkan beberapa, tetapi tidak harus semua instance diperlukan untuk mendukung node yang diminta. Ia mencoba menyediakan kapasitas sebanyak mungkin untuk melayani beban kerja yang diajukan.

Dimulai dengan ParallelCluster versi 3.7.0, ParallelCluster menggunakan penskalaan tingkat pekerjaan dengan strategi peluncuran `all-or-nothingEC2` untuk pekerjaan yang dikirimkan dalam mode eksklusif. Saat Anda mengirimkan pekerjaan dalam mode eksklusif, pekerjaan tersebut memiliki akses eksklusif ke node yang dialokasikan. Untuk informasi lebih lanjut, lihat [EKSKLUSIF](#) dalam Slurm dokumentasi.

Untuk mengirimkan pekerjaan dalam mode eksklusif:

- Lewati bendera eksklusif saat mengirimkan Slurm pekerjaan ke cluster. Misalnya, `sbatch ... --exclusive`.

ATAU

- Kirim pekerjaan ke antrian cluster yang telah dikonfigurasi dengan [JobExclusiveAllocation](#) set ke `true`.

Saat mengirimkan pekerjaan dalam mode eksklusif:

- ParallelCluster saat ini batch meluncurkan permintaan untuk menyertakan hingga 500 node. Jika pekerjaan meminta lebih dari 500 node, ParallelCluster membuat permintaan all-or-nothing peluncuran untuk setiap set 500 node dan permintaan peluncuran tambahan untuk sisa node.
- Jika alokasi node dalam sumber daya komputasi tunggal, ParallelCluster buat permintaan all-or-nothing peluncuran untuk setiap set 500 node dan permintaan peluncuran tambahan untuk sisa node. Jika permintaan peluncuran gagal, ParallelCluster menghentikan kapasitas yang tidak terpakai yang dibuat oleh semua permintaan peluncuran.
- Jika alokasi node mencakup beberapa sumber daya komputasi, ParallelCluster perlu membuat permintaan all-or-nothing peluncuran untuk setiap sumber daya komputasi. Permintaan ini juga dikelompokkan. Jika permintaan peluncuran gagal untuk salah satu sumber daya komputasi, ParallelCluster menghentikan kapasitas yang tidak terpakai yang dibuat oleh semua permintaan peluncuran sumber daya komputasi.

penskalaan tingkat pekerjaan dengan strategi all-or-nothing peluncuran batasan yang diketahui:

- Saat Anda mengirimkan pekerjaan dalam sumber daya komputasi dengan satu jenis instans, dalam antrian yang mencakup beberapa Availability Zone, panggilan API peluncuran all-or-nothing EC2 hanya berhasil jika semua kapasitas dapat disediakan dalam satu Availability Zone.
- Saat Anda mengirimkan pekerjaan di sumber daya komputasi dengan beberapa jenis instans, dalam antrian dengan satu Availability Zone, panggilan API peluncuran all-or-nothing EC2 hanya berhasil jika semua kapasitas dapat disediakan oleh satu jenis instans.
- Saat Anda mengirimkan pekerjaan di sumber daya komputasi dengan beberapa jenis instans, dalam antrian yang mencakup beberapa Availability Zone, panggilan API peluncuran all-or-nothing EC2 tidak didukung dan ParallelCluster melakukan penskalaan upaya terbaik sebagai gantinya.

Slurmstrategi alokasi node dinamis di versi 3.6.x dan sebelumnya

AWS ParallelCluster hanya menggunakan satu jenis strategi alokasi node dinamis untuk menskalakan cluster:

- Alokasi berdasarkan informasi node yang diminta yang tersedia:
 - Semua node melanjutkan atau penskalaan daftar simpul: meningkatkan ParallelCluster skala cluster hanya berdasarkan nama daftar node yang diminta Slurm saat dijalankan. Slurm `ResumeProgram` ini mengalokasikan sumber daya komputasi ke node hanya dengan nama node. Daftar nama node dapat mencakup beberapa pekerjaan.
- Alokasi dengan strategi peluncuran EC2:
 - Penskalaan upaya terbaik: meningkatkan ParallelCluster skala cluster dengan menggunakan panggilan API instans peluncuran EC2 dengan kapasitas target minimum sama dengan 1, untuk meluncurkan beberapa, tetapi tidak harus semua instance diperlukan untuk mendukung node yang diminta.

ParallelCluster menggunakan penskalaan daftar simpul dengan strategi peluncuran EC2 upaya terbaik untuk meluncurkan beberapa, tetapi tidak harus semua instance diperlukan untuk mendukung node yang diminta. Ia mencoba menyediakan kapasitas sebanyak mungkin untuk melayani beban kerja yang diajukan.

Batasan

- Kemungkinan instance idle running di akhir proses penskalaan, untuk kasus ketika tidak mungkin mengalokasikan semua node yang diminta oleh pekerjaan.


Slurmakuntansi dengan AWS ParallelCluster

Dimulai dengan versi 3.3.0, AWS ParallelCluster mendukung Slurm akuntansi dengan parameter konfigurasi kluster [SlurmSettings/Database](#).

Dengan Slurm akuntansi, Anda dapat mengintegrasikan database akuntansi eksternal untuk melakukan hal berikut:

- Mengelola pengguna kluster atau grup pengguna dan entitas lainnya. Dengan kemampuan ini, Anda dapat menggunakan Slurm fitur-fitur yang lebih canggih, seperti penegakan batas sumber daya, fairshare, dan QOS.

- Kumpulkan dan simpan data pekerjaan, seperti pengguna yang menjalankan pekerjaan, durasi pekerjaan, dan sumber daya yang digunakannya. Anda dapat melihat data yang disimpan dengan `sacct` utilitas.

 Note

AWS ParallelCluster mendukung Slurm akuntansi untuk server [database MySQL yang Slurm didukung](#).


Bekerja dengan Slurm akuntansi di AWS ParallelCluster

Sebelum Anda mengkonfigurasi Slurm akuntansi, Anda harus memiliki server database eksternal yang ada dan database yang menggunakan `mysql` protokol.

Untuk mengonfigurasi Slurm akuntansi dengan AWS ParallelCluster, Anda harus menentukan yang berikut:

- URI untuk server database eksternal di [Database/Uri](#). Server harus ada dan dapat dijangkau dari node kepala.
- Kredensial untuk mengakses database eksternal yang didefinisikan dalam [Database/PasswordSecretArn](#) dan [Database/UserName](#). AWS ParallelCluster menggunakan informasi ini untuk mengkonfigurasi akuntansi di Slurm tingkat dan `slurmdbd` layanan pada node kepala. `slurmdbd` adalah daemon yang mengelola komunikasi antara cluster dan server database.

Untuk melangkah melalui tutorial, lihat [Membuat cluster dengan akuntansi Slurm](#).

 Note

AWS ParallelCluster melakukan bootstrap dasar dari database Slurm akuntansi dengan mengatur pengguna cluster default sebagai admin database dalam Slurm database. AWS ParallelCluster tidak menambahkan pengguna lain ke database akuntansi. Pelanggan bertanggung jawab untuk mengelola entitas akuntansi dalam Slurm database.

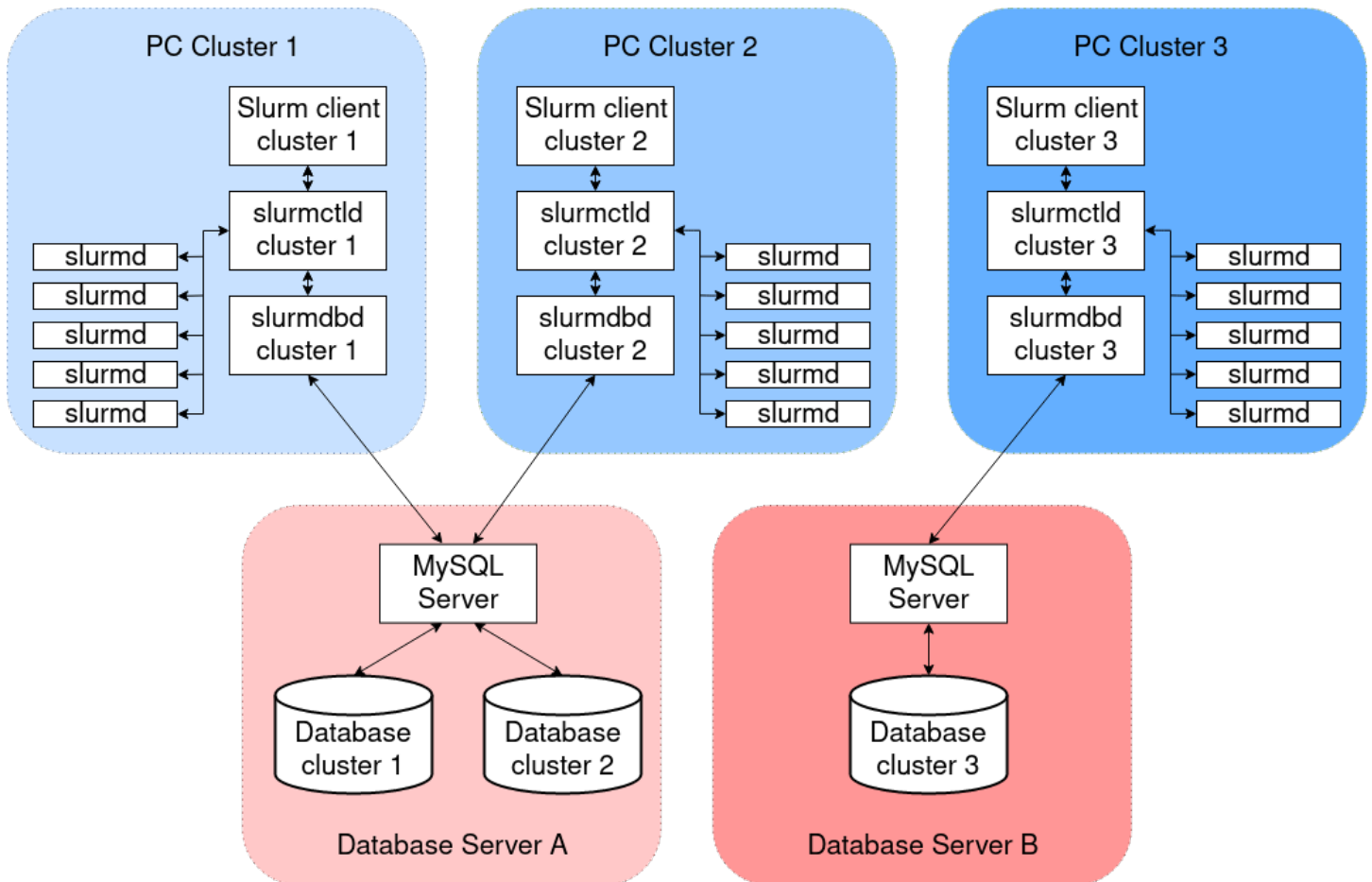
AWS ParallelCluster mengkonfigurasi [slurmdbd](#) untuk memastikan bahwa cluster memiliki Slurm database sendiri di server database. Server database yang sama dapat digunakan di beberapa

cluster, tetapi setiap cluster memiliki database tersendiri. AWS ParallelCluster menggunakan nama cluster untuk menentukan nama untuk database dalam [StorageLoc](#)parameter file `slurmdbd` konfigurasi. Pertimbangkan situasi berikut. Database yang ada di server database menyertakan nama cluster yang tidak dipetakan ke nama cluster aktif. Dalam hal ini, Anda dapat membuat cluster baru dengan nama cluster tersebut untuk dipetakan ke database tersebut. Slurmmenggunakan kembali database untuk cluster baru.

Warning

- Kami tidak menyarankan menyiapkan lebih dari satu cluster untuk menggunakan database yang sama sekaligus. Melakukannya dapat menyebabkan masalah kinerja atau bahkan situasi kebuntuan database.
- Jika Slurm akuntansi diaktifkan pada node kepala cluster, sebaiknya gunakan tipe instans dengan CPU yang kuat, lebih banyak memori, dan bandwidth jaringan yang lebih tinggi. Slurmakuntansi dapat menambah ketegangan pada simpul kepala cluster.

Dalam arsitektur fitur AWS ParallelCluster Slurm akuntansi saat ini, setiap cluster memiliki instance `slurmdbd` daemon sendiri seperti yang ditunjukkan pada konfigurasi contoh diagram berikut.



Jika Anda menambahkan fungsionalitas Slurm multi-cluster atau federasi khusus ke lingkungan cluster Anda, semua cluster harus mereferensikan instance yang sama. `slurmdbd` Untuk alternatif ini, kami menyarankan Anda mengaktifkan AWS ParallelCluster Slurm akuntansi pada satu cluster dan secara manual mengkonfigurasi cluster lain untuk terhubung ke `slurmdbd` yang di-host pada cluster pertama.

Jika Anda menggunakan AWS ParallelCluster versi sebelum versi 3.3.0, lihat metode alternatif untuk menerapkan Slurm akuntansi yang dijelaskan dalam Posting [Blog HPC](#) ini.

Slurmpertimbangan akuntansi

Database dan cluster pada VPC yang berbeda

Untuk mengaktifkan Slurm akuntansi, server database diperlukan untuk berfungsi sebagai backend untuk operasi baca dan tulis yang dilakukan `slurmdbd` daemon. Sebelum cluster dibuat atau diperbarui untuk mengaktifkan Slurm akuntansi, node kepala harus dapat mencapai server database.

Jika Anda perlu menyebarkan server database pada VPC selain yang digunakan cluster, pertimbangkan hal berikut:

- Untuk mengaktifkan komunikasi antara `slurmdbd` sisi cluster dan server database, Anda harus mengatur konektivitas antara dua VPC. Untuk informasi selengkapnya, lihat [VPC Peering di Panduan Pengguna Amazon Virtual Private Cloud](#).
- Anda harus membuat grup keamanan yang ingin Anda lampirkan ke node kepala pada VPC cluster. Setelah dua VPC telah diintip, cross-linking antara sisi database dan kelompok keamanan sisi cluster tersedia. Untuk informasi selengkapnya, lihat [Aturan Grup Keamanan](#) di Panduan Pengguna Amazon Virtual Private Cloud.

Mengkonfigurasi enkripsi TLS antara `slurmdbd` dan server database

Dengan konfigurasi Slurm akuntansi default yang AWS ParallelCluster menyediakan, `slurmdbd` menetapkan koneksi terenkripsi TLS ke server database, jika server mendukung enkripsi TLS. AWS layanan database seperti Amazon RDS dan Amazon Aurora mendukung enkripsi TLS secara default.

Anda dapat memerlukan koneksi aman di sisi server dengan mengatur `require_secure_transport` parameter pada server database. Ini dikonfigurasi dalam CloudFormation template yang disediakan.

Mengikuti praktik keamanan terbaik, kami menyarankan Anda juga mengaktifkan verifikasi identitas server pada `slurmdbd` klien. Untuk melakukan ini, konfigurasi [StorageParameters](#) di `slurmdbd.conf`. Unggah sertifikat CA server ke node kepala cluster. Selanjutnya, atur opsi [SSL_CA](#) dari `StorageParameters` in `slurmdbd.conf` ke jalur sertifikat CA server pada node kepala. Melakukan hal ini memungkinkan verifikasi identitas server di `slurmdbd` samping. Setelah Anda membuat perubahan ini, restart `slurmdbd` layanan untuk membangun kembali konektivitas ke server database dengan verifikasi identitas diaktifkan.

Memperbarui kredensi database

Untuk memperbarui nilai untuk [Database/UserName](#) atau [PasswordSecretArn](#), Anda harus terlebih dahulu menghentikan armada komputasi. Misalkan nilai rahasia yang disimpan dalam AWS Secrets Manager rahasia diubah dan ARN-nya tidak berubah. Dalam situasi ini, cluster tidak secara otomatis memperbarui kata sandi database ke nilai baru. Untuk memperbarui cluster untuk nilai rahasia baru, jalankan perintah berikut dari node kepala.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```


Warning

Untuk menghindari kehilangan data akuntansi, sebaiknya Anda hanya mengubah kata sandi database saat armada komputasi dihentikan.

Pemantauan basis data

Kami menyarankan Anda mengaktifkan fitur pemantauan layanan AWS database. Untuk informasi selengkapnya, lihat pemantauan [Amazon RDS](#) atau dokumentasi pemantauan [Amazon Aurora](#).

Slurm kustomisasi konfigurasi

Dimulai dengan AWS ParallelCluster versi 3.6.0, Anda dapat menyesuaikan `slurm.conf` Slurm konfigurasi dalam AWS ParallelCluster konfigurasi cluster.

Dalam konfigurasi cluster, Anda dapat menyesuaikan Slurm parameter konfigurasi dengan menggunakan pengaturan konfigurasi cluster berikut:

- Sesuaikan Slurm parameter untuk seluruh cluster dengan menggunakan salah satu [SlurmSettings/CustomSlurmSettings](#) atau [CustomSlurmSettingsIncludeFile](#) parameter. AWS ParallelCluster gagal jika Anda menentukan keduanya.
- Sesuaikan Slurm parameter untuk antrian dengan menggunakan [SlurmQueues/CustomSlurmSettings](#) (dipetakan ke Slurm partisi).
- Sesuaikan Slurm parameter untuk sumber daya komputasi dengan menggunakan [SlurmQueues/ComputeResources/CustomSlurmSettings](#) (dipetakan ke Slurm simpul).

Slurm batasan kustomisasi konfigurasi dan pertimbangan saat menggunakan AWS ParallelCluster

- Untuk [CustomSlurmSettings](#) dan [CustomSlurmSettingsIncludeFile](#) pengaturan, Anda hanya dapat menentukan dan memperbarui `slurm.conf` parameter yang termasuk dalam [Slurm versi](#) yang didukung oleh AWS ParallelCluster versi yang Anda gunakan untuk mengkonfigurasi cluster.
- Jika Anda menentukan kustom Slurm konfigurasi di salah satu [CustomSlurmSettings](#) parameter, AWS ParallelCluster melakukan pemeriksaan validasi

dan mencegah pengaturan atau pembaruanSlurmparameter konfigurasi yang bertentangan denganAWS ParallelClusterlogika. TheSlurmparameter konfigurasi yang diketahui bertentangan denganAWS ParallelClusterdiidentifikasi dalam daftar penolakan. Daftar penolakan dapat berubah di masa mendatangAWS ParallelClusterversi jika lainnyaSlurmfitur ditambahkan. Untuk informasi selengkapnya, lihat [Terdaftar penolakanSlurmparameter konfigurasi untukCustomSlurmSettings](#).

- AWS ParallelClusterhanya memeriksa apakah parameter ada dalam daftar penolakan.AWS ParallelClustertidak memvalidasi kustom AndaSlurmsintaks parameter konfigurasi atau semantik. Anda bertanggung jawab untuk memvalidasi kebiasaan AndaSlurmparameter konfigurasi. Kustom tidak validSlurmparameter konfigurasi dapat menyebabkanSlurmkegagalan daemon yang dapat menyebabkan kegagalan pembuatan dan pembaruan cluster.
- Jika Anda menentukan kustomSlurmkonfigurasi diCustomSlurmSettingsIncludeFile,AWS ParallelClustertidak melakukan validasi apa pun.
- Anda dapat memperbaruiCustomSlurmSettingsdanCustomSlurmSettingsIncludeFiletanpa berhenti dan memulai armada komputasi. Dalam hal ini,AWS ParallelClustermemulai ulangslurmctlddaemon dan menjalankanscontrol reconfigureperintah.

BeberapaSlurmparameter konfigurasi mungkin memerlukan operasi yang berbeda sebelum perubahan terdaftar di seluruh cluster. Misalnya, mereka mungkin memerlukan restart semua daemon di cluster. Anda bertanggung jawab untuk memverifikasi apakahAWS ParallelClusteroperasi cukup untuk menyebarkan kustom AndaSlurmpengaturan parameter konfigurasi selama pembaruan. Jika Anda menemukan ituAWS ParallelClusteroperasi tidak cukup, Anda bertanggung jawab untuk memberikan tindakan tambahan yang diperlukan untuk menyebarkan pengaturan yang diperbarui seperti yang direkomendasikan di[Slurmdokumentasi](#).

Terdaftar penolakanSlurmparameter konfigurasi untuk**CustomSlurmSettings**

Tabel berikut mencantumkan parameter denganAWS ParallelClusterversi yang menolak penggunaannya, dimulai dengan versi 3.6.0.CustomSlurmSettingstidak didukung untukAWS ParallelClusterversi lebih awal dari versi 3.6.0.

Parameter yang terdaftar penolakan di tingkat cluster:

Slurmparameter	Deny-terdaftar diAWS ParallelClusterversi
CommunicationParameters	3.6.0

Slurmparameter	Deny-terdaftar diAWS ParallelClusterversi
Epilog	3.6.0
GresTypes	3.6.0
LaunchParameters	3.6.0
Prolog	3.6.0
ReconfigFlags	3.6.0
ResumeFailProgram	3.6.0
ResumeProgram	3.6.0
ResumeTimeout	3.6.0
SlurmctldHost	3.6.0
SlurmctldLogFile	3.6.0
SlurmctldParameters	3.6.0
SlurmdLogfile	3.6.0
SlurmUser	3.6.0
SuspendExcNodes	3.6.0
SuspendProgram	3.6.0
SuspendTime	3.6.0
TaskPlugin	3.6.0
TreeWidth	3.6.0

Parameter yang terdaftar penolakan pada tingkat cluster saat [asli Slurm integrasi akuntansi](#) dikonfigurasi dalam konfigurasi cluster:

Slurmparameter	Deny-terdaftar diAWS ParallelClusterversi
AccountingStorageType	3.6.0
AccountingStorageHost	3.6.0
AccountingStoragePort	3.6.0
AccountingStorageUser	3.6.0
JobAcctGatherType	3.6.0

Parameter yang terdaftar penolakan pada tingkat antrian (partisi) untuk antrian yang dikelola olehAWS ParallelCluster:

Slurmparameter	Deny-terdaftar diAWS ParallelClusterversi
Simpul	3.6.0
PartitionName	3.6.0
ResumeTimeout	3.6.0
Status	3.6.0
SuspendTime	3.6.0

Parameter yang terdaftar penolakan pada tingkat sumber daya komputasi (node) untuk sumber daya komputasi yang dikelola olehAWS ParallelCluster:

Slurmparameter	Deny-terdaftar diAWS ParallelClusterversi dan versi yang lebih baru
CPU	3.6.0
Fitur	3.6.0

Slurmparameter	Deny-terdaftar diAWS ParallelClusterversi dan versi yang lebih baru
Gres	3.6.0
NodeAddr	3.6.0
NodeHostname	3.6.0
NodeName	3.6.0
Berat Badan	3.7.0

Slurm **prolog** dan **epilog**

Dimulai dengan AWS ParallelCluster versi 3.6.0, Slurm konfigurasi yang diterapkan dengan AWS ParallelCluster menyertakan parameter Prolog dan Epilog konfigurasi:

```
# PROLOG AND EPILOG
Prolog=/opt/slurm/etc/scripts/prolog.d/*
Epilog=/opt/slurm/etc/scripts/epilog.d/*
SchedulerParameters=nohold_on_prolog_fail
BatchStartTimeout=180
```

Untuk informasi selengkapnya, lihat [Panduan Prolog dan Epilog](#) di dokumentasi. Slurm

AWS ParallelClustertermasuk skrip prolog dan epilog berikut:

- 90_plcluster_health_check_manager(dalam Prolog folder)
- 90_pcluster_noop(dalam Epilog folder)

Note

Kedua Epilog folder Prolog dan harus berisi setidaknya satu file.

Anda dapat menggunakan kustom prolog atau epilog skrip Anda sendiri dengan menambahkannya ke Epilog folder Prolog dan yang sesuai.

⚠ Warning

Slurmmenjalankan setiap script dalam folder, dalam urutan abjad terbalik.

Durasi waktu berjalan prolog dan epilog skrip memengaruhi waktu yang dibutuhkan untuk menjalankan pekerjaan. Perbarui pengaturan `BatchStartTimeout` konfigurasi saat menjalankan beberapa atau prolog skrip yang berjalan lama. Waktu default adalah 3 menit.

Jika Anda menggunakan kustom prolog dan epilog skrip, cari skrip di masing-masing Prolog dan Epilog folder. Kami menyarankan agar Anda tetap mempertahankan `90_plcluster_health_check_manager` skrip yang berjalan sebelum setiap skrip kustom. Untuk informasi selengkapnya, lihat [Slurm kustomisasi konfigurasi](#).

Ukuran dan pembaruan kapasitas cluster

Kapasitas cluster ditentukan oleh jumlah node komputasi yang dapat diskalakan oleh cluster. Node komputasi didukung oleh instans EC2 yang didefinisikan dalam sumber daya komputasi dalam AWS ParallelCluster konfigurasi (`Scheduling/SlurmQueues/ComputeResources`), dan diatur ke dalam antrian yang memetakan 1:1 ke partisi. (`Scheduling/SlurmQueues`) Slurm

Dalam sumber daya komputasi, dimungkinkan untuk mengonfigurasi jumlah minimum node komputasi (instance) yang harus selalu berjalan di cluster (`MinCount`), dan jumlah maksimum instance yang dapat diskalakan oleh sumber daya komputasi ke (3). `MaxCount`

Pada waktu pembuatan klaster, atau pada pembaruan klaster, AWS ParallelCluster meluncurkan sebanyak mungkin instance EC2 seperti yang dikonfigurasi `MinCount` untuk setiap sumber daya komputasi (`Scheduling/SlurmQueues/ComputeResources`) yang ditentukan dalam cluster. Instance yang diluncurkan untuk mencakup jumlah minimal node untuk sumber daya komputasi di cluster disebut node statis. Setelah dimulai, node statis dimaksudkan untuk persisten di cluster dan mereka tidak dihentikan oleh sistem, kecuali peristiwa atau kondisi tertentu terjadi. Peristiwa tersebut termasuk, misalnya, kegagalan Slurm atau pemeriksaan kesehatan EC2 dan perubahan status node Slurm menjadi DRAIN atau DOWN.

Instans EC2, dalam kisaran **1** hingga **'MaxCount - MinCount'** (**MaxCount** minus **MinCount**), diluncurkan sesuai permintaan untuk menangani peningkatan beban cluster, disebut sebagai node dinamis. Sifat mereka fana, mereka diluncurkan untuk melayani pekerjaan yang tertunda dan dihentikan setelah mereka tetap menganggur untuk jangka waktu yang ditentukan oleh

Scheduling/SlurmSettings/[ScaledownIdleTime](#) dalam konfigurasi cluster (default: 10 menit).

Node statis dan simpul dinamis mematuhi skema penamaan berikut:

- Node statis `<Queue/Name>-st-<ComputeResource/Name>-<num>` di mana `<num> = 1..ComputeResource/MinCount`
- Node dinamis `<Queue/Name>-dy-<ComputeResource/Name>-<num>` di mana `<num> = 1..(ComputeResource/MaxCount - ComputeResource/MinCount)`

Misalnya diberikan AWS ParallelCluster konfigurasi berikut:

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

Node berikut akan didefinisikan dalam Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up      infinite  100    idle queue1-st-c5xlarge-[1-100]
```

Ketika sumber daya komputasi memiliki `MinCount == MaxCount`, semua node komputasi yang sesuai akan statis dan semua instance akan diluncurkan pada waktu pembuatan/pembaruan cluster dan terus berjalan. Sebagai contoh:

```
Scheduling:
  Scheduler: slurm
```

```
SlurmQueues:
- Name: queue1
  ComputeResources:
  - Name: c5xlarge
    Instances:
    - InstanceType: c5.xlarge
  MinCount: 100
  MaxCount: 100
```

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up       infinite   100   idle queue1-st-c5xlarge-[1-100]
```

Pembaruan kapasitas cluster

Pembaruan kapasitas cluster termasuk menambahkan atau menghapus antrian, menghitung sumber daya atau mengubah sumber daya komputasi. MinCount/MaxCount Mulai dari AWS ParallelCluster versi 3.9.0, mengurangi ukuran antrian memerlukan armada komputasi dihentikan atau [QueueUpdateStrategy](#) disetel ke TERMINATE sebelum pembaruan klaster berlangsung. Tidak perlu menghentikan armada komputasi atau menyetel [QueueUpdateStrategy](#) ke TERMINATE saat:

- Menambahkan antrian baru ke Penjadwalan/[SlurmQueues](#)
- Menambahkan sumber daya komputasi baru Scheduling/SlurmQueues/[ComputeResources](#) ke antrian
- Meningkatkan [MaxCount](#) sumber daya komputasi
- Peningkatan MinCount sumber daya komputasi dan peningkatan MaxCount sumber daya komputasi yang sama setidaknya dengan jumlah yang sama

Pertimbangan dan batasan

Bagian ini dimaksudkan untuk menguraikan faktor penting, kendala, atau batasan yang harus diperhitungkan saat mengubah ukuran kapasitas cluster.

- Saat menghapus antrian dari Scheduling/[SlurmQueues](#) semua node komputasi dengan nama<Queue/Name>-*, baik statis maupun dinamis, akan dihapus dari Slurm konfigurasi dan instance EC2 yang sesuai akan dihentikan.
- Saat menghapus sumber daya komputasi Scheduling/[SlurmQueues](#)/[ComputeResources](#) dari antrian, semua node komputasi dengan nama<Queue/Name>-*-<ComputeResource/Name>-*, baik statis maupun dinamis, akan dihapus dari Slurm konfigurasi dan instance EC2 yang sesuai akan dihentikan.

Ketika mengubah MinCount parameter sumber daya komputasi kita dapat membedakan dua skenario yang berbeda, jika MaxCount tetap sama dengan MinCount (kapasitas statis saja), dan jika MaxCount lebih besar dari MinCount (kapasitas statis dan dinamis campuran).

Perubahan kapasitas hanya dengan node statis

- Jika `MinCount == MaxCount`, saat meningkatkan MinCount (dan MaxCount), cluster akan dikonfigurasi dengan memperluas jumlah node statis ke nilai baru MinCount `<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` dan sistem akan terus mencoba meluncurkan instans EC2 untuk memenuhi kapasitas statis baru yang diperlukan.
- Jika `MinCount == MaxCount`, ketika mengurangi MinCount (dan MaxCount) jumlah N, cluster akan dikonfigurasi dengan menghapus node statis N terakhir `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`] dan sistem akan menghentikan instance EC2 yang sesuai.

- Keadaan awal `MinCount = MaxCount = 100`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   100   idle queue1-st-c5xlarge-[1-100]
```

- Update `-30` pada MinCount dan MaxCount: `MinCount = MaxCount = 70`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite    70   idle queue1-st-c5xlarge-[1-70]
```

Perubahan kapasitas dengan node campuran

Jika $\text{MinCount} < \text{MaxCount}$, ketika meningkat MinCount dengan jumlah N (dengan asumsi MaxCount akan tetap tidak berubah), cluster akan dikonfigurasi dengan memperluas jumlah node statis ke nilai baru MinCount ($\text{old_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` dan sistem akan terus mencoba meluncurkan instans EC2 untuk memenuhi kapasitas statis baru yang diperlukan. Selain itu, untuk menghormati MaxCount kapasitas sumber daya komputasi, konfigurasi cluster diperbarui dengan menghapus node dinamis N terakhir: `<Queue/Name>-dy-<ComputeResource/Name>- [<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` dan sistem akan menghentikan instance EC2 yang sesuai.

- Keadaan awal: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite    50   idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite   100   idle queue1-st-c5xlarge-[1-100]
```

- Perbarui +30 ke MinCount : $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite    20   idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up    infinite   130   idle queue1-st-c5xlarge-[1-130]
```

Jika $\text{MinCount} < \text{MaxCount}$, ketika meningkatkan MinCount dan MaxCount dengan jumlah yang sama N , cluster akan dikonfigurasi dengan memperluas jumlah node statis ke nilai baru MinCount ($\text{old_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` dan sistem akan terus mencoba meluncurkan instans EC2 untuk memenuhi kapasitas statis baru yang diperlukan. Selain itu, tidak ada perubahan yang akan dilakukan pada jumlah node dinamis untuk menghormati yang baru

MaxCount nilai.

- Keadaan awal: $\text{MinCount} = 100$; $\text{MaxCount} = 150$

- ```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Perbarui +30 ke MinCount : MinCount = 130 (MaxCount = 180)

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

Jika $\text{MinCount} < \text{MaxCount}$, ketika mengurangi MinCount jumlah N (dengan asumsi MaxCount akan tetap tidak berubah), cluster akan dikonfigurasi dengan menghapus node statis N terakhir node statis `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>` dan sistem akan menghentikan instance EC2 yang sesuai. Selain itu, untuk menghormati MaxCount kapasitas sumber daya komputasi, konfigurasi cluster diperbarui dengan memperluas jumlah node dinamis untuk mengisi celah $\text{MaxCount} - \text{new_MinCount}$: `<Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]` Dalam hal ini, karena itu adalah node dinamis, tidak ada instance EC2 baru yang akan diluncurkan kecuali penjadwal memiliki pekerjaan di pending pada node baru.

- Keadaan awal: MinCount = 100; MaxCount = 150

- ```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Perbarui -30 pada MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 80 idle~ queue1-dy-c5xlarge-[1-80]
queue1* up infinite 70 idle queue1-st-c5xlarge-[1-70]
```

Jika  $\text{MinCount} < \text{MaxCount}$ , ketika mengurangi  $\text{MinCount}$  dan  $\text{MaxCount}$  dengan jumlah yang sama  $N$ , cluster akan dikonfigurasi dengan menghapus node statis  $N$  terakhir `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] dan sistem akan menghentikan instance EC2 yang sesuai.

Selain itu, tidak ada perubahan yang akan dilakukan pada jumlah node dinamis untuk menghormati  $\text{MaxCount}$  nilai baru.

- Keadaan awal:  $\text{MinCount} = 100$ ;  $\text{MaxCount} = 150$

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Perbarui -30 pada  $\text{MinCount}$  :  $\text{MinCount} = 70$  ( $\text{MaxCount} = 120$ )

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 80 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 70 idle queue1-st-c5xlarge-[1-70]
```

Jika  $\text{MinCount} < \text{MaxCount}$ , ketika mengurangi  $\text{MaxCount}$  jumlah  $N$  (dengan asumsi  $\text{MinCount}$  akan tetap tidak berubah), cluster akan dikonfigurasi dengan menghapus node dinamis  $N$  terakhir `<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>`] dan sistem akan menghentikan instance EC2 yang sesuai jika mereka berjalan. Tidak ada dampak yang diharapkan pada node statis.

- Keadaan awal: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Perbarui -30 pada MaxCount : MinCount = 100 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 20 idle~ queue1-dy-c5xlarge-[1-20]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

## Dampak pada Pekerjaan

Dalam semua kasus di mana node dihapus dan instance EC2 dihentikan, pekerjaan sbatch yang berjalan pada node yang dihapus akan diantrian ulang, kecuali tidak ada node lain yang memenuhi persyaratan pekerjaan. Dalam kasus terakhir ini pekerjaan akan gagal dengan status `NODE_FAIL` dan menghilang dari antrian; jika demikian, itu harus dikirim ulang secara manual.

Jika Anda berencana untuk melakukan pembaruan pengubahan ukuran cluster, Anda dapat mencegah pekerjaan berjalan di node yang akan dihapus selama pembaruan yang direncanakan. Ini dimungkinkan dengan mengatur node yang akan dihapus dalam pemeliharaan. Perlu diketahui bahwa menyetel node dalam pemeliharaan tidak akan memengaruhi pekerjaan yang pada akhirnya sudah berjalan di node.

Misalkan dengan pembaruan pengubahan ukuran cluster yang direncanakan Anda akan menghapus `nodequeueu-st-computeresource-[9-10]`. Anda dapat membuat Slurm reservasi dengan perintah berikut

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queueu-st-
computeresource-[9-10]
```

Ini akan membuat Slurm reservasi bernama `maint_for_update` pada `nodequeue-st-computeresource-[9-10]`. Dari saat reservasi dibuat, tidak ada lagi pekerjaan yang bisa berjalan ke `nodequeue-st-computeresource-[9-10]`. Perlu diketahui bahwa reservasi tidak akan mencegah pekerjaan akhirnya dialokasikan pada `nodequeue-st-computeresource-[9-10]`.

Setelah pembaruan perubahan ukuran klaster, jika Slurm reservasi ditetapkan hanya pada node yang telah dihapus selama pembaruan perubahan ukuran, reservasi pemeliharaan akan dihapus secara otomatis. Jika sebaliknya Anda telah membuat Slurm reservasi pada node yang masih ada setelah pembaruan perubahan ukuran cluster, kami mungkin ingin menghapus reservasi pemeliharaan pada node setelah pembaruan perubahan ukuran dilakukan, dengan menggunakan perintah berikut

```
sudo -i scontrol delete ReservationName=maint_for_update
```

Untuk detail tambahan tentang Slurm reservasi, lihat dokumen SchedMD resmi [di sini](#).

Proses pembaruan cluster pada perubahan kapasitas

Setelah perubahan konfigurasi penjadwal, langkah-langkah berikut dijalankan selama proses pembaruan cluster:

- Berhenti AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- Hasilkan konfigurasi Slurm partisi yang diperbarui dari AWS ParallelCluster konfigurasi
- Mulai ulang `slurmctld` (dilakukan melalui resep layanan Chef)
- Periksa `slurmctld` status (`systemctl is-active --quiet slurmctld.service`)
- Muat ulang konfigurasi Slurm (`scontrol reconfigure`)
- Mulai `clustermgtd` (`supervisorctl start clustermgtd`)

## AWS Batch (**awsbatch**)

Untuk informasi tentang AWS Batch, lihat [AWS Batch](#). Untuk dokumentasi, lihat [Panduan AWS Batch Pengguna](#).

AWS ParallelClusterPerintah CLI untuk AWS Batch

Bila Anda menggunakan `awsbatch` scheduler, perintah AWS ParallelCluster CLI untuk AWS Batch secara otomatis diinstal di node AWS ParallelCluster kepala. CLI menggunakan operasi AWS Batch API dan mengizinkan operasi berikut:

- Kirim dan kelola pekerjaan.
- Memantau pekerjaan, antrian, dan host.
- Cermin perintah penjadwal tradisional.

### Important

AWS ParallelClustertidak mendukung pekerjaan GPU untukAWS Batch. Untuk informasi lebih lanjut, lihat [pekerjaan GPU](#).

CLI ini didistribusikan sebagai paket terpisah. Untuk informasi selengkapnya, lihat [Dukungan Scheduler](#).

### Topik

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

## awsbsub

Mengirimkan pekerjaan ke antrian pekerjaan cluster.

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

### Important

AWS ParallelClustertidak mendukung pekerjaan GPU untukAWS Batch. Untuk informasi selengkapnya, lihat [tugas GPU](#).

## Argumen Posisi

### ***command***

Mengirimkan pekerjaan (perintah yang ditentukan harus tersedia pada contoh komputasi) atau nama file yang akan ditransfer. Lihat juga `--command-file`.

### **arguments**

(Opsional) Menentukan argumen untuk perintah atau file perintah.

## Argumen Bernama

### **-jn *JOB\_NAME*, --job-name *JOB\_NAME***

Nama pekerjaan. Karakter pertama harus berupa huruf atau angka. Nama tugas dapat berisi huruf (huruf besar dan huruf kecil), angka, tanda hubung, dan garis bawah, dan garis bawah, dan panjangnya 128 karakter.

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

Menentukan klaster yang akan digunakan.

### **-cf, --command-file**

Menunjukkan bahwa perintah adalah file yang akan ditransfer ke contoh komputasi.

Default: Salah

### **-w *WORKING\_DIR*, --working-dir *WORKING\_DIR***

Menentukan folder untuk digunakan sebagai direktori kerja pekerjaan. Jika direktori kerja tidak ditentukan, pekerjaan dijalankan di job-`<AWS_BATCH_JOB_ID>` subfolder direktori home pengguna. Anda dapat menggunakan parameter ini atau `--parent-working-dir` parameter.

### **-pw *PARENT\_WORKING\_DIR*, --parent-working-dir *PARENT\_WORKING\_DIR***

Menentukan folder induk dari direktori kerja pekerjaan. Jika direktori kerja induk tidak ditentukan, itu default ke direktori home pengguna. Sebuah subfolder bernama job-`<AWS_BATCH_JOB_ID>` dibuat dalam direktori induk kerja. Anda dapat menggunakan parameter ini atau `--working-dir` parameter.

### **-if *INPUT\_FILE*, --input-file *INPUT\_FILE***

Menentukan file yang akan ditransfer ke contoh komputasi, dalam direktori kerja pekerjaan. Anda dapat menentukan beberapa parameter file input.



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

Menentukan jumlah vCPUs yang akan disimpan untuk kontainer. Ketika digunakan bersama dengan `--nodes`, itu mengidentifikasi jumlah vCPUs untuk setiap node.

Default: 1

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

Menentukan batas keras memori (dalam MiB) untuk menyediakan tugas tersebut. Jika tugas Anda mencoba untuk melebihi batas memori yang ditentukan di sini, tugas Anda akan berakhir.

Default: 128

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

Menentukan daftar dipisahkan koma dari nama variabel lingkungan yang akan diekspor ke lingkungan tugas. Untuk mengeksport semua variabel lingkungan, tentukan 'semua'. Perhatikan bahwa daftar variabel lingkungan 'all' tidak menyertakan variabel yang tercantum dalam `--env-blacklist` parameter, atau variabel yang dimulai dengan `AWS_*` awalan `PCLUSTER_*` atau.

**-eb *ENV\_DENYLIST*, --env-blacklist *ENV\_DENYLIST***

Menentukan daftar dipisahkan koma nama variabel lingkungan untuk tidak ekspor ke lingkungan pekerjaan. Secara default, `HOME`, `PWD`, `USER`, `PATH`, `LD_LIBRARY_PATH`, `TERM`, dan tidak `TERMCAP` diekspor.

**-r *RETRY\_ATTEMPTS*, --retry-attempts *RETRY\_ATTEMPTS***

Menentukan jumlah kali untuk memindahkan tugas ke `RUNNABLE` status. Anda dapat menentukan usaha antara 1 dan 10. Jika nilai percobaan lebih besar dari 1, tugas akan dicoba lagi jika gagal, hingga nilai percobaan lebih besar dari 1, tugas akan dicoba lagi `RUNNABLE` jika gagal, hingga nilai percobaan yang ditentukan.

Default: 1

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

Tentukan durasi waktu dalam detik (diukur dari `startedAt` timestamp percobaan tugas) setelah `AWS Batch` mengakhiri tugas Anda jika belum selesai. Nilai batas waktu harus paling sedikit 60 detik.

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

Menentukan jumlah simpul yang akan disimpan untuk tugas tersebut. Tentukan nilai untuk parameter ini untuk mengaktifkan pengiriman parallel multi-node.

**Note**

Ketika `CapacityType` parameter `SchedulerAwsBatchQueues` diatur ke SPOT, pekerjaan parallel multi-node tidak didukung. Selain itu, harus ada peran `AWSServiceRoleForEC2Spot` terkait layanan di akun Anda. Anda dapat membuat peran ini dengan AWS CLI perintah berikut:

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

Untuk informasi selengkapnya, lihat [peran terkait layanan untuk permintaan Instans Spot](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

**-a *ARRAY\_SIZE*, --array-size *ARRAY\_SIZE***

Menunjukkan ukuran array. Anda dapat menentukan nilai antara 2 hingga 10.000. Jika Anda menentukan properti array untuk suatu tugas, itu menjadi tugas array.

**-d *DEPENDS\_ON*, --depends-on *DEPENDS\_ON***

Menentukan daftar dipisahkan titik koma dependensi untuk pekerjaan. Sebuah pekerjaan dapat bergantung pada maksimum 20 pekerjaan. Anda dapat menentukan ketergantungan SEQUENTIAL tipe tanpa menentukan ID pekerjaan untuk pekerjaan array. Ketergantungan berurutan memungkinkan setiap pekerjaan array turunan untuk menyelesaikan secara berurutan, mulai dari indeks 0. Anda juga dapat menentukan tipe dependensi `N_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO` Dependensi `N_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO_TO`. Sintaks untuk parameter ini adalah "jobid=<string>, type=<string>;...".

**awsbstat**

Menunjukkan pekerjaan yang diajukan dalam antrian pekerjaan kluster.

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

## Argumen Posisi

### *job\_ids*

Menentukan daftar spasi-dipisahkan dari ID pekerjaan untuk menunjukkan dalam output. Jika pekerjaan adalah array pekerjaan, semua pekerjaan anak ditampilkan. Jika satu pekerjaan diminta, itu ditampilkan dalam versi rinci.

### Argumen yang dinamakan

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

Menunjukkan cluster untuk digunakan.

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

Menentukan daftar dipisahkan koma dari status pekerjaan pekerjaan untuk memasukkan. Status pekerjaan default adalah “aktif.”. Nilai yang diterima adalah: SUBMITTED,PENDING,,RUNNABLE,STARTING,RUNNING,SUCCEEDED,FAILED, dan ALL.

Default: “SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING”

#### **-e, --expand-children**

Memperluas pekerjaan dengan anak-anak (baik array dan multi-node parallel).

Default: Salah

#### **-d, --details**

Menunjukkan rincian pekerjaan.

Default: Salah

## **awsbout**

Menunjukkan output dari pekerjaan tertentu.

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

## Argumen Posisi

### ***job\_id***

Menentukan ID pekerjaan.

## Argumen Bernama

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

Menunjukkan cluster untuk digunakan.

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

Mendapat baris *HEAD* pertama dari output pekerjaan.

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

Mendapat <tail>baris terakhir dari output pekerjaan.

### **-s, --stream**

Mendapat output pekerjaan, dan kemudian menunggu output tambahan yang akan diproduksi. Argumen ini dapat digunakan bersama-sama dengan `—tail` untuk memulai dari <tail>baris terbaru dari output pekerjaan.

Default: Salah

### **-sp *STREAM\_PERIOD*, --stream-period *STREAM\_PERIOD***

Menetapkan periode streaming.

Default: 5

## **awsbkill**

Membatalkan atau mengakhiri pekerjaan yang diajukan dalam klaster.

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

## Argumen Posisi

### ***job\_ids***

Menentukan daftar spasi dipisahkan dari ID pekerjaan untuk membatalkan atau mengakhiri.

## Argumen Nama

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

Menunjukkan klaster untuk digunakan.

**-r *REASON*, --reason *REASON***

Menunjukkan pesan untuk dilampirkan ke pekerjaan, menjelaskan alasan untuk membatalkannya.

Bawaan: "Terminated by the user"

## awsbqueues

Menunjukkan antrian pekerjaan yang dikaitkan dengan klaster.

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

### Argumen posisi

#### *job\_queues*

Menentukan daftar yang dipisahkan oleh spasi dari nama antrian yang akan ditampilkan. Jika satu antrian diminta, itu ditampilkan dalam versi rinci.

### Argumen bernama

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

Menentukan nama klaster yang akan digunakan.

**-d, --details**

Menunjukkan apakah akan menampilkan detail antrian.

Default: Salah

## awsbhosts

Menunjukkan host yang termasuk dalam lingkungan komputasi klaster.

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

## Argumen Posisi

### ***instance\_ids***

Menentukan daftar ID ID yang dipisahkan oleh spasi. Jika satu contoh diminta, itu ditampilkan dalam versi rinci.

## Argumen yang Dinamakan

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

Menentukan nama klaster yang akan digunakan.

### **-d, --details**

Menunjukkan apakah akan menampilkan detail host.

Default: Salah

## Penyimpanan bersama

AWS ParallelCluster [mendukung baik menggunakan Amazon EBS, FSx untuk ONTAP, dan FSx untuk volume penyimpanan bersama OpenZFS, Amazon EFS dan FSx for Lustre sistem file penyimpanan bersama, atau File Cache](#). Kami menyarankan Anda mengikuti panduan [pilar keandalan AWS kerangka kerja yang dirancang dengan baik](#) dan mencadangkan volume dan sistem file Anda.

Pilih sistem penyimpanan yang memenuhi persyaratan I/O aplikasi HPC Anda. Anda dapat mengoptimalkan setiap sistem file berdasarkan kasus penggunaan spesifik Anda. Untuk informasi selengkapnya, lihat [ikhtisar opsi penyimpanan](#).

Volume Amazon EBS dilampirkan ke node kepala dan dibagikan dengan node komputasi melalui NFS. Opsi ini dapat hemat biaya, tetapi kinerja tergantung pada sumber daya node kepala karena skala kebutuhan penyimpanan. Ini bisa menjadi hambatan karena lebih banyak node komputasi ditambahkan ke cluster dan permintaan throughput meningkat.

Sistem file Amazon EFS diskalakan karena kebutuhan penyimpanan berubah. Anda dapat mengonfigurasi sistem file ini untuk berbagai kasus penggunaan. Gunakan sistem file Amazon EFS untuk menjalankan aplikasi paralel dan sensitif latensi di cluster Anda.

Sistem file FSx for Lustre dapat memproses kumpulan data besar-besaran hingga ratusan gigabyte per detik throughput, jutaan IOPS, dan latensi sub-milidetik. Gunakan sistem file FSx for Lustre untuk menuntut lingkungan komputasi kinerja tinggi.

Di dalam [Bagian SharedStorage](#), Anda dapat menentukan penyimpanan eksternal atau AWS ParallelCluster terkelola:

- Penyimpanan eksternal mengacu pada volume atau sistem file yang ada yang Anda kelola. AWS ParallelClustertidak membuat atau menghapus penyimpanan ini.
- Penyimpanan terkelola mengacu pada volume atau sistem file yang AWS ParallelCluster dibuat dan dapat dihapus.

### Penyimpanan eksternal

Anda dapat mengonfigurasi AWS ParallelCluster untuk melampirkan penyimpanan eksternal ke cluster saat cluster dibuat atau diperbarui. Demikian pula Anda dapat mengonfigurasinya untuk melepaskan penyimpanan eksternal dari cluster saat cluster dihapus atau diperbarui. Data Anda disimpan dan Anda dapat menggunakannya untuk penyimpanan bersama permanen jangka panjang di luar siklus hidup klaster.

#### Note

Versi AWS ParallelCluster sebelum 3.8 tidak memungkinkan sistem file yang dikelola secara eksternal untuk dipasang di `/home`. Mulai dari versi 3.8, AWS ParallelCluster memungkinkan Anda untuk menggunakan `/home` sebagai titik pemasangan untuk sistem file terkelola eksternal. Anda dapat memasang sistem file yang dikelola secara eksternal `/home` dengan menentukan `/home` sebagai nilai ke `MountDir` parameter di bawah. [Bagian SharedStorage](#) Amazon File Cache tidak cocok untuk digunakan sebagai `/home` direktori sistem dan oleh karena itu tidak didukung saat ini untuk pemasangan `/home`.

Saat menentukan `/home` direktori di bawah opsi `SharedStorageType` konfigurasi akan diganti, artinya pengaturan di bawah [Bagian SharedStorage](#) akan digunakan sebagai gantinya.

#### [Bagian SharedStorage](#)

Saat memasang sistem file eksternal ke `/home` direktori AWS ParallelCluster menyalin `/home` konten node kepala ke sistem file eksternal, tanpa menimpa file yang ada di penyimpanan eksternal. Ini termasuk mentransfer kunci SSH cluster untuk pengguna default, jika tidak ada pada sistem file eksternal. Untuk informasi lebih lanjut lihat [AWS ParallelCluster pertimbangan penyimpanan bersama](#).

## AWS ParallelCluster penyimpanan terkelola

AWS ParallelCluster penyimpanan terkelola bergantung pada siklus hidup cluster secara default dalam konfigurasi. Parameter `SharedStorage DeletionPolicy` konfigurasi diatur ke secara `Delete` default.

Secara default, sistem file AWS ParallelCluster terkelola atau volume dan datanya dihapus jika salah satu dari berikut ini benar.

- Anda menghapus cluster.
- Anda mengubah konfigurasi penyimpanan bersama yang dikelolaName.
- Anda menghapus penyimpanan bersama yang dikelola dari konfigurasi.

Setel `DeletionPolicy Retain` untuk mempertahankan sistem file bersama terkelola atau volume dan data Anda. Kami menyarankan Anda mencadangkan data Anda secara teratur untuk menghindari hilangnya data. Anda dapat menggunakannya [AWS Backup](#) untuk mengelola cadangan secara terpusat untuk semua opsi penyimpanan Anda.

Anda dapat menghapus ketergantungan siklus hidup dengan pengaturan konfigurasi. Untuk informasi selengkapnya, lihat [Ubah penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal](#).

Untuk informasi tentang kuota penyimpanan bersama, lihat [Kuota untuk penyimpanan bersama](#).

Untuk informasi selengkapnya tentang penyimpanan bersama dan beralih ke AWS ParallelCluster versi baru, lihat [Praktik terbaik: memindahkan klaster ke versi AWS ParallelCluster minor atau patch baru](#).

Anda dapat mengonfigurasi AWS ParallelCluster untuk melampirkan penyimpanan eksternal ke cluster saat cluster dibuat atau diperbarui. Demikian pula, Anda dapat mengonfigurasinya untuk melepaskan penyimpanan eksternal dari cluster saat cluster dihapus atau diperbarui. Data Anda disimpan dan Anda dapat menggunakannya untuk solusi penyimpanan bersama permanen jangka panjang yang independen dari siklus hidup klaster.

Secara default, penyimpanan terkelola bergantung pada siklus hidup klaster. Anda dapat menghapus ketergantungan dengan pengaturan konfigurasi yang dijelaskan di [Ubah penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal](#).

Dengan pengaturan tertentu, Anda dapat mengoptimalkan setiap solusi penyimpanan yang didukung untuk kasus penggunaan Anda.



Untuk kuota penyimpanan bersama, lihat [Kuota untuk penyimpanan bersama](#).

Untuk informasi selengkapnya tentang penyimpanan bersama dan beralih ke AWS ParallelCluster versi baru, lihat [Praktik terbaik: memindahkan klaster ke versi AWS ParallelCluster minor atau patch baru](#).

Topik

- [Konfigurasi penyimpanan bersama](#)
- [Bekerja dengan penyimpanan bersama di AWS ParallelCluster](#)
- [Kuota untuk penyimpanan bersama](#)

## Konfigurasi penyimpanan bersama

Pelajari tentang pengaturan konfigurasi yang dapat Anda gunakan untuk menentukan penyimpanan bersama untuk klaster Anda.

Topik

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx for Lustre](#)
- [Konfigurasi fsX untuk ONTAP, FSx untuk OpenZFS, dan penyimpanan bersama Cache File](#)

## Amazon Elastic Block Store

Untuk menggunakan volume Amazon EBS eksternal yang ada untuk penyimpanan permanen jangka panjang yang tidak tergantung pada siklus hidup cluster, tentukan [EbsSettings/VolumeId](#).

Jika Anda tidak menentukan [VolumeId](#), secara default, AWS ParallelCluster membuat volume EBS terkelola dari [EbsSettings](#) saat cluster Anda dibuat. AWS ParallelCluster juga menghapus volume dan data ketika cluster dihapus atau volume dihapus dari konfigurasi cluster.

Untuk volume EBS AWS ParallelCluster terkelola, Anda dapat menggunakan [EbsSettings/DeletionPolicy](#) AWS ParallelCluster untuk menginstruksikan DeleteRetain,, atau Snapshot volume saat cluster dihapus atau ketika volume dihapus dari konfigurasi cluster. Secara default, DeletionPolicy diatur ke Delete.

**⚠ Warning**

Untuk penyimpanan bersama AWS ParallelCluster terkelola, `DeletionPolicy` diatur ke secara `Delete` default.

Ini berarti bahwa, jika salah satu dari berikut ini benar, volume terkelola dan datanya dihapus:

- Anda menghapus cluster.
- Anda mengubah konfigurasi penyimpanan bersama yang dikelola [SharedStorage/Name](#).
- Anda menghapus penyimpanan bersama yang dikelola dari konfigurasi.

Kami menyarankan Anda mencadangkan penyimpanan bersama Anda dengan snapshot secara teratur untuk menghindari hilangnya data. Untuk informasi selengkapnya tentang snapshot Amazon EBS, lihat snapshot [Amazon EBS di](#) Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux. Untuk mempelajari cara mengelola cadangan data Layanan AWS, lihat Backup di [AWS Panduan AWS Backup](#) Pengembang.

## Amazon Elastic File System

Untuk menggunakan sistem file Amazon EFS eksternal yang ada untuk penyimpanan permanen jangka panjang di luar siklus hidup kluster, tentukan [EfsSettings/FileSystemId](#), secara default, AWS ParallelCluster membuat sistem file Amazon EFS terkelola [EfsSettings](#) sejak membuat cluster. AWS ParallelCluster juga menghapus sistem file dan data ketika cluster dihapus atau ketika sistem file dihapus dari konfigurasi cluster.

Untuk sistem file Amazon EFS AWS ParallelCluster terkelola, Anda dapat menggunakan [EfsSettings/DeletionPolicy](#) untuk menginstruksikan AWS ParallelCluster `Delete` `Retain` atau kapan kluster dihapus atau ketika sistem file dihapus dari konfigurasi cluster. Secara default, `DeletionPolicy` diatur ke `Delete`.

**⚠ Warning**

Untuk penyimpanan bersama AWS ParallelCluster terkelola, `DeletionPolicy` diatur ke secara `Delete` default.

Ini berarti bahwa, jika salah satu dari berikut ini benar, sistem file terkelola dan datanya dihapus:

- Anda menghapus cluster.

- Anda mengubah konfigurasi penyimpanan bersama yang dikelola [SharedStorage/Name](#).
- Anda menghapus penyimpanan bersama yang dikelola dari konfigurasi.

Kami menyarankan Anda mencadangkan penyimpanan bersama Anda secara teratur untuk menghindari hilangnya data. Untuk informasi selengkapnya tentang cara mencadangkan volume Amazon EFS individual, lihat [Mencadangkan sistem file Amazon EFS Anda](#) di Panduan Pengguna Amazon Elastic File System. Untuk mempelajari cara mengelola cadangan data Layanan AWS, lihat Backup di [AWS Panduan AWS Backup](#) Pengembang.

## Amazon FSx for Lustre

Untuk menggunakan sistem file FSx for Lustre eksternal yang ada untuk penyimpanan permanen jangka panjang di luar siklus hidup cluster, tentukan/. [FsxLustreSettingsFileSystemId](#)

Jika Anda tidak menentukan [FsxLustreSettings/FileSystemId](#), secara default, AWS ParallelCluster membuat [FsxLustreSettings](#) sistem file FSx for Lustre terkelola dari saat membuat cluster. AWS ParallelCluster juga menghapus sistem file dan data ketika cluster dihapus atau ketika sistem file dihapus dari konfigurasi cluster.

Untuk sistem file FSx for Lustre yang AWS ParallelCluster dikelola, Anda dapat [FsxLustreSettings](#) menggunakan [DeletionPolicy](#)/AWS ParallelCluster untuk Delete menginstruksikan Retain ke atau sistem file ketika cluster dihapus atau ketika sistem file dihapus dari konfigurasi cluster. Secara default, [DeletionPolicy](#) diatur ke Delete.

### Warning

Untuk penyimpanan bersama AWS ParallelCluster terkelola, [DeletionPolicy](#) diatur ke secara Delete default.

Ini berarti bahwa, jika salah satu dari berikut ini benar, sistem file terkelola dan datanya dihapus:

- Anda menghapus cluster.
- Anda mengubah konfigurasi penyimpanan bersama yang dikelola [SharedStorage/Name](#).
- Anda menghapus penyimpanan bersama yang dikelola dari konfigurasi.

Kami menyarankan Anda mencadangkan penyimpanan bersama Anda secara teratur untuk menghindari hilangnya data. Anda dapat menentukan cadangan di cluster Anda dengan [SharedStorage/FsxLustreSettings/AutomaticBackupRetentionDays](#) dan [DailyAutomaticBackupStartTime](#). Untuk mempelajari cara mengelola cadangan data Layanan AWS, lihat Backup di [AWS Panduan AWS Backup](#) Pengembang.

## Konfigurasi fsX untuk ONTAP, FSx untuk OpenZFS, dan penyimpanan bersama Cache File

Untuk FSx untuk ONTAP, FSx untuk OpenZFS, dan File Cache, Anda dapat menggunakan [FsxOntapSettings/VolumeId](#), dan [FileCacheSettings/VolumeIdFileCacheId](#) untuk menentukan pemasangan volume eksternal yang ada atau File Cache untuk cluster Anda. [FsxOpenZfsSettings](#)

AWS ParallelCluster penyimpanan bersama terkelola tidak didukung untuk FSx untuk ONTAP, FSx untuk OpenZFS, dan Cache File.

## Bekerja dengan penyimpanan bersama di AWS ParallelCluster

Pelajari tentang bekerja dengan AWS ParallelCluster dan penyimpanan bersama.

### Topik


- [AWS ParallelCluster pertimbangan penyimpanan bersama](#)
- [Ubah penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal](#)

## AWS ParallelCluster pertimbangan penyimpanan bersama

Pertimbangkan hal berikut saat bekerja dengan penyimpanan bersama di AWS ParallelCluster.

- Cadangkan data sistem file Anda dengan [AWS Backup](#) atau metode lain untuk mengelola cadangan untuk semua sistem penyimpanan Anda.
- Untuk menambahkan penyimpanan bersama, Anda menambahkan bagian penyimpanan bersama ke file konfigurasi Anda dan membuat atau memperbarui kluster.
- Untuk menghapus penyimpanan bersama, Anda menghapus bagian penyimpanan bersama dari file konfigurasi Anda dan memperbarui cluster.

- Untuk mengganti penyimpanan bersama AWS ParallelCluster terkelola yang ada dengan penyimpanan terkelola baru, ubah nilai untuk [SharedStorage/Name](#) dan perbarui klaster.

 Warning

Secara default, penyimpanan dan data AWS ParallelCluster terkelola yang ada akan dihapus saat Anda melakukan pembaruan klaster dengan Name parameter baru. Jika Anda perlu mengubah Name dan mempertahankan data penyimpanan bersama terkelola yang ada, pastikan Anda menyetel `DeletionPolicy` ke `Retain` atau mencadangkan data sebelum memperbarui klaster.

- Jika Anda tidak mencadangkan data penyimpanan AWS ParallelCluster terkelola `Delete`, data Anda akan dihapus saat klaster Anda dihapus atau ketika penyimpanan terkelola dihapus dari konfigurasi klaster dan klaster diperbarui. `DeletionPolicy`
- Jika Anda tidak mencadangkan data penyimpanan AWS ParallelCluster terkelola `Retain`, sistem file Anda terlepas sebelum cluster dihapus dan dapat dilampirkan kembali ke cluster lain sebagai sistem file eksternal. `DeletionPolicy` Data Anda disimpan.
- Jika penyimpanan AWS ParallelCluster terkelola dihapus dari konfigurasi klaster dan `DeletionPolicy` sekarang `Retain`, penyimpanan tersebut dapat dilampirkan kembali ke klaster sebagai sistem file eksternal dengan data klaster Anda dipertahankan.
- Dimulai dengan AWS ParallelCluster versi 3.4.0, Anda dapat meningkatkan keamanan untuk pemasangan sistem file Amazon EFS dengan mengonfigurasi [SharedStorage/EfsSettings/EncryptionInTransit](#) dan pengaturan. [IamAuthorization](#)
- Saat memasang sistem file eksternal ke direktori `/home`, AWS ParallelCluster menyalin isi direktori `/home` node head ke sistem file eksternal. Ini menyalin data yang ada di direktori `/home` tanpa menimpa file atau direktori yang ada di penyimpanan eksternal. Ini termasuk kunci SSH cluster untuk pengguna default jika belum ada di sistem file eksternal. Akibatnya semua cluster lain yang memasang sistem file eksternal yang sama ke direktori `/home` masing-masing juga akan memiliki kunci SSH yang sama untuk pengguna default cluster mereka.
- Dalam lingkungan multi-cluster yang memasang sistem file eksternal yang sama ke direktori `/home` cluster, kunci SSH yang memberikan akses ke node komputasi, yang dibuat pada node kepala oleh AWS ParallelCluster, dihasilkan hanya sekali ketika cluster pertama memasang sistem file eksternal ke `/home`. Semua cluster lain menggunakan kunci SSH yang sama. Akibatnya, siapa pun yang memiliki kunci SSH untuk pengguna default cluster bersama ini dapat mengakses cluster apa pun. Semua node komputasi memungkinkan koneksi menggunakan kunci yang awalnya dihasilkan.

## Ubah penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal

Pelajari cara mengonversi penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal.

Prosedur didasarkan pada contoh cuplikan file konfigurasi berikut.

```
...
- MountDir: /fsx
 Name: fsx
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeletionPolicy: Delete
...
```

## Ubah penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal

1. Atur `DeletionPolicy` ke `Retain` dalam file konfigurasi cluster.

```
...
- MountDir: /fsx
 Name: fsx
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeletionPolicy: Retain
...
```

2. Untuk mengatur `DeletionPolicy` perubahan, jalankan perintah berikut.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

3. Hapus `SharedStorage` bagian dari file konfigurasi cluster.

```
...
...
```

4. Untuk mengubah yang dikelola `SharedStorage` ke eksternal `SharedStorage` dan melepaskannya dari cluster, jalankan perintah berikut.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. Penyimpanan bersama Anda sekarang eksternal dan terlepas dari cluster.
6. Untuk melampirkan sistem file eksternal Anda ke cluster asli atau cluster lain, ikuti langkah-langkah ini.

a. Dapatkan ID sistem file FSx for Lustre.

- i. Untuk menggunakan AWS CLI jalankan perintah berikut dan temukan sistem file dengan nama yang menyertakan nama cluster asli Anda dan catat ID sistem file.

```
aws fsx describe-file-systems
```

- ii. Untuk menggunakan AWS Management Console, masuk dan navigasikan ke <https://console.aws.amazon.com/fsx/>. Dalam daftar sistem file, temukan sistem file dengan nama yang menyertakan nama cluster asli Anda dan catat ID sistem file.

b. Perbarui aturan grup keamanan sistem file untuk menyediakan akses ke dan dari sistem file dan subnet cluster. Anda dapat menemukan nama dan ID grup keamanan sistem file di konsol Amazon FSx.

Tambahkan aturan ke grup keamanan sistem file yang memungkinkan lalu lintas TCP masuk dan keluar dari dan ke node kepala dan rentang atau awalan IP CIDR node komputasi. Tentukan port TCP 988, 1021, 1022, dan 1023 untuk lalu lintas TCP masuk dan keluar.

Untuk informasi selengkapnya, lihat

[SharedStorageFsxLustreSettings//FileSystemId](#) dan [Membuat, mengonfigurasi, dan menghapus grup keamanan untuk Amazon EC2](#) di Panduan Pengguna untuk AWS Command Line Interface Versi 2.

c. Tambahkan SharedStorage bagian ke konfigurasi cluster.

```
...
- MountDir: /fsx
 Name: fsx-external
 StorageType: FsxLustre
 FsxLustreSettings:
 FileSystemId: fs-02e5b4b4abd62d51c
...
```

d. Untuk menambahkan penyimpanan bersama eksternal ke cluster, jalankan perintah berikut.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

## Kuota untuk penyimpanan bersama

Konfigurasi cluster SharedStorage untuk memasang penyimpanan file bersama yang ada dan membuat penyimpanan file bersama baru berdasarkan kuota yang tercantum dalam tabel berikut.

Kuota penyimpanan file yang dipasang untuk setiap cluster

| Jenis penyimpanan bersama file | AWS ParallelCluster penyimpanan terkelola | Penyimpanan eksternal | Total kuota bersih |
|--------------------------------|-------------------------------------------|-----------------------|--------------------|
| Amazon EBS                     | 5                                         | 5                     | 5                  |
| PENYERBUAN                     | 1                                         | 0                     | 1                  |
| Amazon EFS                     | 1                                         | 20                    | 21                 |
| Amazon FSx †                   | 1 FSx for Lustre                          | 20                    | 21                 |

### Note

Tabel kuota ini ditambahkan dalam AWS ParallelCluster versi 3.2.0.

† AWS ParallelCluster hanya mendukung pemasangan Amazon FSx yang ada untuk NetApp ONTAP, Amazon FSx untuk OpenZFS, dan sistem Cache File. Itu tidak mendukung pembuatan FSx baru untuk ONTAP, FSx untuk OpenZFS, dan sistem File Cache.

### Note

Jika Anda menggunakan AWS Batch sebagai penjadwal, FSx for Lustre hanya tersedia di node kepala cluster.

File Cache tidak mendukung AWS Batch penjadwal.



## AWS ParallelCluster sumber daya dan penandaan

dengan AWS ParallelCluster Anda dapat membuat tag untuk melacak dan mengelola AWS ParallelCluster sumber daya. Anda menentukan tag yang Anda inginkan AWS CloudFormation untuk membuat dan menyebarkan ke semua sumber daya cluster di [Bagian Tags](#) dari file konfigurasi cluster. Anda juga dapat menggunakan tag yang AWS ParallelCluster secara otomatis menghasilkan untuk melacak dan mengelola sumber daya Anda.

Saat Anda membuat cluster, cluster dan sumber dayanya ditandai dengan AWS ParallelCluster dan AWSTag sistem didefinisikan dalam bagian ini.

AWS ParallelCluster menerapkan tag ke instance, volume, dan sumber daya cluster. Untuk mengidentifikasi tumpukan cluster, AWS CloudFormation berlaku AWSTag sistem ke instance cluster. Untuk mengidentifikasi template peluncuran EC2 cluster, EC2 menerapkan tag sistem ke instance. Anda dapat menggunakan tag ini untuk melihat dan mengelola AWS ParallelCluster sumber daya.

Anda tidak dapat memodifikasi AWSTag sistem. Untuk menghindari dampak terhadap AWS ParallelCluster fungsionalitas, jangan ubah AWS ParallelCluster tag.

Berikut ini adalah contoh dari AWSTag sistem untuk AWS ParallelCluster sumber daya. Anda tidak dapat memodifikasinya.

```
"aws:cloudformation:stack-name"="clustername"
```

Berikut ini adalah contoh dari AWS ParallelCluster tag diterapkan ke sumber daya. Jangan memodifikasinya.

```
"parallelcluster:cluster-name"="clustername"
```

Anda dapat melihat tag ini di bagian EC2 AWS Management Console.

Lihat tag

1. Arahkan konsol EC2 di <https://console.aws.amazon.com/ec2/>.
2. Untuk melihat semua tag cluster, pilih Tag di panel navigasi.
3. Untuk melihat tag cluster berdasarkan contoh, pilih Contoh di panel navigasi.
4. Pilih contoh cluster.
5. Pilih Mengelola tag tab dalam rincian contoh dan melihat tag.

6. Pilih Penyimpanan tab di detail instance.
7. Pilih ID Volume.
8. Di Volume, pilih volumenya.
9. Pilih Tag tab di detail volume dan lihat tag.

#### AWS ParallelCluster tag instance simpul kepala

| Key                                        | Nilai tanda                                                                                                              |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>parallelcluster:cluster-name</code>  | <i>clustername</i>                                                                                                       |
| Name                                       | HeadNode                                                                                                                 |
| <code>aws:ec2launchtemplate:id</code>      | <i>lt-1234567890abcdef0</i>                                                                                              |
| <code>aws:ec2launchtemplate:version</code> | <i>1</i>                                                                                                                 |
| <code>parallelcluster:node-type</code>     | HeadNode                                                                                                                 |
| <code>aws:cloudformation:stack-name</code> | <i>clustername</i>                                                                                                       |
| <code>aws:cloudformation:logical-id</code> | HeadNode                                                                                                                 |
| <code>aws:cloudformation:stack-id</code>   | <i>arn:aws:cloudformation: <b>region-id</b> :ACCOUNTID :stack/clustername /1234abcd-12ab-12ab-12ab-1234567890abcdef0</i> |
| <code>parallelcluster:version</code>       | <i>3.7.0</i>                                                                                                             |

#### AWS ParallelCluster tag volume akar simpul kepala

| Kunci tanda                               | Nilai tanda        |
|-------------------------------------------|--------------------|
| <code>parallelcluster:cluster-name</code> | <i>clustername</i> |
| <code>parallelcluster:node-type</code>    | HeadNode           |
| <code>parallelcluster:version</code>      | <i>3.7.0</i>       |

## AWS ParallelCluster menghitung tag instance node

| Key                                                | Nilai tanda                  |
|----------------------------------------------------|------------------------------|
| <code>parallelcluster:cluster-name</code>          | <i>clustername</i>           |
| <code>parallelcluster:compute-resource-name</code> | <i>compute-resource-name</i> |
| <code>aws:ec2launchtemplate:id</code>              | <i>lt-1234567890abcdef0</i>  |
| <code>aws:ec2launchtemplate:version</code>         | <i>1</i>                     |
| <code>parallelcluster:node-type</code>             | Compute                      |
| <code>parallelcluster:queue-name</code>            | <i>queue-name</i>            |
| <code>parallelcluster:version</code>               | <i>3.7.0</i>                 |

## AWS ParallelCluster menghitung tag volume akar simpul

| Kunci tanda                                        | Nilai tanda                  |
|----------------------------------------------------|------------------------------|
| <code>parallelcluster:cluster-name</code>          | <i>clustername</i>           |
| <code>parallelcluster:compute-resource-name</code> | <i>compute-resource-name</i> |
| <code>parallelcluster:node-type</code>             | Compute                      |
| <code>parallelcluster:queue-name</code>            | <i>queue-name</i>            |
| <code>parallelcluster:version</code>               | <i>3.7.0</i>                 |

## AWS ParallelCluster Tag UI

| Kunci tanda                     | Nilai tanda |
|---------------------------------|-------------|
| <code>parallelcluster-ui</code> | true        |

# Pemantauan AWS ParallelCluster dan log

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan kinerja AWS ParallelCluster dan solusi AWS lain Anda. AWS menyediakan alat pemantauan berikut untuk memantau AWS ParallelCluster, melaporkan jika ada yang salah, dan mengambil tindakan otomatis jika diperlukan:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS di secara langsung. 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 membuat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru ketika diperlukan. Untuk informasi lebih lanjut, lihat [Panduan Pengguna Amazon CloudWatch](#).
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari instans Amazon EC2, CloudTrail, dan sumber lainnya. Log CloudWatch dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Anda juga dapat mengarsipkan data log Anda dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat [Panduan Pengguna Log CloudWatch Amazon](#).
- AWS CloudTrail merekam panggilan API dan kejadian terkait yang dilakukan oleh atau atas Akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang memanggil AWS, alamat IP sumber yang melakukan panggilan, dan kapan panggilan tersebut terjadi. Untuk mengetahui informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).
- Amazon EventBridge adalah layanan bus peristiwa nirkabel yang membantu menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge mengirimkan aliran data waktu nyata dari aplikasi Anda sendiri, aplikasi SaaS oftware-as-a (SaaS), dan AWS layanan serta merutekan data tersebut ke target seperti Lambda. Hal ini memungkinkan Anda memantau kejadian yang terjadi dalam layanan, dan membangun arsitektur yang didorong kejadian. Untuk informasi lebih lanjut, lihat [Panduan Pengguna Amazon EventBridge](#).

## Topik

- [Integrasi dengan Amazon CloudWatch Logs](#)
- [CloudWatchDasbor Amazon](#)
- [CloudWatchAlarm Amazon untuk metrik klaster](#)

- [AWS ParallelCluster rotasi log yang dikonfigurasi](#)
- [pclusterLog CLI](#)
- [Log keluaran konsol EC2](#)
- [Ambil AWS ParallelCluster UI dan AWS ParallelCluster runtime log](#)
- [Mengambil dan melestarikan log](#)

## Integrasi dengan Amazon CloudWatch Logs

Untuk informasi selengkapnya tentang CloudWatch Log, lihat [Panduan Pengguna Amazon CloudWatch Logs](#). Untuk mengkonfigurasi integrasi CloudWatch Log, lihat [Monitoring](#) bagian. Untuk mempelajari cara menambahkan log kustom ke CloudWatch konfigurasi menggunakan `append-config`, lihat [Beberapa file konfigurasi CloudWatch agen](#) dalam Panduan CloudWatch Pengguna Amazon.

### CloudWatchLog kluster Amazon Logs

Grup log dibuat untuk setiap kluster dengan nama, `/aws/parallelcluster/cluster-name-<timestamp>` (misalnya, `/aws/parallelcluster/testCluster-202202050215`). Setiap log (atau kumpulan log jika path berisi `*`) pada setiap node memiliki aliran log bernama `{hostname}.{instance_id}.{logIdentifier}`. (Misalnya `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`.) Data log dikirim CloudWatch oleh [CloudWatchagen](#), yang berjalan seperti `root` pada semua instance kluster.

CloudWatchDasbor Amazon dibuat ketika kluster dibuat. Dasbor ini memberi Anda kemampuan untuk meninjau log yang disimpan dalam CloudWatch Log. Untuk informasi selengkapnya, lihat [CloudWatchDasbor Amazon](#).

Daftar ini berisi *logIdentifier* dan jalur untuk aliran log yang tersedia untuk platform, penjadwal, dan node.

Log stream tersedia untuk platform, penjadwal, dan node

| Platform | Penjadw | Simpul | Pengaliran Log                                        |
|----------|---------|--------|-------------------------------------------------------|
| amazon   | awsbatc | HeadNc | dcv-otentikator: /var/log/parallelcluster/pcluster_dc |
| centos   | slurm   |        | v_authenticator.log                                   |
| redhat   |         |        |                                                       |

| Platform                             | Penjadw           | Simpul                     | Pengaliran Log                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------|-------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ubuntu                               |                   |                            | dcv-ext-authenticator: /var/log/parallelcluster/parallelcluster_dcv_connect.log<br><br>DCV-agent: /var/log/dcv/agent.*.log<br><br>DCV-XSession: /var/log/dcv/dcv-xsession.*.log<br><br>dcv-server: /var/log/dcv/server.log<br><br>dcv-session-launcher: /var/log/dcv/sessionlauncher.log<br><br>Xdcv: /var/log/dcv/Xdcv.*.log<br><br>cfn-init terkait /var/log/cfn-init.log<br><br>koki klien: /var/log/chef-client.log |
| amazon<br>centos<br>redhat<br>ubuntu | awsbatch<br>slurm | Compute<br>eet<br>HeadNode | awan-init: /var/log/cloud-init.log<br><br>pengawas: /var/log/supervisord.log                                                                                                                                                                                                                                                                                                                                            |
| amazon<br>centos<br>redhat<br>ubuntu | slurm             | Compute<br>eet             | cloud-init-output: /var/log/cloud-init-output.log<br><br>komputemgtd: /var/log/parallelcluster/computemgtd<br><br>slurmd: /var/log/slurmd.log<br><br>slurm_prolog_epilog: /var/log/parallelcluster/slurm_prolog_epilog.log                                                                                                                                                                                              |

| Platform | Penjadw  | Simpul   | Pengaliran Log                                                                      |
|----------|----------|----------|-------------------------------------------------------------------------------------|
| amazon   | slurm    | HeadNode | sssd: /var/log/sssds/sssds.log                                                      |
| centos   |          |          | sssd_domain_default: /var/log/sssds/sssds_default.log                               |
| redhat   |          |          | pam_ssh_key_generator: /var/log/parallelcluster/pam_ssh_key_generator.log           |
| ubuntu   |          |          | clusterstatusmgt: /var/log/parallelcluster/clusterstatusmgt                         |
|          |          |          | clustermgt: /var/log/parallelcluster/clustermgt                                     |
|          |          |          | compute_console_output: /var/log/parallelcluster/compute_console_output             |
|          |          |          | slurm_resume: /var/log/parallelcluster/slurm_resume.log                             |
|          |          |          | slurm_suspend: /var/log/parallelcluster/slurm_suspend.log                           |
|          |          |          | slurmctld: /var/log/slurmctld.log                                                   |
|          |          |          | slurm_fleet_status_manager: /var/log/parallelcluster/slurm_fleet_status_manager.log |
| amazon   | awsbatch | Compute  | sistem-pesan: /var/log/messages                                                     |
| centos   | slurm    | HeadNode |                                                                                     |
| redhat   |          |          |                                                                                     |
| ubuntu   | awsbatch | Compute  | syslog: /var/log/syslog                                                             |
|          | slurm    | HeadNode |                                                                                     |

Pekerjaan di cluster yang menggunakan AWS Batch menyimpan output dari pekerjaan yang mencapai keadaan `RUNNING`, `SUCCEEDED`, atau `FAILED` di CloudWatch Log. Grup log adalah `/aws/batch/job`, dan format nama aliran log adalah `jobDefinitionName/default/ecs_task_id`. Secara default, log ini diatur agar tidak kedaluwarsa, tetapi Anda dapat mengubah periode penyimpanannya. Untuk informasi selengkapnya, lihat [Mengubah penyimpanannya dalam CloudWatch Logs](#) dalam Panduan Pengguna Amazon CloudWatch Logs.

## Amazon CloudWatch Logs membangun log gambar

Grup log dibuat untuk setiap image build kustom dengan nama `/aws/imagebuilder/ParallelClusterImage-<image-id>`. Aliran log unik dengan nama `{pcluster-version}/1` berisi output dari proses image build.

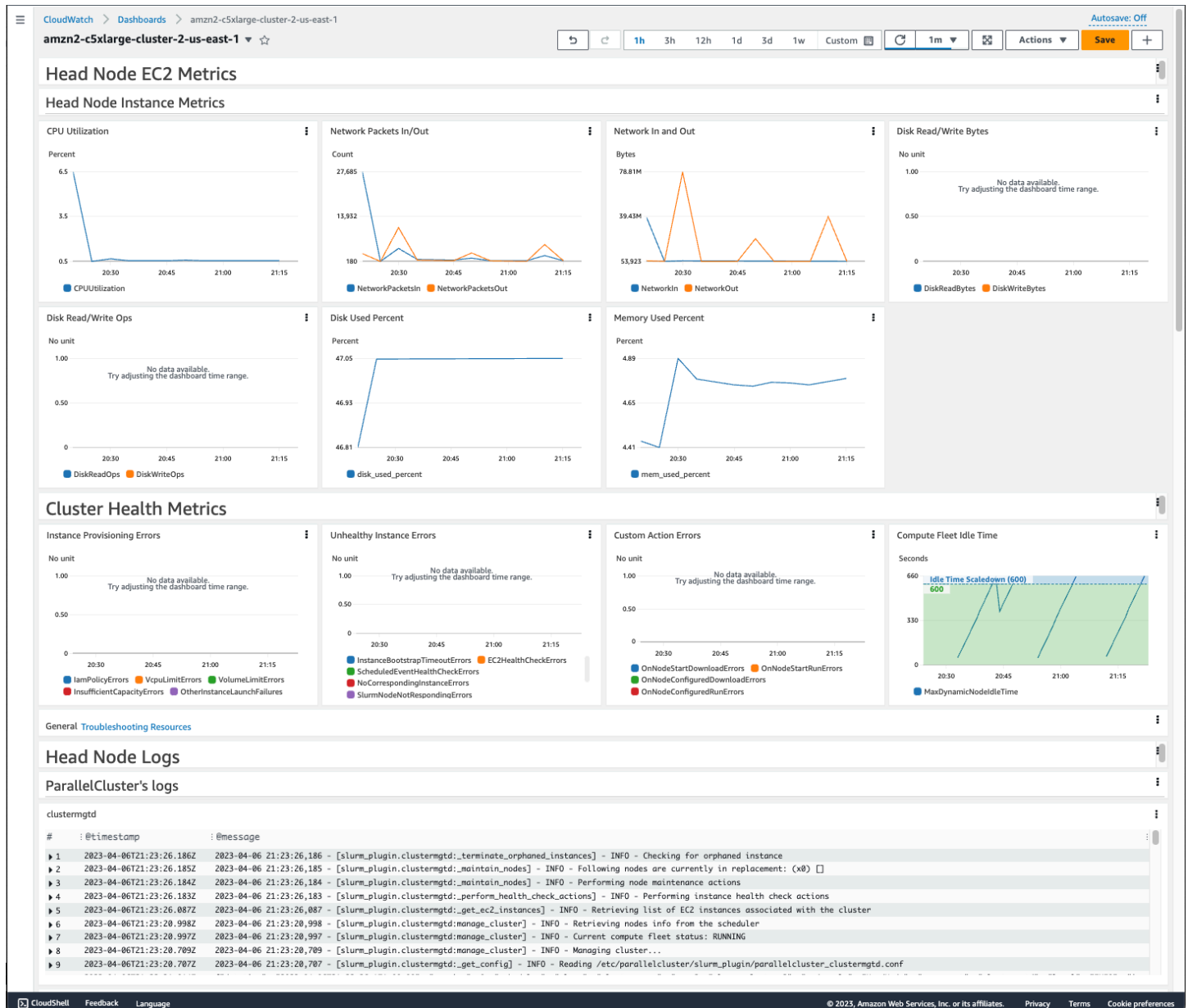
Anda dapat mengakses log dengan menggunakan perintah `pcluster` gambar. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Kustomisasi AMI](#).

## CloudWatchDasbor Amazon

CloudWatchDasbor Amazon dibuat ketika sebuah klaster dibuat. Ini membuatnya lebih mudah untuk memantau node di klaster Anda, dan untuk melihat log yang disimpan di Amazon CloudWatch Logs. Nama dasbor `ClusterName-Region`. `ClusterName` adalah nama cluster Anda dan `Region` adalah cluster berada Wilayah AWS di. Anda dapat mengakses dasbor tersebut di konsol, atau dengan membuka `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`.

Gambar berikut menunjukkan CloudWatch dasbor contoh untuk sebuah klaster.





## Metrik Instans Node Kepala

Bagian pertama dasbor menampilkan grafik metrik EC2 node kepala.

Jika klaster Anda memiliki penyimpanan bersama, bagian selanjutnya menampilkan metrik penyimpanan bersama.

## Metrik Health Cluster

Jika klaster Anda menggunakan Slurm penjadwalan, grafik metrik kesehatan klaster menunjukkan kesalahan node komputasi klaster waktu nyata. Untuk informasi selengkapnya, lihat [Pemecahan](#)

[masalah metrik metrik metrik metrik metrik Kondisi Kondisi](#). Metrik kesehatan kluster ditambahkan ke dasbor yang dimulai dengan AWS ParallelCluster versi 3.6.0.

## Kepala Node Log

Bagian terakhir mencantumkan log simpul kepala yang AWS ParallelCluster dikelompokkan berdasarkan log, log Penjadwal, log integrasi NICE DCV, dan log Sistem.

Untuk informasi selengkapnya tentang CloudWatch dasbor Amazon, lihat [Menggunakan CloudWatch dasbor Amazon](#) di CloudWatchPanduan Pengguna Amazon.

Jika Anda tidak ingin membuat CloudWatch dasbor Amazon, Anda dapat mematikannya dengan menyetel [Monitoring/Dashboards/CloudWatch/Enabled](#) ke false.

### Note

Jika Anda menonaktifkan pembuatan CloudWatch dasbor Amazon, Anda juga menonaktifkan Amazon CloudWatch `disk_used_percent` dan `memory_used_percent` alarm untuk kluster Anda. Untuk informasi selengkapnya, lihat [CloudWatchAlarm Amazon untuk metrik kluster](#).

The `disk_used_percent` dan `memory_used_percent` alarm ditambahkan dimulai dengan AWS ParallelCluster versi 3.6.

## CloudWatchAlarm Amazon untuk metrik kluster

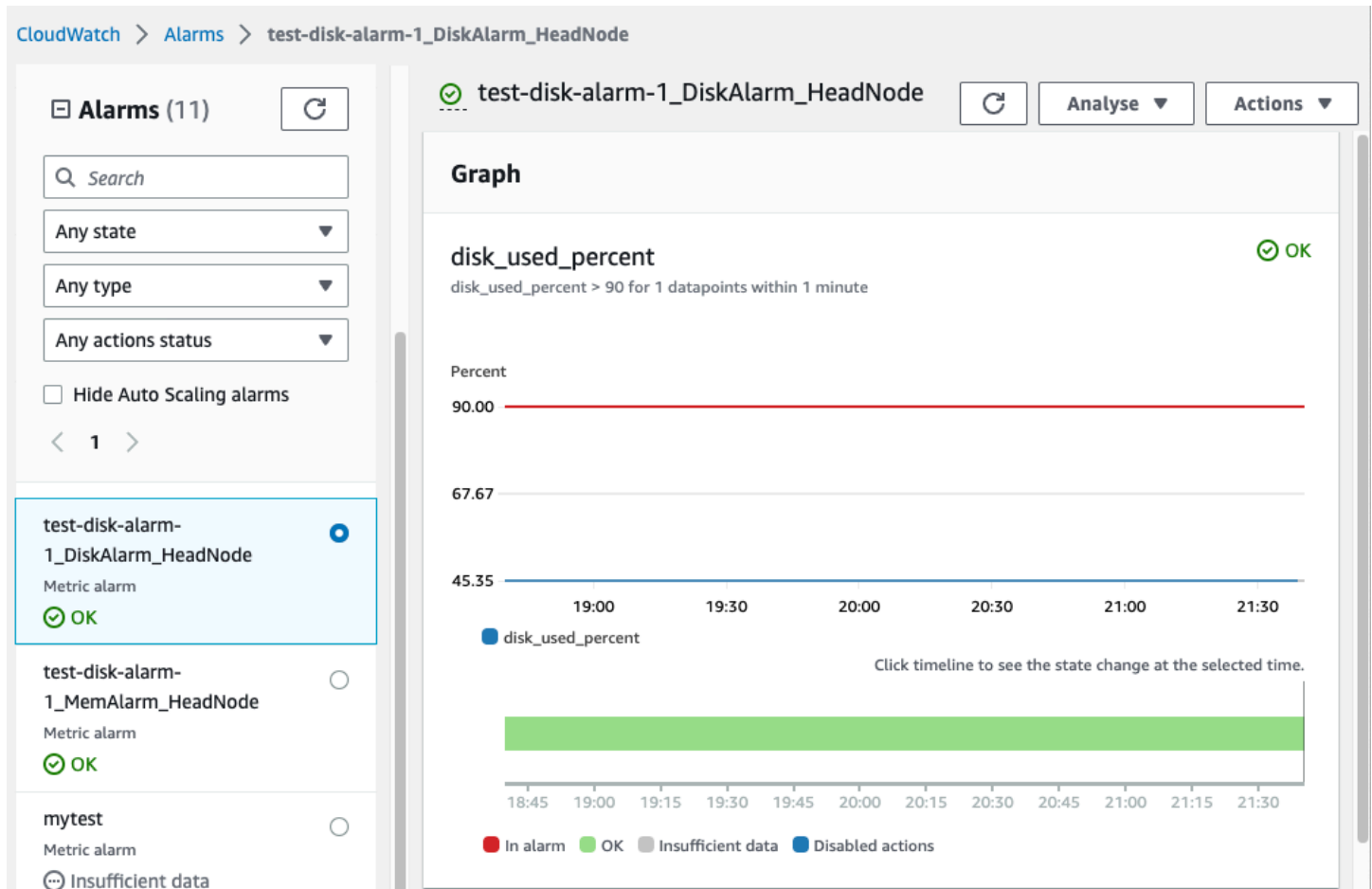
Dimulai dengan AWS ParallelCluster versi 3.6, Anda dapat mengonfigurasi kluster Anda dengan CloudWatch alarm Amazon untuk memantau head node. Satu alarm memonitor volume `rootdisk_used_percent`. Alarm lainnya memonitor `mem_used_percent` metrik. Untuk informasi selengkapnya, lihat [Metrik yang dikumpulkan oleh CloudWatch agen](#) di Panduan CloudWatch Pengguna Amazon.

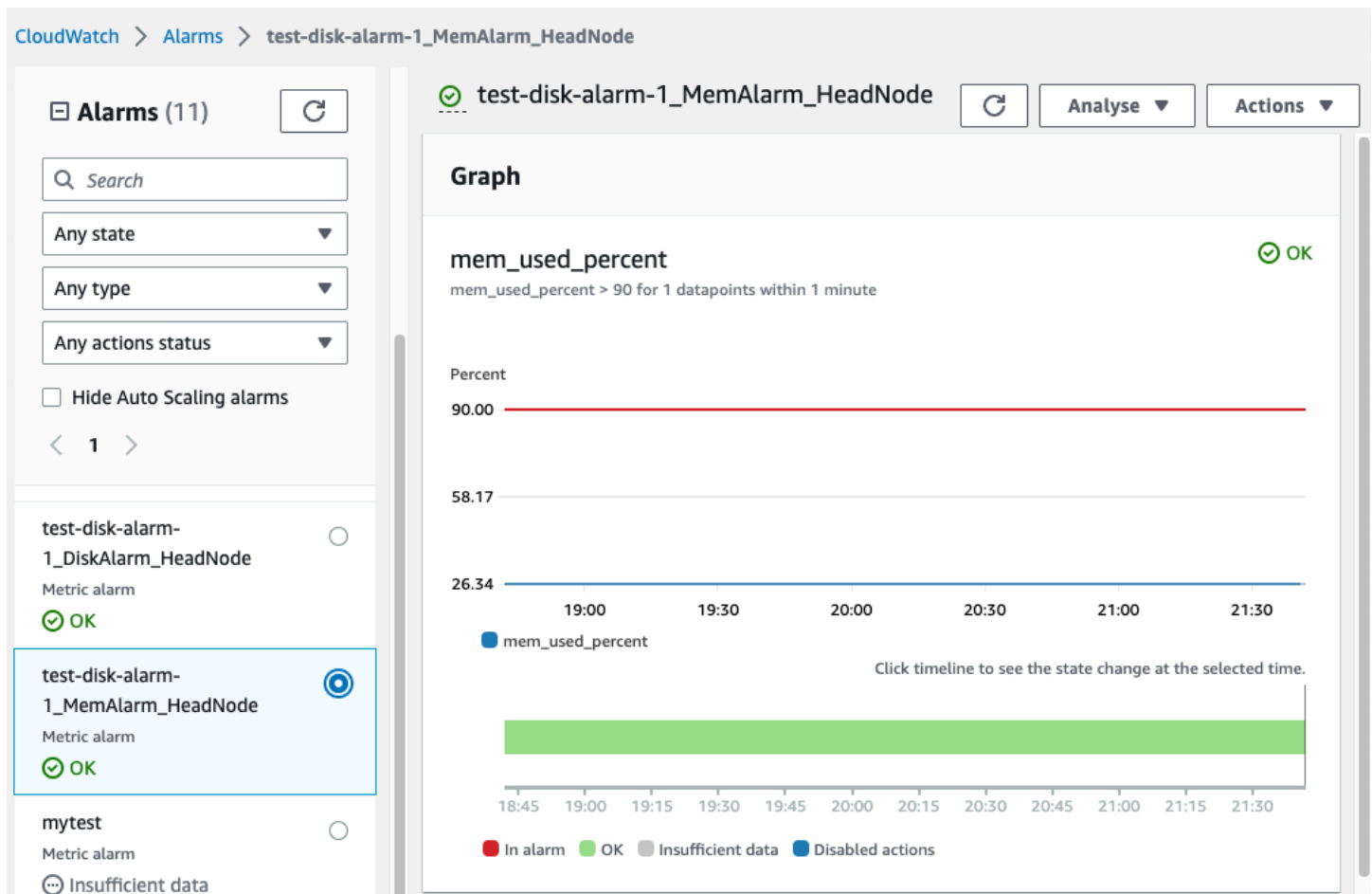
Alarm diberi nama sebagai berikut:

- `cluster-name_DiskAlarm_HeadNode`
- `cluster-name_MemAlarm_HeadNode`

Nama *Kluster* adalah *nama* kluster Anda.

Akses alarm di CloudWatch konsol dengan memilih Alarm di panel navigasi. Gambar berikut menunjukkan alarm penggunaan disk dan alarm penggunaan memori untuk cluster.





Alarm penggunaan disk dalam ALARM keadaan ketika persentase penggunaan disk lebih besar dari 90% untuk 1 titik data, dalam periode waktu 1 menit.

Alarm penggunaan memori dalam ALARM keadaan ketika persentase penggunaan memori lebih besar dari 90% untuk 1 titik data, dalam periode waktu 1 menit.

### Note

AWS ParallelCluster tidak mengkonfigurasi tindakan alarm secara default. Untuk informasi tentang cara mengatur tindakan alarm, seperti mengirim notifikasi, lihat [Tindakan alarm](#). Untuk informasi selengkapnya tentang CloudWatch alarm Amazon, lihat [Menggunakan CloudWatch alarm Amazon](#) di CloudWatch Panduan Pengguna Amazon.

Jika Anda tidak ingin membuat alarm Amazon ini, nonaktifkan CloudWatch alarm tersebut dengan menyetel [Monitoring/DashboardsCloudWatch/Enabled](#) ke false dalam konfigurasi kluster.

Ini juga menonaktifkan pembuatan CloudWatch dasbor Amazon. Untuk informasi selengkapnya, lihat [CloudWatchDasbor Amazon](#).

#### Note

Jika Anda menonaktifkan pembuatan CloudWatch dasbor Amazon, Anda juga menonaktifkan Amazon CloudWatch `disk_used_percent` dan `memory_used_percent` alarm untuk klaster Anda.

## AWS ParallelClusterrotasi log yang dikonfigurasi

TheAWS ParallelClusterkonfigurasi rotasi log terletak di`/etc/logrotate.d/parallelcluster*_log_rotation`berkas. Saat log yang dikonfigurasi berputar, konten log saat ini disimpan dalam satu cadangan dan log yang dikosongkan melanjutkan pencatatan.

Hanya 1 cadangan yang dipertahankan untuk setiap log yang dikonfigurasi.

AWS ParallelClustermengonfigurasi log yang tumbuh cepat untuk diputar ketika ukurannya mencapai 50 MB. Log yang tumbuh cepat terkait dengan penskalaan danSlurm, termasuk`/var/log/parallelcluster/clustermgtd`,`/var/log/parallelcluster/slurm_resume.log`, dan`/var/log/slurmctld.log`.

AWS ParallelClustermengonfigurasi log yang tumbuh lambat untuk diputar ketika ukurannya mencapai 10 MB.

Anda dapat melihat log sebelumnya yang disimpan untuk jumlah hari yang ditentukan dalam konfigurasi cluster[Logs/CloudWatch/RetentionInDays](#)pengaturan dengan CloudFormation logging diaktifkan. Periksa[RetentionInDays](#)pengaturan untuk melihat apakah jumlah hari perlu ditingkatkan untuk kasus penggunaan Anda.

AWS ParallelClustermengkonfigurasi dan memutar log berikut:

Log simpul kepala

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cfn-init.log
/var/log/chef-client.log
/var/log/dcv/server.log
/var/log/dcv/sessionlauncher.log
```

```
/var/log/dcv/agent.*.log
/var/log/dcv/dcv-xsession.*.log
/var/log/dcv/Xdcv.*.log
/var/log/parallelcluster/pam_ssh_key_generator.log
/var/log/parallelcluster/clustermgtd
/var/log/parallelcluster/clusterstatusmgtd
/var/log/parallelcluster/slurm_fleet_status_manager.log
/var/log/parallelcluster/slurm_resume.log
/var/log/parallelcluster/slurm_suspend.log
/var/log/slurmctld.log
/var/log/slurmdbd.log
/var/log/parallelcluster/compute_console_output.log
```

## Hitung log simpul

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log
```

## Log simpul masuk

```
/var/log/cloud-init.log
/var/log/cloud-init.log
/var/log/cloud-init-output.log
/var/log/supervisord.log
/var/log/parallelcluster/pam_ssh_key_generator.log
```

## pclusterLog CLI

pclusterCLI menulis log perintah Anda ke `pcluster.log`. # file di `/home/user/.parallelcluster/`

Untuk setiap perintah, log umumnya menyertakan perintah dengan input, salinan versi CLI API yang digunakan untuk membuat perintah, respons, dan kedua info dan pesan kesalahan. Untuk perintah `create` dan `build`, log juga menyertakan file konfigurasi, operasi validasi file konfigurasi, CloudFormation template, dan perintah `stack`.

Anda dapat menggunakan log ini untuk memverifikasi kesalahan, input, versi, dan perintah pcluster CLI. Mereka juga dapat berfungsi sebagai catatan kapan perintah dibuat.

## Log keluaran konsol EC2

Ketika AWS ParallelCluster mendeteksi bahwa instance node komputasi statis berakhir secara tak terduga, ia mencoba untuk mengambil output konsol EC2 dari instance node yang diakhiri setelah periode waktu berlalu. Dengan cara ini, jika node komputasi tidak dapat berkomunikasi dengan AmazonCloudWatch, informasi pemecahan masalah yang berguna tentang mengapa node dihentikan mungkin masih diambil dari output konsol. Output konsol ini dicatat dalam `/var/log/parallelcluster/compute_console_output` log pada node kepala. Untuk informasi selengkapnya tentang output konsol EC2, lihat output [instans](#) di Panduan Pengguna Amazon EC2 untuk Contoh Linux.

Secara default, AWS ParallelCluster hanya mengambil output konsol dari subset sampel node yang diakhiri. Hal ini mencegah node kepala cluster kewalahan dengan beberapa permintaan output konsol yang disebabkan oleh sejumlah besar terminasi. Secara default, AWS ParallelCluster menunggu 5 menit antara deteksi terminasi dan pengambilan output konsol untuk memberikan waktu EC2 untuk mengambil output konsol akhir dari node.

Anda dapat mengedit ukuran sampel dan menunggu nilai parameter waktu dalam `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` file pada node kepala.

Fitur ini ditambahkan di AWS ParallelCluster versi 3.5.0.

### Parameter output konsol EC2

Anda dapat mengedit nilai-nilai parameter output konsol EC2 berikut dalam `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` file pada node kepala.

#### **compute\_console\_logging\_enabled**

Untuk menonaktifkan koleksi log keluaran konsol, atur `compute_console_logging_enabled` ke `false`. Defaultnya adalah `true`.

Anda dapat memperbarui parameter ini kapan saja, tanpa menghentikan armada komputasi.

#### **compute\_console\_logging\_max\_sample\_size**

`compute_console_logging_max_sample_size` menetapkan jumlah maksimum node komputasi dari yang AWS ParallelCluster mengumpulkan output konsol setiap kali mendeteksi penghentian tak

terduga. Jika nilai ini kurang dari 1, AWS ParallelCluster mengambil output konsol dari semua node diakhiri. Nilai default-nya adalah 1.

Anda dapat memperbarui parameter ini kapan saja, tanpa menghentikan armada komputasi.

### **compute\_console\_wait\_time**

`compute_console_wait_time` mengatur waktu, dalam hitungan detik, yang AWS ParallelCluster menunggu antara mendeteksi kegagalan node dan mengumpulkan output konsol dari node itu. Anda dapat meningkatkan waktu tunggu jika Anda menentukan bahwa EC2 membutuhkan lebih banyak waktu untuk mengumpulkan output akhir dari node diakhiri. Nilai default adalah 300 detik (5 menit).

Anda dapat memperbarui parameter ini kapan saja, tanpa menghentikan armada komputasi.

## Ambil AWS ParallelCluster UI dan AWS ParallelCluster runtime log

Pelajari cara mengambil AWS ParallelCluster UI dan log AWS ParallelCluster runtime untuk pemecahan masalah. Untuk memulai, cari nama AWS ParallelCluster UI dan AWS ParallelCluster tumpukan yang relevan. Gunakan nama tumpukan untuk menemukan grup log instalasi. Untuk menyelesaikannya, ekspor log. Log ini khusus untuk AWS ParallelCluster runtime. Untuk log kluster, lihat [Mengambil dan melestarikan log](#).

### Prasyarat

- AWS CLI terinstal.
- Anda memiliki kredensi untuk menjalankan AWS CLI perintah pada AWS ParallelCluster UI Akun AWS yang aktif.
- Anda dapat mengakses CloudWatch konsol Amazon di Akun AWS tempat AWS ParallelCluster UI aktif.

### Langkah 1: Cari nama tumpukan untuk tumpukan yang relevan

Pada contoh berikut, ganti teks yang disorot merah dengan nilai aktual Anda.

Daftar tumpukan, menggunakan Wilayah AWS tempat Anda menginstal AWS ParallelCluster UI:

```
$ aws cloudformation list-stacks --region aws-region-id
```

Perhatikan nama tumpukan untuk tumpukan berikut:



- Nama tumpukan yang menyebarkan AWS ParallelCluster UI di akun Anda. Anda memasukkan nama ini ketika Anda menginstal AWS ParallelCluster UI; misalnya, `pcluster-ui`.
- AWS ParallelCluster Tumpukan yang diawali dengan nama tumpukan yang Anda masukkan; misalnya, `pcluster-ui-ParallelClusterApi-ABCD1234EFGH`.

## Langkah 2: Temukan grup log

Buat daftar grup log dari tumpukan AWS ParallelCluster UI, seperti yang ditunjukkan dalam contoh berikut:

```
$ aws cloudformation describe-stack-resources \
 --region aws-region-id \
 --stack-name pcluster-ui \
 --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' &&
(LogicalResourceId == 'ApiGatewayAccessLog' || LogicalResourceId ==
'ParallelClusterUILambdaLogGroup')].PhysicalResourceId" \
 --output text
```

Buat daftar grup log dari tumpukan AWS ParallelCluster API, seperti yang ditunjukkan dalam contoh berikut:

```
$ aws cloudformation describe-stack-resources \
 --region aws-region-id \
 --stack-name pcluster-ui-ParallelCluster-Api-ABCD1234EFGH \
 --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' && LogicalResourceId
== 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \
 --output text
```

Perhatikan daftar grup log untuk digunakan di langkah berikutnya.

## Langkah 3: Ekspor log

Gunakan langkah-langkah berikut untuk mengumpulkan dan mengekspor log:

1. Masuk ke AWS Management Console, dan kemudian arahkan ke CloudWatch konsol [Amazon](#) di Akun AWS mana AWS ParallelCluster UI aktif.
2. Pilih Log, Wawasan Log di panel navigasi.
3. Pilih semua grup log yang tercantum di langkah sebelumnya.

4. Pilih rentang waktu, seperti 12 jam.
5. Jalankan kueri berikut:

```
$ fields @timestamp, @message
| sort @timestamp desc
| limit 10000
```

6. Pilih Hasil ekspor, Unduh tabel (JSON).

## Mengambil dan melestarikan log

AWS ParallelCluster membuat metrik EC2 untuk HeadNode dan Compute instans dan penyimpanan. Anda dapat melihat metrik di CloudWatch konsol Dasarsip. AWS ParallelCluster juga menciptakan aliran CloudWatch log klaster dalam grup log. Anda dapat melihat log ini di CloudWatch konsol Dasbor Kustom atau grup Log. Bagian Konfigurasi klaster [Monitoring](#) menjelaskan bagaimana Anda dapat memodifikasi CloudWatch log klaster dan dasbor. Untuk informasi selengkapnya, lihat [Integrasi dengan Amazon CloudWatch Logs](#) dan [CloudWatchDasbor Amazon](#).

Log adalah sumber daya yang berguna untuk memecahkan masalah. Misalnya, jika Anda ingin menghapus klaster yang gagal, mungkin berguna untuk terlebih dahulu membuat arsip log klaster. Ikuti langkah-langkah [catatan arsip](#) untuk membuat arsip.

### Topik

- [Log klaster tidak tersedia di CloudWatch](#)
- [catatan arsip](#)
- [Log yang diawetkan](#)
- [Log simpul yang diakhiri](#)

## Log klaster tidak tersedia di CloudWatch

Jika log klaster tidak tersedia di CloudWatch, periksa untuk memastikan Anda belum menimpa konfigurasi AWS ParallelCluster CloudWatch log saat menambahkan log kustom ke konfigurasi.

Untuk menambahkan log kustom ke CloudWatch konfigurasi, pastikan Anda menambahkan ke konfigurasi daripada mengambil dan menyimpannya. Untuk informasi selengkapnya tentang fetch-config dan append-config, lihat [Beberapa file konfigurasi CloudWatch agen](#) di Panduan CloudWatch Pengguna.

Untuk mengembalikan konfigurasi AWS ParallelCluster CloudWatch log, Anda dapat menjalankan perintah berikut di dalam AWS ParallelCluster node:

```
$ PLATFORM="$(ohai platform | jq -r ".[]")"
LOG_GROUP_NAME="$(cat /etc/chef/dna.json | jq -r ".cluster.log_group_name")"
SCHEDULER="$(cat /etc/chef/dna.json | jq -r ".cluster.scheduler")"
NODE_ROLE="$(cat /etc/chef/dna.json | jq -r ".cluster.node_type")"
CONFIG_DATA_PATH="/usr/local/etc/cloudwatch_agent_config.json"
/opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/python /usr/local/bin/
write_cloudwatch_agent_json.py --platform $PLATFORM --config $CONFIG_DATA_PATH --log-
group $LOG_GROUP_NAME --scheduler $SCHEDULER --node-role $NODE_ROLE
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
-c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json -s
```

## catatan arsip

Anda dapat mengarsipkan log di Amazon S3 atau dalam file lokal (tergantung pada `--output-file` parameternya).

### Note

Menambahkan izin ke kebijakan bucket Amazon S3 untuk memberikan akses CloudWatch. Untuk informasi selengkapnya, lihat [Menetapkan izin pada bucket Amazon S3](#) di Panduan Pengguna CloudWatch Log.

```
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
 --bucket bucketname --bucket-prefix logs
{
 "url": "https://bucketname.s3.eu-west-1.amazonaws.com/export-log/mycluster-
logs-202109071136.tar.gz?..."
}

use the --output-file parameter to save the logs locally
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
 --bucket bucketname --bucket-prefix logs --output-file /tmp/archive.tar.gz
{
 "path": "/tmp/archive.tar.gz"
}
```

Arsip berisi aliran Amazon CloudWatch Logs dan kejadian AWS CloudFormation stack dari node head dan node komputasi selama 14 hari terakhir, kecuali ditentukan secara eksplisit dalam konfigurasi atau dalam parameter untuk perintah. `export-cluster-logs` Waktu yang dibutuhkan untuk perintah untuk menyelesaikan tergantung pada jumlah node dalam cluster dan jumlah log stream yang tersedia di CloudWatch Log. Untuk informasi selengkapnya tentang aliran log yang tersedia, lihat [Integrasi dengan Amazon CloudWatch Logs](#).

## Log yang diawetkan

Mulai dari versi 3.0.0, AWS ParallelCluster mempertahankan CloudWatch Log secara default ketika cluster dihapus. Jika Anda ingin menghapus klaster dan mempertahankan log-nya, pastikan bahwa [Monitoring//LogsCloudWatch/DeletionPolicy](#) tidak diatur ke Delete dalam konfigurasi klaster. Jika tidak, ubah nilai untuk bidang ini `Retain`, dan jalankan `pcluster update-cluster` perintahnya. Kemudian, jalankan `pcluster delete-cluster --cluster-name <cluster_name>` untuk menghapus klaster, tetapi pertahankan grup log yang disimpan di AmazonCloudWatch.

## Log simpul yang diakhiri

Jika node komputasi statis tiba-tiba berakhir dan tidak CloudWatch memiliki log untuk itu, periksa apakah AWS ParallelCluster telah merekam output konsol untuk node komputasi pada node kepala di log. `/var/log/parallelcluster/compute_console_output` Untuk informasi selengkapnya, lihat [Log kunci untuk debugging](#).

Jika `/var/log/parallelcluster/compute_console_output` log tidak tersedia atau tidak berisi output untuk node, gunakan AWS CLI untuk mengambil output konsol dari node gagal. Masuk ke node kepala cluster dan dapatkan node gagal `instance-id` dari `/var/log/parallelcluster/slurm_resume.log` file.

Mengambil output konsol dengan menggunakan perintah berikut dengan: `instance-id`

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Jika node komputasi dinamis berhenti sendiri setelah diluncurkan dan tidak CloudWatch memiliki log untuk itu, kirimkan pekerjaan yang mengaktifkan tindakan penskalaan klaster. Tunggu hingga instance gagal dan ambil log konsol instance.

Masuk ke node kepala cluster dan dapatkan node komputasi `instance-id` dari file. `/var/log/parallelcluster/slurm_resume.log`

Ambil catatan arsip dengan menggunakan perintah berikut:

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Log keluaran konsol dapat membantu Anda men-debug akar penyebab kegagalan node komputasi ketika log node komputasi tidak tersedia.

## AWS CloudFormation sumber daya khusus

Dimulai dengan AWS ParallelCluster versi 3.6.0, Anda dapat menggunakan sumber daya AWS ParallelCluster CloudFormation khusus dalam tumpukan. AWS CloudFormation Sumber daya khusus adalah tumpukan yang AWS ParallelCluster dihosting. Dengan cara ini, Anda dapat menggunakan CloudFormation untuk mengkonfigurasi dan mengelola cluster Anda. Misalnya, Anda dapat mengonfigurasi sumber daya eksternal cluster seperti jaringan, penyimpanan bersama, dan infrastruktur grup keamanan dalam CloudFormation tumpukan. Selanjutnya, Anda dapat mengelola cluster Anda dengan CloudFormation infrastruktur sebagai pipeline kode.

Tambahkan sumber daya AWS ParallelCluster khusus ke CloudFormation template Anda dengan melakukan hal berikut:

1. Tambahkan tumpukan penyedia sumber daya khusus yang dimiliki dan dihosting oleh AWS ParallelCluster.
2. Referensikan tumpukan penyedia di CloudFormation template Anda sebagai sumber daya khusus.

Penyedia sumber daya khusus menumpuk menangani dan merespons CloudFormation permintaan. Misalnya, saat Anda menerapkan CloudFormation tumpukan, Anda juga mengonfigurasi dan membuat cluster. Untuk memperbarui cluster, Anda memperbarui CloudFormation tumpukan Anda. Anda menghapus klaster saat menghapus tumpukan Anda. Untuk informasi selengkapnya tentang CloudFormation sumber daya [kustom, lihat Sumber daya khusus](#) di Panduan AWS CloudFormation Pengguna.

### Warning

CloudFormation tidak mendeteksi penyimpangan sumber daya khusus. Hanya gunakan CloudFormation untuk memperbarui konfigurasi cluster dan menghapus cluster.

Anda dapat menggunakan [pcluster](#) CLI atau [AWS ParallelClusterUI](#) untuk memantau status cluster atau untuk memperbarui armada komputasi, tetapi Anda tidak boleh menggunakannya untuk memperbarui konfigurasi cluster atau untuk menghapus cluster.

### Note

Kami menyarankan Anda menambahkan [perlindungan terminasi](#) ke tumpukan Anda untuk menghindari penghapusan yang tidak disengaja.

## Tumpukan penyedia dihosting oleh AWS ParallelCluster

Tumpukan penyedia sumber daya kustom diformat seperti yang ditunjukkan pada cuplikan CloudFormation templat berikut:

```
PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 Parameters:
 CustomLambdaRole: # (Optional) RoleARN to override default
 AdditionalIamPolicies: # (Optional) comma-separated list of IAM policies to add
 TemplateURL: !Sub
 - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.${AWS::URLSuffix}/
parallelcluster/${Version}/templates/custom_resource/cluster.yaml
 - { Version: 3.7.0 }
```

### Properti:

#### Parameter:

##### CustomLambdaRole (opsional):

Peran khusus dengan izin untuk menjalankan AWS Lambda yang membuat dan mengelola cluster. Secara default, peran menggunakan kebijakan yang sama yang ditentukan secara default dalam [AWS ParallelCluster dokumentasi](#).

##### AdditionalIamPolicies (opsional):

Daftar tambahan IAM Policy Amazon Resource Names (ARN) yang dipisahkan koma untuk ditambahkan ke peran yang digunakan Lambda. Ini hanya digunakan jika a CustomLambdaRole tidak ditentukan dan dapat tetap kosong.

Jika Anda memerlukan kebijakan tambahan untuk node kepala, node komputasi, atau untuk akses ke bucket Amazon S3, tambahkan ke `CustomLambdaRole` properti atau `AdditionalIamPolicy`

Untuk informasi selengkapnya tentang kebijakan default, lihat [AWS Identity and Access Management izin di AWS ParallelCluster](#).

`templateUrl` (wajib):

URL file sumber daya AWS ParallelCluster kustom.

Keluaran:

`ServiceToken`:

Nilai yang dapat digunakan sebagai `ServiceToken` properti sumber daya kustom. Sumber daya khusus `ServiceToken` menentukan tempat AWS CloudFormation mengirim permintaan. Ini adalah masukan yang diperlukan untuk sumber daya cluster yang Anda sertakan dalam AWS CloudFormation template Anda.

`LogGroupArn`:

ARN dari yang dicatat oleh sumber daya CloudWatch LogGroup yang mendasarinya.

`LambdaLayerArn`:

ARN dari lapisan Lambda yang digunakan untuk menjalankan operasi. AWS ParallelCluster

## Sumber daya cluster

Sumber daya CloudFormation cluster diformat seperti yang ditunjukkan dalam cuplikan CloudFormation template berikut:

```
PclusterCluster:
 Type: Custom::PclusterCluster
 Properties:
 ServiceToken: !GetAtt [PclusterClusterProvider , Outputs.ServiceToken]
 ClusterName: !Sub 'c-${AWS::StackName}' # Must be different from StackName
 ClusterConfiguration:
 # Your Cluster Configuration
```

**Properti:****ServiceToken:**

`ServiceTokenOutput` tumpukan AWS ParallelCluster penyedia.

**ClusterName:**

Nama cluster yang akan dibuat dan dikelola. Nama tidak boleh cocok dengan nama CloudFormation tumpukan. Nama tidak dapat diubah setelah cluster dibuat.

**ClusterConfiguration:**

File YAMAL konfigurasi cluster, seperti yang dijelaskan dalam [File konfigurasi cluster](#) file. Namun, Anda dapat menggunakan CloudFormation konstruksi biasa, seperti fungsi [Intrinsik](#).

**DeletionPolicy:**

Mendefinisikan apakah akan menghapus cluster saat tumpukan root dihapus. Nilai default-nya `Delete`.

**Mempertahankan:**

Pertahankan klaster jika sumber daya kustom dihapus.

**Note**

Agar klaster tetap berfungsi, sumber daya yang bergantung pada klaster, seperti penyimpanan dan jaringan, harus memiliki kebijakan penghapusan yang disetel untuk dipertahankan.

**Hapus:**

Hapus cluster jika sumber daya kustom dihapus.

**Fn::GetAtt** mengembalikan nilai:

Fungsi `Fn::GetAtt` intrinsik mengembalikan nilai untuk atribut tertentu dari tipe. Untuk informasi selengkapnya tentang penggunaan `Fn::GetAtt` intrinsik fungsi, lihat [Fn::GetAtt](#).

**ClusterProperties:**

Nilai-nilai dari [pcluster describe-cluster](#) operasi.



### ValidationMessages:

String yang berisi semua pesan validasi yang terjadi selama operasi pembuatan atau pembaruan terakhir.

### logGroupName:

Nama grup log yang digunakan untuk mencatat operasi kluster Lambda. Peristiwa log dipertahankan selama 90 hari dan grup log dipertahankan setelah penghapusan kluster.

### Contoh:Fn::GetAtt:

```
Provide the public IP address of the head node as an output of a stack
Outputs:
 HeadNodeIp:
 Description: The public IP address of the head node
 Value: !GetAtt [PclusterCluster, headNode.publicIpAddress]
```

### Contoh: CloudFormation Template sederhana dan lengkap dengan sumber daya AWS ParallelCluster khusus:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
 HeadNodeSubnet:
 Description: Subnet where the HeadNode will run
 Type: AWS::EC2::Subnet::Id

 ComputeSubnet:
 Description: Subnet where the Compute Nodes will run
 Type: AWS::EC2::Subnet::Id

 KeyName:
 Description: KeyPair to login to the head node
 Type: AWS::EC2::KeyPair::KeyName

Resources:
 PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 TemplateURL: !Sub
```

```
- https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
- { Version: 3.7.0 }
```

**PclusterCluster:**

Type: Custom::PclusterCluster

**Properties:**

ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]

ClusterName: !Sub 'c-\${AWS::StackName}'

**ClusterConfiguration:****Image:**

Os: alinux2

**HeadNode:**

InstanceType: t2.medium

**Networking:**

SubnetId: !Ref HeadNodeSubnet

**Ssh:**

KeyName: !Ref KeyName

**Scheduling:**

Scheduler: slurm

**SlurmQueues:**

- Name: queue0

**ComputeResources:**

- Name: queue0-cr0

InstanceType: t2.micro

**Networking:****SubnetIds:**

- !Ref ComputeSubnet

**Outputs:****HeadNodeIp:**

Description: The Public IP address of the HeadNode

Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]

**ValidationMessages:**

Description: Any warnings from cluster create or update operations.

Value: !GetAtt PclusterCluster.validationMessages

Untuk mempelajari lebih lanjut tentang cara menggunakan sumber daya CloudFormation AWS ParallelCluster kustom, lihat [Membuat cluster dengan AWS CloudFormation](#).

## Operasi cluster

Ketika sumber daya kustom cluster ditambahkan ke CloudFormation tumpukan, CloudFormation dapat melakukan operasi klaster berikut:

- CloudFormation membuat cluster di tumpukan terpisah baru saat menyebarkan tumpukan yang menyertakan sumber daya AWS ParallelCluster khusus.
- Jika Anda memperbarui konfigurasi klaster yang ditentukan dalam tumpukan, sesuai dengan kebijakan pembaruan konfigurasi, CloudFormation perbarui klaster. Penyedia sumber daya AWS ParallelCluster khusus tidak menghentikan armada komputasi sebelum memperbarui klaster. Kami menyarankan Anda menggunakan [QueueUpdateStrategy](#) pengaturan untuk pembaruan cluster. Dengan cara ini, Anda dapat menghindari `pc1uster update-compute-fleet` panggilan eksplisit sebelum dan sesudah pembaruan saat menggunakan sumber daya AWS ParallelCluster khusus.
- Jika Anda menghapus tumpukan, cluster dihapus.

## Memecahkan masalah tumpukan yang menyertakan sumber daya kustom AWS ParallelCluster

Dengan sumber daya AWS ParallelCluster khusus, CloudFormation menyebarkan cluster dari tumpukan baru yang terpisah. Anda dapat memantau pembuatan cluster dengan mengambil langkah-langkah berikut:

1. Arahkan ke CloudFormation dalam AWS Management Console dan pilih Tumpukan di panel navigasi.
2. Pilih tumpukan dengan nama yang Anda tentukan untuk nama cluster.
3. Jika status tumpukan `ROLLBACK_COMPLETE`, kesalahan terjadi selama pembuatan cluster.
4. Pilih Stack details, dan pilih tab Events.
5. Cari Peristiwa pada ID Logis untuk nama yang Anda tetapkan untuk nama cluster. Ini memiliki Status `reason` yang memberikan alasan untuk suatu masalah.
6. Anda juga dapat memilih menu tarik-turun Tumpukan, dan kemudian Dihapus untuk melihat daftar tumpukan yang dihapus. Pilih tumpukan dengan nama cluster dan lihat Acara untuk lebih jelasnya.
7. Untuk melihat output dari penyedia sumber daya khusus yang mengelola cluster, pilih tumpukan dengan Deskripsi "Sumber Daya Kustom AWS ParallelCluster Cluster." Pilih tab Sumber Daya,

temukan sumber daya dengan ID `LogisPcClusterCfnFunctionLogGroup`, dan ikuti tautan yang diberikan. Lihat aliran log yang menampilkan output debug Lambda.

8. Untuk memecahkan masalah cluster, lihat [Memecahkan Masalah AWS ParallelCluster](#)

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) adalah perangkat jaringan yang memiliki kemampuan OS-bypass untuk komunikasi jaringan latensi rendah dengan instans lain pada subnet yang sama. EFA diekspos dengan menggunakan Libfabric, dan dapat digunakan oleh aplikasi menggunakan Messaging Passing Interface (MPI).

Untuk menggunakan EFA dengan AWS ParallelCluster dan Slurm scheduler, set [SlurmQueues/ComputeResources/Efa/Enabled](#) ke `true`.

Untuk melihat daftar instans EC2 yang mendukung EFA, lihat [Jenis instans yang didukung](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Kami menyarankan Anda menjalankan instans berkemampuan EFA di grup penempatan. Dengan cara ini instans diluncurkan ke grup latensi rendah di Availability Zone tunggal. Untuk informasi selengkapnya tentang cara mengkonfigurasi grup penempatan AWS ParallelCluster, lihat [SlurmQueues/Networking/PlacementGroup](#).

Untuk informasi selengkapnya, lihat [Adaptor Kain Elastis](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux dan [Skala beban kerja HPC dengan adaptor kain elastis dan AWS ParallelCluster](#) di Blog AWS Open Source.

### Note

Elastic Fabric Adapter (EFA) tidak didukung di Availability Zone yang berbeda. Untuk informasi selengkapnya, lihat [Penjadwalan/SlurmQueues/Networking/SubnetIds](#).

### Note

Secara default, Ubuntu distribusi mengaktifkan perlindungan `ptrace` (pelacakan proses). `ptrace` perlindungan dinonaktifkan sehingga Libfabric bekerja dengan baik. Untuk informasi selengkapnya, lihat [Menonaktifkan perlindungan ptrace](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

## Aktifkan Intel MPI

Intel MPI tersedia di AWS ParallelCluster AMI untuk `linux2`, `centos7`, `rhel8`, `ubuntu2204`, dan `ubuntu2004` nilai untuk `Image/OS` pengaturan.

### Note

Untuk menggunakan Intel MPI, Anda harus mengakui dan menyetujui persyaratan [Lisensi perangkat lunak Intel yang disederhanakan](#).

Secara default, Open MPI ditempatkan di jalur. Untuk mengaktifkan Intel MPI alih-alih Open MPI, Anda harus terlebih dahulu memuat modul Intel MPI. Kemudian, Anda perlu menginstal versi terbaru dengan menggunakan `module load intelmpi`. Nama pasti modul berubah dengan setiap pembaruan. Untuk melihat modul mana yang tersedia, jalankan `module avail`. Outputnya adalah sebagai berikut.

```
$ module avail
-----/usr/share/Modules/modulefiles

dot modules
libfabric-aws/1.16.0~amzn3.0 null
module-git openmpi/4.1.4
module-info use.own

-----/opt/intel/mpi/2021.6.0/modulefiles

intelmpi
```

Untuk memuat modul, jalankan `module load modulename`. Anda dapat menambahkan ini ke skrip yang digunakan untuk menjalankan `mpirun`.

```
$ module load intelmpi
```

Untuk melihat modul mana yang dimuat, jalankan `module list`.

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

Untuk memverifikasi bahwa Intel MPI diaktifkan, jalankan `mpirun --version`.

```
$ mpirun --version
```

```
Intel(R) MPI Library for Linux* OS, Version 2021.6 Build 20220227 (id: 28877f3f32)
Copyright 2003-2022, Intel Corporation.
```

Setelah modul Intel MPI dimuat, beberapa jalur diubah untuk menggunakan alat Intel MPI. Untuk menjalankan kode yang dikompilasi oleh alat Intel MPI, muat modul Intel MPI terlebih dahulu.

#### Note

Intel MPI tidak kompatibel dengan AWS Contoh berbasis graviton.

#### Note

Sebelum AWS ParallelCluster versi 2.5.0, Intel MPI tidak tersedia di AWS ParallelCluster AMI di Wilayah China (Beijing) dan China (Ningxia).

## API AWS ParallelCluster

Apa itu AWS ParallelCluster API?

AWS ParallelCluster API adalah aplikasi tanpa server yang, setelah diterapkan ke Akun AWS Anda, menyediakan akses terprogram ke AWS ParallelCluster fitur melalui API.

AWS ParallelCluster API didistribusikan sebagai [AWS CloudFormation](#) template mandiri yang menyertakan titik akhir [Amazon API Gateway](#), yang mengekspos AWS ParallelCluster fitur, dan [AWS Lambda](#) fungsi, yang menangani pemrosesan fitur yang dipanggil.

Gambar berikut menunjukkan diagram arsitektur tingkat tinggi dari infrastruktur AWS ParallelCluster API.

## AWS ParallelCluster Dokumentasi API

File spesifikasi OpenAPI yang menjelaskan AWS ParallelCluster API dapat diunduh dari:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

[Mulai dari file spesifikasi OpenAPI, Anda dapat menghasilkan dokumentasi untuk AWS ParallelCluster API dengan menggunakan salah satu dari banyak alat yang tersedia seperti Swagger UI atau Redoc.](#)

## Cara menerapkan API AWS ParallelCluster

Untuk menerapkan AWS ParallelCluster API, Anda harus menjadi Administrator dari Akun AWS

Template yang digunakan untuk menyebarkan API tersedia di URL berikut:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

di mana **<REGION>** adalah Wilayah AWS tempat API perlu diterapkan dan **<VERSION>** merupakan AWS ParallelCluster versinya (misalnya 3.7.0).

AWS Lambda memproses fitur yang dipanggil API dengan menggunakan antarmuka lapisan Lambda dengan [AWS ParallelCluster API pustaka Python](#)

### Warning

Setiap pengguna di Akun AWS, yang memiliki akses istimewa ke AWS Lambda atau layanan Amazon API Gateway, secara otomatis mewarisi izin untuk mengelola AWS ParallelCluster sumber daya API.

## Menyebarkan dengan AWS CLI

Konfigurasi AWS Kredensial yang akan digunakan dengan CLI jika Anda belum melakukannya.

```
$ aws configure
```

Jalankan perintah berikut untuk menerapkan API:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This can be any name
$ VERSION=3.7.0
```

```
$ aws cloudformation create-stack \
 --region ${REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-create-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

Sesuaikan penerapan Anda

Anda dapat menyesuaikan penerapan API dengan menggunakan AWS CloudFormation parameter yang diekspos oleh template. Untuk mengkonfigurasi nilai parameter saat menerapkan melalui CLI, opsi berikut dapat digunakan: `--parameters ParameterKey=KeyName,ParameterValue=Value`

Parameter berikut opsional:

- **Region** - Gunakan `Region` parameter untuk menentukan apakah API mampu mengontrol sumber daya di semua Wilayah AWS (default) atau dalam satu Wilayah AWS. Tetapkan nilai ini Wilayah AWS ke API yang sedang digunakan untuk membatasi akses.
- **ParallelClusterFunctionRole** - Ini mengesampingkan peran IAM yang ditugaskan ke fitur implementasi AWS Lambda fungsi. AWS ParallelCluster Parameter menerima ARN dari peran IAM. Peran tersebut perlu dikonfigurasi untuk memiliki AWS Lambda sebagai kepala sekolah IAM.
- **CustomDomainName, CustomDomainCertificate, CustomDomainHostedZoneId** - Gunakan parameter ini untuk menyetel domain kustom untuk titik akhir Amazon API Gateway. `CustomDomainName` adalah nama domain yang akan digunakan, `CustomDomainCertificate` adalah ARN dari sertifikat AWS terkelola untuk nama domain ini dan `CustomDomainHostedZoneId` merupakan ID dari zona yang dihosting [Amazon Route 53](#) tempat Anda ingin membuat catatan.

#### Warning

Anda dapat mengonfigurasi pengaturan domain khusus untuk menerapkan versi minimum Transport Layer Security (TLS) untuk API. Untuk informasi selengkapnya, lihat [Memilih versi TLS minimum untuk domain kustom di API Gateway](#).

- **EnableIamAdminAccess** - Secara default, operasi AWS ParallelCluster API pemrosesan AWS Lambda fungsi dikonfigurasi dengan peran IAM yang mencegah akses IAM () yang memiliki hak



istimewa. `EnableIamAdminAccess=false` Hal ini membuat API tidak dapat memproses operasi yang memerlukan pembuatan peran atau kebijakan IAM. Karena itu, pembuatan cluster atau gambar khusus hanya berhasil ketika peran IAM disediakan sebagai input sebagai bagian dari konfigurasi sumber daya.

Kapan `EnableIamAdminAccess` disetel ke `true` AWS ParallelCluster API diberikan izin untuk mengelola pembuatan peran IAM yang diperlukan untuk menyebarkan cluster atau membuat AMI kustom.

#### Warning

Menyetel ini ke `true` memberikan hak istimewa admin IAM ke operasi pemrosesan AWS Lambda fungsi. AWS ParallelCluster

Lihat [AWS ParallelCluster kebijakan contoh pengguna untuk mengelola sumber daya IAM](#) untuk detail tambahan tentang fitur yang dapat dibuka kuncinya saat mengaktifkan mode ini.

- `PermissionsBoundaryPolicy`- Parameter opsional ini menerima ARN kebijakan IAM yang ada yang akan ditetapkan sebagai batas izin untuk semua peran IAM yang dibuat oleh infrastruktur API PC dan sebagai syarat pada izin IAM administratif sehingga hanya peran dengan kebijakan ini yang dapat dibuat oleh PC API.

Lihat [PermissionsBoundarymodus](#) untuk rincian tambahan tentang pembatasan yang diberlakukan oleh mode ini.

- `CreateApiUserRole`- Secara default, penerapan AWS ParallelCluster API mencakup pembuatan peran IAM yang ditetapkan sebagai satu-satunya peran yang diizinkan untuk memanggil API. Titik akhir Amazon API Gateway dikonfigurasi dengan kebijakan berbasis sumber daya untuk memberikan izin pemanggilan hanya kepada pengguna yang dibuat. Untuk mengubah ini, atur `CreateApiUserRole=false` lalu berikan akses API ke pengguna IAM yang dipilih. Untuk informasi selengkapnya, lihat [Mengontrol akses untuk menjalankan API](#) di Panduan Pengembang Amazon API Gateway.

#### Warning

Jika `CreateApiUserRole=true` akses ke titik akhir API tidak dibatasi oleh kebijakan sumber daya Amazon API Gateway, semua peran IAM yang memiliki izin tidak dibatasi dapat `execute-api:Invoke` mengakses fitur. AWS ParallelCluster Untuk informasi

selengkapnya, lihat [Mengontrol akses ke API dengan kebijakan sumber daya API Gateway](#) di Panduan Pengembang API Gateway.

### Warning

ParallelClusterApiUserRoleMemiliki izin untuk memanggil semua operasi AWS ParallelCluster API. Untuk membatasi akses ke subset resource API, lihat [Kontrol yang dapat memanggil metode API Gateway API dengan kebijakan IAM](#) di Panduan Pengembang API Gateway.

- IAM RoleAndPolicyPrefix - Parameter opsional ini menerima string maksimal 10 karakter yang akan digunakan sebagai awalan untuk peran dan kebijakan IAM yang dibuat sebagai bagian dari infrastruktur API PC.

## Memperbarui API

Upgrade ke versi yang lebih baru AWS ParallelCluster

Opsi 1: Hapus API yang ada dengan menghapus AWS CloudFormation tumpukan yang sesuai dan menerapkan API baru seperti yang ditunjukkan di atas.

Opsi 2: Perbarui API yang ada dengan menjalankan perintah berikut:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.7.0
$ aws cloudformation update-stack \
 --region ${REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-update-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

## Memanggil API AWS ParallelCluster

Titik akhir AWS ParallelCluster Amazon API Gateway dikonfigurasi dengan [jenis AWS\\_IAM otorisasi](#), dan mengharuskan semua permintaan ditandatangani SigV4 dengan kredensial IAM yang valid (referensi [API](#): membuat permintaan http).

Saat diterapkan dengan pengaturan default, izin pemanggilan API hanya diberikan kepada pengguna IAM default yang dibuat dengan API.

Untuk mengambil ARN pengguna IAM default, jalankan:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiUserRole'].OutputValue" --
output text
```

[Untuk mendapatkan kredensial sementara untuk pengguna IAM default, jalankan perintah STS. AssumeRole](#)

Anda dapat mengambil endpoint AWS ParallelCluster API dengan menjalankan perintah berikut:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiInvokeUrl'].OutputValue" --
output text
```

AWS ParallelClusterAPI dapat dipanggil oleh klien HTTP mana pun yang sesuai dengan spesifikasi OpenAPI yang dapat ditemukan di sini:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

[Permintaan harus ditandatangani SigV4 seperti yang didokumentasikan di sini.](#)

Saat ini, kami tidak menawarkan implementasi klien API resmi apa pun. [Namun, klien API dapat dengan mudah dihasilkan dari model OpenAPI dengan menggunakan OpenAPI Generator.](#) Setelah klien dihasilkan SigV4, penandatanganan perlu ditambahkan jika tidak disediakan di luar kotak.

[Implementasi referensi untuk klien API Python dapat ditemukan di repositori. AWS ParallelCluster](#)

Untuk mempelajari lebih lanjut tentang bagaimana Anda dapat menggunakan klien API Python, lihat tutorialnya. [Menggunakan API AWS ParallelCluster ini](#)

[Untuk menerapkan mekanisme kontrol akses yang lebih canggih, seperti Amazon Cognito atau Lambda Authorizers, atau untuk lebih melindungi API dengan AWS WAF atau kunci API, ikuti dokumentasi Amazon API Gateway.](#)

#### Warning

Pengguna IAM yang diberi wewenang untuk menjalankan AWS ParallelCluster API, secara tidak langsung dapat mengontrol semua AWS sumber daya yang dikelola oleh AWS ParallelCluster di. Akun AWS Ini termasuk pembuatan AWS sumber daya yang tidak dapat dikontrol pengguna secara langsung karena pembatasan kebijakan IAM pengguna. Misalnya, pembuatan AWS ParallelCluster klaster, tergantung pada konfigurasinya, mungkin termasuk penyebaran instans Amazon EC2, Amazon Route 53, sistem file Amazon Elastic File System, sistem file Amazon FSx, peran IAM, dan sumber daya Layanan AWS dari yang lain yang AWS ParallelCluster digunakan oleh pengguna yang mungkin tidak memiliki kontrol langsung atas.

#### Warning

Saat membuat cluster dengan `AdditionalIamPolicies` ditentukan dalam konfigurasi, kebijakan tambahan harus cocok dengan salah satu pola berikut:

```
- !Sub arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/parallelcluster*
- !Sub arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/parallelcluster/*
- !Sub arn:${AWS::Partition}:iam::aws:policy/CloudWatchAgentServerPolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonSSMManagedInstanceCore
- !Sub arn:${AWS::Partition}:iam::aws:policy/AWSBatchFullAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonS3ReadOnlyAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/AWSBatchServiceRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
 AmazonEC2ContainerServiceforEC2Role
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
 AmazonECSTaskExecutionRolePolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
 AmazonEC2SpotFleetTaggingRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/EC2InstanceProfileForImageBuilder
```

```
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
 AWSLambdaBasicExecutionRole
```

Jika Anda memerlukan kebijakan tambahan lainnya, Anda dapat melakukan salah satu hal berikut:

- Edit `DefaultParallelClusterIamAdminPolicy` di:

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

Tambahkan kebijakan di `ArnLike/iam:PolicyARN` bagian.

- Hilangkan kebijakan yang ditentukan `AdditionalIamPolicies` dalam file konfigurasi dan tambahkan kebijakan secara manual ke Peran AWS ParallelCluster Instance yang dibuat dalam klaster.

## Mengakses log dan metrik API

Log API dipublikasikan ke Amazon CloudWatch dengan retensi 30 hari. Untuk mengambil LogGroup nama yang terkait dengan penerapan API, jalankan perintah berikut:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --
 stack-name ${API_STACK_NAME} --query "Stacks[0].Outputs[?
 OutputKey=='ParallelClusterLambdaLogGroup'].OutputValue" --output text
```

Metrik Lambda, log, dan log [AWS X-Ray](#) pelacakan juga dapat diakses melalui konsol Lambda. Untuk mengambil ARN dari fungsi Lambda yang terkait dengan penerapan API, jalankan perintah berikut:

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterLambdaArn'].OutputValue" --
 output text
```

## Connect ke node kepala melalui NICE DCV

NICE DCV adalah teknologi visualisasi jarak jauh yang memungkinkan pengguna terhubung dengan aman ke aplikasi 3D grafis yang dihosting di server berperforma tinggi jarak jauh. Untuk informasi lebih lanjut, lihat [NICE DCV](#).

Perangkat lunak NICE DCV secara otomatis diinstal pada node kepala dan dapat diaktifkan dengan menggunakan `Dcv` bagian dari `HeadNode` konfigurasi.

```
HeadNode:
 Dcv:
 Enabled: true
```

Dengan cara ini, AWS ParallelCluster set/home/<DEFAULT\_AMI\_USER> di node kepala ke [folder penyimpanan server DCV](#). Untuk informasi lebih lanjut tentang parameter konfigurasi NICE DCV, lihat [HeadNode/Dcv](#). Untuk terhubung ke sesi NICE DCV, gunakan `pcluster dcv-connect` perintah.

## Sertifikat HTTPS NICE DCV

NICE DCV secara otomatis menghasilkan sertifikat yang ditandatangani sendiri untuk mengamankan lalu lintas antara klien NICE DCV dan server NICE DCV.

Untuk mengganti sertifikat NICE DCV yang ditandatangani sendiri default dengan sertifikat lain, sambungkan dulu ke node kepala. Kemudian, salin sertifikat dan kunci ke /etc/dcv folder sebelum menjalankan `pcluster dcv-connect` perintah.

Untuk informasi selengkapnya, lihat [Mengubah sertifikat TLS](#) di Panduan Administrator NICE DCV.

## Lisensi NICE DCV

Server NICE DCV tidak memerlukan server lisensi saat berjalan di instans Amazon EC2. Namun, server NICE DCV harus terhubung secara berkala ke bucket Amazon S3 untuk menentukan apakah lisensi yang valid tersedia.

AWS ParallelCluster secara otomatis menambahkan izin yang diperlukan ke kebijakan IAM simpul kepala. Saat menggunakan Kebijakan Instans IAM kustom, gunakan izin yang dijelaskan dalam [NICE DCV di Amazon EC2](#) di Panduan Administrator NICE DCV.

Untuk tips pemecahan masalah, lihat [Memecahkan masalah di NICE DCV](#).

# Menggunakan `pcluster update-cluster`

Dalam AWS ParallelCluster 3.x, [pcluster update-cluster](#) menganalisis pengaturan yang digunakan untuk membuat cluster saat ini dan pengaturan dalam file konfigurasi untuk masalah. Jika ada masalah yang ditemukan, mereka dilaporkan, dan langkah-langkah yang harus diambil untuk memperbaiki masalah ditampilkan. Misalnya, jika komputasi [InstanceType](#) diubah, armada komputasi harus dihentikan sebelum pembaruan dapat dilanjutkan. Masalah ini dilaporkan ketika ditemukan. Jika tidak ada masalah pemblokiran yang ditemukan, proses pembaruan dimulai dan perubahan dilaporkan.

Anda dapat menggunakan `pcluster update-cluster --dryrun` option untuk melihat perubahan sebelum dijalankan. Untuk informasi selengkapnya, lihat [pcluster update-cluster contoh](#).

Untuk panduan pemecahan masalah, lihat. [Memecahkan Masalah AWS ParallelCluster](#)

## Kebijakan Pembaruan: definisi

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

Setelah mengubah pengaturan ini, cluster dapat diperbarui menggunakan [pcluster update-cluster](#).

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

Setelah mengubah pengaturan ini, cluster tidak dapat diperbarui. Anda harus mengembalikan pengaturan untuk cluster asli dan membuat cluster baru dengan pengaturan yang diperbarui. Anda dapat menghapus cluster asli di kemudian hari. Untuk membuat cluster baru, gunakan [pcluster create-cluster](#). Untuk menghapus cluster asli, gunakan [pcluster delete-cluster](#).

Kebijakan pembaruan: Pengaturan ini tidak dianalisis selama pembaruan.

Pengaturan ini dapat diubah, dan cluster diperbarui menggunakan [pcluster update-cluster](#).

Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.

Pengaturan ini tidak dapat diubah saat armada komputasi ada. Perubahan harus dikembalikan atau armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)). Setelah armada komputasi dihentikan, Anda dapat memperbarui cluster ([pcluster update-](#)

[cluster](#)) untuk mengaktifkan perubahan. Misalnya, jika Anda menggunakan Slurm penjadwal dengan `SlurmQueues// - ComputeResourcesName/MinCount > 0`, armada komputasi dimulai.

Kebijakan pembaruan: Armada komputasi dan node login harus dihentikan agar pengaturan ini diubah untuk pembaruan.

Pengaturan ini tidak dapat diubah saat armada komputasi ada atau jika node login sedang digunakan. Entah perubahan harus dikembalikan atau armada komputasi dan node login harus dihentikan (Armada komputasi dapat dihentikan menggunakan). [pcluster update-compute-fleet](#) Setelah armada komputasi dan node login dihentikan, Anda dapat memperbarui cluster ([pcluster update-cluster](#)) untuk mengaktifkan perubahan.

Kebijakan pembaruan: Pengaturan ini tidak dapat dikurangi selama pembaruan.


Pengaturan ini dapat diubah, tetapi tidak dapat dikurangi. Jika pengaturan ini harus dikurangi, Anda harus mengembalikan pengaturan untuk cluster asli dan membuat cluster baru dengan pengaturan yang diperbarui. Anda dapat menghapus cluster asli di kemudian hari. Untuk membuat cluster baru, gunakan [pcluster create-cluster](#). Untuk menghapus cluster asli, gunakan [pcluster delete-cluster](#).

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan. Jika Anda memaksa pembaruan, nilai baru akan diabaikan dan nilai lama akan digunakan.

Setelah mengubah pengaturan ini, cluster tidak dapat diperbarui. Anda harus mengembalikan pengaturan untuk cluster asli dan membuat cluster baru dengan pengaturan yang diperbarui. Anda dapat menghapus cluster asli di kemudian hari. Untuk membuat cluster baru, gunakan [pcluster create-cluster](#). Untuk menghapus cluster asli, gunakan [pcluster delete-cluster](#).

Kebijakan pembaruan: Armada komputasi harus dihentikan atau [QueueUpdateStrategy](#) harus disetel agar pengaturan ini diubah untuk pembaruan.

Pengaturan ini dapat diubah. Entah armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)) atau [QueueUpdateStrategy](#) harus diatur. Setelah armada komputasi dihentikan atau [QueueUpdateStrategy](#) disetel, Anda dapat memperbarui cluster ([pcluster update-cluster](#)) untuk mengaktifkan perubahan.

 Note

Kebijakan pembaruan ini didukung dimulai dengan AWS ParallelCluster versi 3.2.0.




Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, nilai baru dapat ditambahkan selama pembaruan atau armada komputasi harus dihentikan saat menghapus nilai yang ada.

Nilai baru untuk pengaturan ini dapat ditambahkan selama pembaruan. Setelah menambahkan nilai baru ke daftar, cluster dapat diperbarui menggunakan ([pcluster update-cluster](#)).

Untuk menghapus nilai yang ada dari daftar, armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)).

Misalnya, jika Anda menggunakan Slurm penjadwal dan menambahkan jenis instans baru ke [Instances/ InstanceType](#), Anda dapat memperbarui klaster tanpa menghentikan armada komputasi. [Untuk menghapus jenis instans yang ada dari Instances/ InstanceType, armada komputasi harus dihentikan terlebih dahulu \(menggunakan pcluster\). update-compute-fleet](#)

 Note

Kebijakan pembaruan ini didukung dimulai dengan AWS ParallelCluster versi 3.2.0.

Kebijakan pembaruan: Mengurangi ukuran antrian memerlukan armada komputasi dihentikan atau [QueueUpdateStrategy](#) harus disetel ke TERMINATE agar pengaturan ini diubah untuk pembaruan.

Pengaturan ini dapat diubah, tetapi jika perubahan akan mengurangi ukuran antrian, armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)) atau [QueueUpdateStrategy](#) harus diatur ke TERMINATE. Setelah armada komputasi dihentikan atau [QueueUpdateStrategy](#) disetel ke TERMINATE, Anda dapat memperbarui cluster ([pcluster update-cluster](#)) untuk mengaktifkan perubahan.

Set TERMINATE saat mengubah ukuran kapasitas cluster, hanya akan mengakhiri node dari bagian belakang daftar node, dan akan meninggalkan semua node lain dari partisi yang sama.

Misalnya, jika kapasitas awal cluster adalah  $MinCount = 5$  dan  $MaxCount = 10$ , node adalah  $st-[1-5]; dy-[1-5]$ . Saat mengubah ukuran cluster ke  $MinCount = 3$  dan  $MaxCount = 5$ , kapasitas cluster baru akan disusun oleh  $node st-[1-3]; dy-[1-2]$ , yang tidak akan disentuh selama pembaruan. Hanya node  $st-[4-5]; dy-[3-5]$  yang akan dihentikan selama pembaruan.

Perubahan berikut didukung dan tidak memerlukan armada komputasi dihentikan atau [QueueUpdateStrategy](#) disetel ke TERMINATE:

- Yang baru [SlurmQueue](#) ditambahkan

- Yang baru [ComputeResource](#) ditambahkan
- [MaxCount](#) meningkat
- [MinCount](#) meningkat dan [MaxCount](#) meningkat setidaknya jumlah yang sama

Catatan: Kebijakan pembaruan ini didukung dimulai dengan AWS ParallelCluster versi 3.9.0.

Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, armada komputasi harus dihentikan atau [QueueUpdateStrategy](#) harus disetel untuk menambahkan nilai baru; armada komputasi harus dihentikan saat menghapus nilai yang ada.

Nilai baru untuk pengaturan ini dapat ditambahkan selama pembaruan. Entah armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)) atau [QueueUpdateStrategy](#) harus diatur. Setelah armada komputasi dihentikan atau [QueueUpdateStrategy](#) disetel, Anda dapat memperbarui cluster ([pcluster update-cluster](#)) untuk mengaktifkan perubahan.

Untuk menghapus nilai yang ada dari daftar, armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)).

#### Note

Kebijakan pembaruan ini didukung dimulai dengan AWS ParallelCluster versi 3.3.0.

Kebijakan pembaruan: Semua node komputasi harus dihentikan untuk penghapusan grup penempatan terkelola. Armada komputasi harus dihentikan atau [QueueUpdateStrategy](#) harus disetel agar pengaturan ini diubah untuk pembaruan.

Armada komputasi harus dihentikan (menggunakan [pcluster update-compute-fleet](#)) untuk menghapus grup penempatan terkelola. Jika Anda menjalankan pembaruan klaster untuk menghapus grup penempatan terkelola sebelum menghentikan armada komputasi, pesan konfigurasi yang tidak valid akan ditampilkan dan pembaruan tidak dilanjutkan. Menghentikan armada komputasi menjamin tidak ada instance yang berjalan.

## **pcluster update-cluster** contoh

Pengaturan ini dapat diubah, tetapi jika perubahan akan mengurangi ukuran antrian, armada komputasi harus dihentikan (menggunakan `pcluster update-compute-fleet`) atau [QueueUpdateStrategy](#) harus diatur ke TERMINATE. Setelah armada komputasi dihentikan atau

[QueueUpdateStrategy](#) disetel ke TERMINATE, Anda dapat memperbarui cluster ([pcluster update-cluster](#)) untuk mengaktifkan perubahan.

- Contoh ini menunjukkan pembaruan dengan beberapa perubahan yang diizinkan dan pembaruan dimulai secara langsung.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
 "cluster": {
 "clusterName": cluster_name,
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "cloudformationStackArn": stack_arn,
 "region": "us-east-1",
 "version": "3.7.0",
 "clusterStatus": "UPDATE_IN_PROGRESS"
 },
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 }
]
}
```

- Contoh ini menunjukkan pembaruan dryrun dengan beberapa perubahan yang diizinkan. Dryrun berguna untuk melaporkan set perubahan tanpa memulai pembaruan.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1 --dryrun true
{
 "message": "Request would have succeeded, but DryRun flag is set.",
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 }
]
}
```

```

 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 }
]
}

```

- Contoh ini menunjukkan pembaruan dengan beberapa perubahan yang memblokir pembaruan.

```

$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
 "message": "Update failure",
 "updateValidationErrors": [
 {
 "parameter": "HeadNode.Ssh.KeyName",
 "requestedValue": "mykey_2",
 "message": "Update actions are not currently supported for the 'KeyName'
parameter. Restore 'KeyName' value to 'jenkinsjun'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
 "currentValue": "mykey_1"
 },
 {
 "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
 "requestedValue": "c4.xlarge",
 "message": "All compute nodes must be stopped. Stop the compute fleet with the
pcluster update-compute-fleet command",
 "currentValue": "t2.micro"
 },
 {
 "parameter": "SharedStorage[ebs1].MountDir",
 "requestedValue": "/my/very/very/long/shared_dir",
 "message": "Update actions are not currently supported for the 'MountDir'
parameter. Restore 'MountDir' value to '/shared'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
 "currentValue": "/shared"
 }
],
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"

```

```
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 },
 {
 "parameter": "HeadNode.Ssh.KeyName",
 "requestedValue": "mykey_2",
 "currentValue": "mykey_1"
 },
 {
 "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
 "requestedValue": "c4.xlarge",
 "currentValue": "t2.micro"
 },
 {
 "parameter": "SharedStorage[ebs1].MountDir",
 "requestedValue": "/my/very/very/long/shared_dir",
 "currentValue": "/shared"
 }
]
}
```

## AWS ParallelCluster Kustomisasi AMI

Ada skenario di mana membangun AMI khusus AWS ParallelCluster diperlukan. Bagian ini mencakup apa yang harus dipertimbangkan saat membuat AWS ParallelCluster AMI khusus.

Anda dapat membuat AWS ParallelCluster AMI kustom menggunakan salah satu metode berikut:

1. Buat [file konfigurasi image build](#), lalu gunakan `pcluster` CLI untuk membuat image dengan EC2 Image Builder. Proses ini otomatis, berulang, dan mendukung pemantauan. Untuk informasi selengkapnya, lihat perintah [pcluster](#) gambar.
2. Buat instance dari AWS ParallelCluster AMI, lalu masuk ke dalamnya dan buat modifikasi manual. Terakhir, gunakan Amazon EC2 untuk membuat AMI baru dari instans yang dimodifikasi. Proses ini membutuhkan waktu lebih sedikit. Namun, ini tidak otomatis atau berulang, dan tidak mendukung penggunaan perintah pemantauan gambar `pcluster` CLI.

Untuk informasi lebih lanjut tentang metode ini, lihat [Membangun kebiasaan AWS ParallelCluster AMI](#).

## AWS ParallelCluster Pertimbangan kustomisasi AMI

Tidak peduli bagaimana Anda membuat gambar kustom Anda, kami sarankan Anda melakukan tes validasi awal dan menyertakan ketentuan untuk memantau status gambar yang sedang dibuat.

Untuk membuat AMI kustom menggunakan `pcluster`, Anda membuat [file konfigurasi gambar build](#) dengan `ImageBagian Build` dan yang digunakan [EC2 Image Builder untuk membuat gambar kustom Anda](#). `BuildBagian` ini menentukan apa yang dibutuhkan Image Builder untuk membangun gambar. Ini termasuk [ParentImage](#) (gambar dasar), dan [Components](#). [Komponen Image Builder](#) mendefinisikan urutan langkah yang diperlukan untuk menyesuaikan instance sebelum gambar dibuat atau untuk menguji instance yang diluncurkan oleh gambar yang dibuat. Untuk contoh AWS ParallelCluster komponen, lihat [AMI kustom](#). `ImageBagian` ini menentukan properti gambar.

Saat dipanggil dari `pcluster build-image` untuk membuat gambar khusus, Image Builder menggunakan konfigurasi image build dengan AWS ParallelCluster buku masak untuk bootstrap AWS ParallelCluster pada gambar Anda. [ParentImage](#) Image Builder mengunduh komponen, menjalankan fase build dan validasi, membuat AMI, meluncurkan instance dari AMI, dan menjalankan pengujian. Ketika proses selesai, Image Builder kemudian menghasilkan gambar baru atau pesan berhenti.

### Lakukan tes validasi komponen kustom

Sebelum Anda menyertakan komponen Image Builder dalam konfigurasi, uji dan validasi menggunakan salah satu metode berikut. Karena proses Image Builder dapat memakan waktu hingga 1 jam, kami sarankan Anda menguji komponen terlebih dahulu. Ini dapat menghemat banyak waktu.

#### Kasus skrip

Uji skrip dalam instance yang sedang berjalan, di luar proses image build, dan verifikasi bahwa skrip keluar dengan kode keluar 0.

#### Kasus Nama Sumber Daya Amazon (ARN)

Uji dokumen komponen dalam instance yang sedang berjalan, di luar proses image build. Untuk daftar persyaratan, lihat [Manajer komponen](#) di Panduan Pengguna Image Builder.

Setelah validasi berhasil, tambahkan komponen ke konfigurasi image build Anda

Setelah Anda memverifikasi bahwa komponen kustom berfungsi, tambahkan ke [file konfigurasi gambar Build](#).

## Pantau proses Image Builder dengan **pcluster** perintah untuk membantu debugging

### [describe-image](#)

Gunakan perintah ini untuk memantau status image build.

### [list-image-log-streams](#)

Gunakan perintah ini untuk mendapatkan ID aliran log yang dapat Anda gunakan untuk mengambil peristiwa log. [get-image-log-events](#)

### [get-image-log-events](#)

Gunakan perintah ini untuk mendapatkan aliran log dari peristiwa proses image build.

Misalnya, Anda dapat mengekor peristiwa build image menggunakan perintah berikut.

```
$ watch -n 1 'pcluster get-image-log-events -i <image-id> \
 --log-stream-name/1 <pcluster-version> \
 --query "events[*].message" | tail -n 50'
```

### [get-image-stack-events](#)

Gunakan perintah ini untuk mengambil peristiwa tumpukan gambar untuk tumpukan yang dibuat Image Builder.

### [export-image-logs](#)

Gunakan perintah ini simpan log gambar.

Untuk informasi selengkapnya tentang AWS ParallelCluster log dan Amazon CloudWatch, lihat [Amazon CloudWatch Logs membangun log gambar](#) dan [CloudWatchDasbor Amazon](#).

## Pertimbangan lainnya

### AWS ParallelCluster Rilis baru dan AMI khusus

Jika Anda membuat dan menggunakan AMI kustom, Anda harus mengulangi langkah-langkah yang Anda gunakan untuk membuat AMI kustom Anda dengan setiap AWS ParallelCluster rilis baru.

## Tindakan bootstrap kustom

Tinjau [Tindakan bootstrap kustom](#) bagian untuk menentukan apakah modifikasi yang ingin Anda buat dapat ditulis dan didukung dengan AWS ParallelCluster rilis future.

## Menggunakan AMI kustom

Anda dapat menentukan AMI kustom dalam konfigurasi cluster di [CustomAmi](#) bagian [ImageScheduling/CustomAmi](#) dan [SlurmQueues/Name/- Image//](#).

Untuk memecahkan masalah peringatan validasi AMI kustom, lihat. [Memecahkan masalah AMI khusus](#)

## Meluncurkan instans dengan OCR (Cadangan Kapasitas Sesuai Permintaan)

Dengan CR [Kapasitas Sesuai Permintaan](#) (ODCR), Anda dapat mencadangkan kapasitas komputasi komputasi Amazon EC2 komputasi komputasi hingga kapan pun. Dengan cara ini, Anda dapat membuat dan mengelola dan mengelola dan mengelola dan mengelola dan mengelola secara independen dari akun penagihan yang ditawarkan [Savings Plans](#) atau [Instans regional](#).

Anda dapat membuat konfigurasi open atau CR Kapasitas targeted Sesuai Permintaan. ODCR terbuka mencakup setiap contoh yang cocok dengan atribut ODCR. Atribut ini adalah jenis instance, platform, dan Availability Zone. Anda harus secara eksplisit mendefinisikan ODCR Target dalam konfigurasi klaster. Untuk menentukan apakah ODCR adalah open atau targeted, jalankan perintah AWS CLI EC2 [describe-capacity-reservation](#).

Anda juga dapat membuat ODCR dalam grup penempatan klaster yang disebut [reservasi kapasitas sesuai permintaan grup penempatan klaster](#) (CPG ODCR).

Beberapa ODCR dapat dikelompokkan dalam grup sumber daya. Ini dapat didefinisikan dalam file konfigurasi cluster. Untuk informasi lebih lanjut tentang resource group, lihat [Apa itu resource group?](#) dalam Panduan Pengguna Resource Groups dan Tag.

## Menggunakan ODCR dengan AWS ParallelCluster

AWS ParallelCluster mendukung ODCR terbuka. Saat menggunakan OCR terbuka, Anda tidak perlu menentukan apa pun. AWS ParallelCluster Instans dipilih secara otomatis untuk klaster. Anda dapat menentukan grup penempatan yang ada atau AWS ParallelCluster membuat grup baru untuk Anda.



## ODCR dalam konfigurasi cluster

Dimulai dengan AWS ParallelCluster versi 3.3.0, Anda dapat menentukan ODCR dalam file konfigurasi cluster, tanpa perlu menentukan penggantian run-instance EC2.

Anda mulai dengan membuat [reservasi kapasitas](#) dan [grup sumber daya](#) menggunakan metode yang dijelaskan dalam dokumentasi tertaut untuk masing-masing. Anda harus menggunakan AWS CLI metode untuk membuat grup reservasi kapasitas. Jika Anda menggunakannya AWS Management Console, Anda hanya dapat membuat grup sumber daya berbasis Tag atau Stack. Grup sumber daya berbasis tag dan berbasis Stack tidak didukung oleh AWS ParallelCluster atau AWS CLI saat meluncurkan instans dengan reservasi kapasitas.

Setelah pemesanan kapasitas dan grup sumber daya dibuat, tentukan di

[SlurmQueues/CapacityReservationTarget](#) atau [SlurmQueuesComputeResources/CapacityReser](#)

yang ditunjukkan pada konfigurasi cluster contoh berikut. Ganti *nilai* yang menyala tinggi dalam warna merah dengan nilai valid Anda.

```
Image:
 Os: os
HeadNode:
 InstanceType: head_node_instance
 Networking:
 SubnetId: public_subnet_id
 Ssh:
 KeyName: key_name
Scheduling:
 Scheduler: scheduler
SlurmQueues:
 - Name: queue1
 Networking:
 SubnetIds:
 - private_subnet_id
 ComputeResources:
 - Name: cr1
 Instances:
 - InstanceType: instance
 MaxCount: max_queue_size
 MinCount: max_queue_size
 Efa:
 Enabled: true
 CapacityReservationTarget:
```

CapacityReservationResourceGroupArn: *capacity\_reservation\_arn*

## USANG/ TIDAK DIREKOMENDASIKAN - ODCR bertarget dengan penggantian instans EC2

### Warning

- Dimulai dengan AWS ParallelCluster versi 3.3.0, kami tidak merekomendasikan metode ini. Bagian ini tetap sebagai referensi untuk implementasi menggunakan versi sebelumnya.
- Metode ini tidak kompatibel dengan Beberapa jenis contoh alokasi dengan Slurm.

Support untuk `targeted` ODCR ditambahkan di AWS ParallelCluster 3.1.1. Dalam rilis ini, mekanisme diperkenalkan yang menimpa `RunInstances` parameter EC2 dan meneruskan informasi tentang reservasi yang akan digunakan untuk setiap sumber daya komputasi yang dikonfigurasi. AWS ParallelCluster Mekanisme ini kompatibel dengan `targeted` ODCR. Namun, ketika Anda menggunakan `targeted` ODCR, Anda harus menentukan konfigurasi `run-instances` override. ODCR yang ditargetkan harus didefinisikan secara eksplisit dalam perintah EC2. AWS CLI [run-instances](#) Untuk menentukan apakah ODCR adalah open atau `targeted` menjalankan perintah AWS CLI EC2. [describe-capacity-reservation](#)

Beberapa ODCR dapat dikelompokkan dalam grup sumber daya. Ini dapat digunakan dalam `run-instance` override untuk menargetkan beberapa ODCR pada saat yang bersamaan.

Jika Anda menggunakan `targeted` ODCR, Anda dapat menentukan grup penempatan. Namun, Anda juga perlu menentukan konfigurasi `run-instances` override.

Misalkan AWS membuat `targeted` ODCR untuk Anda atau Anda memiliki serangkaian Instans Cadangan tertentu. Kemudian, Anda tidak dapat menentukan grup penempatan. Aturan yang dikonfigurasi AWS mungkin bertentangan dengan setelan grup penempatan. Jadi, jika grup penempatan diperlukan untuk aplikasi Anda, gunakan [CPG ODCR](#). Dalam kedua kasus, Anda juga harus menentukan konfigurasi `run-instances` override.

Jika Anda menggunakan CPG ODCR, Anda harus menentukan konfigurasi `run-instances` penggantian dan Anda harus menentukan grup penempatan yang sama dalam konfigurasi kluster.

## Menggunakan Instans cadangan dengan Instans AWS ParallelCluster

Instans cadangan [berbeda](#) dengan Reservasi Kapasitas (ODCR). Ada [2 jenis](#) instance cadangan. Instans Cadangan Regional tidak memiliki kapasitas cadangan. Instans zonal mencadangkan kapasitas di Availability Zone yang ditentukan.

Jika Anda memiliki Instans Cadangan Regional, tidak ada reservasi kapasitas dan Anda bisa mendapatkan Kesalahan Kapasitas Tidak Cukup. Jika Anda memiliki Instans Cadangan zona, Anda memiliki reservasi kapasitas, tetapi tidak ada parameter `run-instances` API yang dapat Anda gunakan untuk menentukannya.

Instans cadangan didukung oleh AWS ParallelCluster versi apa pun. Anda tidak harus menentukan apa pun AWS ParallelCluster dan instans secara otomatis dipilih.

Saat menggunakan Instans Cadangan zona, Anda dapat menghindari potensi Kesalahan Kapasitas Tidak Cukup dengan menghilangkan spesifikasi grup penempatan dalam konfigurasi kluster.

USANG/TIDAK DIREKOMENDASIKAN - Menggunakan **RunInstances** kustomisasi di AWS ParallelCluster 3 untuk CR Kapasitas **targeted** Sesuai Permintaan (ODCR)

### Warning

- Dimulai dengan AWS ParallelCluster versi 3.3.0, kami tidak merekomendasikan metode ini. Bagian ini tetap sebagai referensi untuk implementasi menggunakan versi sebelumnya.
- Metode ini tidak kompatibel dengan Beberapa jenis contoh alokasi dengan Slurm.

Anda dapat mengganti `RunInstances` parameter EC2 untuk setiap sumber daya komputasi yang dikonfigurasi dalam antrian kluster. Untuk melakukannya, buat `/opt/slurm/etc/pcluster/run_instances_overrides.json` file di head node cluster dengan konten cuplikan kode berikut:

- `${queue_name}` adalah nama antrian yang ingin Anda terapkan.
- `${compute_resource_name}` adalah sumber komputasi yang ingin Anda terapkan.
- `${overrides}` adalah objek JSON arbitrer yang berisi daftar `RunInstances` penggantian yang akan digunakan untuk kombinasi spesifik antrian dan tipe instance. Sintaks menimpa perlu mengikuti spesifikasi yang sama yang didokumentasikan dalam panggilan boto3 [run\\_instances](#).

```
{
```

```

 "${queue_name}": {
 "${compute_resource_name}": {
 ${overrides}
 },
 ...
 },
 ...
 }
}

```

Misalnya, JSON berikut mengkonfigurasi grup ODCR yang akan digunakan `group_arn` untuk `p4d.24xlarge` instance yang dikonfigurasi di `dan.my-queue my-compute-resource`

```

{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationResourceGroupArn": "group_arn"
 }
 }
 }
 }
}
}

```

Setelah file JSON ini dibuat, AWS ParallelCluster daemon yang bertanggung jawab untuk penskalaan kluster secara otomatis menggunakan konfigurasi penggantian misalnya diluncurkan. Untuk mengonfirmasi bahwa parameter yang ditentukan sedang digunakan untuk penyediaan instance, lihat file log berikut:

- `/var/log/parallelcluster/clustermgtd`(untuk kapasitas statis)
- `/var/log/parallelcluster/slurm_resume.log`(untuk kapasitas dinamis)

Jika parameternya benar, Anda akan menemukan entri log yang berisi berikut ini:

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

## USANG/ TIDAK DIREKOMENDASIKAN - Buat klaster dengan CR Kapasitas **targeted** Sesuai Permintaan (ODCR)

### Warning

- Dimulai dengan AWS ParallelCluster versi 3.3.0, kami tidak merekomendasikan metode ini. Bagian ini tetap sebagai referensi untuk implementasi menggunakan versi sebelumnya.
- Metode ini tidak kompatibel dengan [Beberapa alokasi tipe instans dengan Slurm](#).

1. Buat grup sumber daya, untuk mengelompokkan kapasitas.

```
$ aws resource-groups create-group --name EC2CRGroup \
 --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

### Note

Grup sumber daya tidak mendukung sumber daya yang dibagikan oleh akun lain. Jika target ODCR dibagikan oleh akun lain, Anda tidak perlu membuat grup sumber daya. Gunakan CapacityReservationId alih-alih grup sumber daya pada langkah 3.

```
#!/bin/bash
set -e

Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationId": "cr-abcdef01234567890"
 }
 }
 }
 }
}
```

EOF

Tambahkan reservasi kapasitas ke grup sumber daya. Setiap kali Anda membuat ODCR baru, tambahkan ke Reservasi Grup. Ganti *ACCOUNT\_ID* dengan ID akun Anda, *PLACEHOLDER\_CAPACITY\_RESERVATION* dengan ID reservasi kapasitas Anda, dan *REGION\_ID* dengan Wilayah AWS ID Anda (misalnya, us-timur-1).

```
$ aws resource-groups group-resources --region REGION_ID --group EC2CRGroup \
 --resource-arns arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-
 reservation/PLACEHOLDER_CAPACITY_RESERVATION
```

Buat dokumen kebijakan di komputer lokal Anda. Ganti *ACCOUNT\_ID* dengan ID akun Anda dan *REGION\_ID* dengan Wilayah AWS ID Anda (misalnya, us-timur-1).

```
cat > policy.json << EOF
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "RunInstancesInCapacityReservation",
 "Effect": "Allow",
 "Action": "ec2:RunInstances",
 "Resource": [
 "arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-reservation/*",
 "arn:aws:resource-groups:REGION_ID:ACCOUNT_ID:group/*"
]
 }
]
}
EOF
```

2. Buat kebijakan IAM tentang Akun AWS penggunaan file json yang Anda buat.

```
$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-
 document file://policy.json
```

3. Buat skrip penginstalan posting berikut secara lokal pada instance dan beri nama. **postinstall.sh**

Ganti **ACCOUNT\_ID** dengan Akun AWS ID Anda, dan **REGION\_ID** dengan Wilayah AWS ID Anda (misalnya, us-timur-1).

```
#!/bin/bash
set -e

Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:REGION_ID:ACCOUNT_ID:group/EC2CRGroup"
 }
 }
 }
 }
}
EOF
```

Unggah file ke keranjang Amazon S3. Ganti **S3\_NAME\_BUCKET** dengan *nama bucket* S3 spesifik Anda.

```
$ aws s3 mb s3://S3_NAME_BUCKET
aws s3 cp postinstall.sh s3://S3_NAME_BUCKET/postinstall.sh
```

4. Buat konfigurasi kluster lokal, ganti placeholder dengan nilai Anda sendiri.

```
Region: REGION_ID
Image:
 Os: alinux2
HeadNode:
 InstanceType: c5.2xlarge
 Ssh:
 KeyName: YOUR_SSH_KEY
 Iam:
 S3Access:
 - BucketName: S3_NAME_BUCKET
 AdditionalIamPolicies:
```

```

- Policy: arn:aws:iam::ACCOUNT_ID:policy/RunInstancesCapacityReservation
This post-install script is executed after the node is configured.
It is used to install scripts at boot time and specific configurations
In the script below we are overriding the calls to RunInstance to force
the provisioning of our my-queue partition to go through
the On-Demand Capacity Reservation
CustomActions:
 OnNodeConfigured:
 Script: s3://S3_NAME_BUCKET/postinstall.sh
Networking:
 SubnetId: YOUR_PUBLIC_SUBNET_IN_TARGET_AZ

Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: my-queue
 ComputeResources:
 - MinCount: 0
 MaxCount: 100
 InstanceType: p4d.24xlarge
 Name: my-compute-resource
 Efa:
 Enabled: true
 Networking:
 ## PlacementGroup:
 ## Enabled: true ## Keep PG disabled if using targeted ODCR
 SubnetIds:
 - YOUR_PRIVATE_SUBNET_IN_TARGET_AZ

```

## 5. Buat cluster.

Gunakan perintah berikut ini untuk membuat kluster. Ganti *cluster-config.yaml* dengan nama file konfigurasi Anda, *cluster-dl* dengan nama kluster Anda, dan *REGION\_ID* dengan *ID* Wilayah Anda (misalnya, us-east-1).

```

$ pcluster create-cluster --cluster-configuration cluster-config.yaml --cluster-name cluster-dl --region REGION_ID

```

Setelah cluster dibuat, skrip pasca-instal berjalan di simpul kepala. Script membuat `run_instances_overrides.json` file dan menimpa panggilan untuk memaksa penyediaan partisi RunInstances untuk pergi melalui On-Demand Capacity Reservation.



AWS ParallelClusterDaemon yang bertanggung jawab untuk penskalaan kluster secara otomatis menggunakan konfigurasi ini untuk instance baru yang diluncurkan. Untuk mengonfirmasi bahwa parameter yang ditentukan digunakan untuk instance penyediaan, Anda dapat melihat file log berikut:

- `/var/log/parallelcluster/clustermgtd`(untuk kapasitas statis - [MinCount](#) > 0)
- `/var/log/parallelcluster/slurm_resume.log`(untuk kapasitas dinamis)

Jika parameternya benar, Anda akan menemukan entri log berisi yang berikut ini.

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

## Memperbarui **RunInstances** menimpa

Anda dapat memperbarui konfigurasi JSON yang dihasilkan kapan saja tanpa menghentikan armada komputasi. Setelah perubahan diterapkan, semua instance baru diluncurkan dengan konfigurasi yang diperbarui. Jika Anda perlu menerapkan konfigurasi yang diperbarui untuk menjalankan node, daur ulang node dengan memaksa penghentian instance dan menunggu AWS ParallelCluster untuk mengganti node tersebut. Anda dapat melakukan ini dengan mengakhiri instance dari konsol EC2 atau AWS CLI, atau dengan mengatur node Slurm dalam keadaan atau. DOWN DRAIN

Gunakan perintah berikut untuk mengatur Slurm node ke DOWN atau DRAIN.

```
$ scontrol update nodename=my-queue-dy-my-compute-resource-1 state=down
reason=your_reason
scontrol update nodename=my-queue-dy-my-compute-resource-1 state=drain
reason=your_reason
```

## Penambahan AMI dan penggantian instans EC2

Untuk memastikan bahwa semua node komputasi kluster yang diluncurkan secara dinamis berperilaku secara konsisten, AWS ParallelCluster menonaktifkan pembaruan OS otomatis instans kluster. Selain itu, serangkaian AWS ParallelCluster AMI khusus dibuat untuk setiap versi AWS ParallelCluster dan CLI yang terkait. Kumpulan AMI khusus ini tetap tidak berubah dan hanya didukung oleh AWS ParallelCluster versi yang mereka buat. AWS ParallelCluster AMI untuk versi yang dirilis tidak diperbarui.

Namun, karena masalah keamanan yang muncul, pelanggan mungkin ingin menambahkan tambalan ke AMI ini dan kemudian memperbarui kluster mereka dengan AMI yang ditambal. Ini sejalan dengan [Model Tanggung Jawab AWS ParallelCluster Bersama](#).

Untuk melihat kumpulan AWS ParallelCluster AMI tertentu yang didukung oleh versi AWS ParallelCluster CLI yang sedang Anda gunakan, jalankan:

```
$ pcluster version
$ pcluster list-official-images
```

Node AWS ParallelCluster kepala adalah contoh statis dan Anda dapat memperbaruinya secara manual. Restart dan reboot node kepala sepenuhnya didukung dimulai dengan AWS ParallelCluster versi 3.0.0.

Jika instans Anda memiliki penyimpanan instans sementara, Anda harus ingat untuk menyimpan data penyimpanan instans sebelum pembaruan manual. Untuk informasi selengkapnya, lihat konfigurasi [HeadNodeLocalStorage//EphemeralVolume](#) kluster dan [jenis Instans dengan volume penyimpanan instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Node komputasi adalah contoh fana. Secara default Anda hanya dapat mengaksesnya dari node kepala. Dimulai dengan AWS ParallelCluster versi 3.0.0, Anda dapat memperbarui AMI yang terkait dengan instance komputasi dengan memodifikasi [CustomAmi](#) parameter [SchedulingSlurmQueuesImage](#)////dan menjalankan `pcluster update-cluster` perintah, setelah menghentikan armada komputasi dengan `pcluster update-compute-fleet`:

```
$ pcluster update-compute-fleet-status --status STOP_REQUESTED
```

Dimungkinkan untuk mengotomatiskan pembuatan AMI kustom yang diperbarui untuk node komputasi dengan menggunakan salah satu metode berikut:

- Gunakan `pcluster build-image` perintah dengan [Build](#)/yang diperbarui [ParentImage](#).
- Jalankan build dengan [Build/UpdateOsPackages/Enabled](#):true.

## Pembaruan atau penggantian instans simpul kepala

Dalam beberapa keadaan, Anda mungkin diminta untuk me-restart atau reboot node kepala. Misalnya, ini diperlukan ketika Anda memperbarui OS secara manual, atau ketika ada [AWSInstance pensiun yang dijadwalkan](#) yang memaksakan instans head node restart.

Jika instans Anda tidak memiliki drive yang singkat, Anda dapat menghentikannya lagi kapan saja. Dalam kasus pensiun yang dijadwalkan, memulai instance berhenti bermigrasi untuk menggunakan perangkat keras baru.

Demikian pula, Anda dapat menghentikan dan memulai instance secara manual yang tidak memiliki toko instance. Untuk kasus ini dan untuk kasus kasus lain tanpa volume sementara, lanjutkan [Menghentikan dan memulai node kepala klaster](#).

Jika instans Anda memiliki drive sementara dan telah dihentikan, data yang ada dalam penyimpanan produk hilang. Anda dapat menentukan apakah jenis instance yang digunakan untuk node head memiliki instance store dari tabel yang ditemukan di [volume toko Instance](#).

## Simpan data dari drive fana

Dimulai dengan AWS ParallelCluster versi 3.0.0, head node restart dan reboot sepenuhnya didukung untuk setiap jenis instance. Namun, jika instance memiliki drive fana, datanya hilang. Ikuti langkah selanjutnya untuk menyimpan data Anda sebelum head node restart atau reboot.

Untuk memeriksa apakah Anda memiliki data yang perlu dilestarikan, lihat konten di [MountDir](#) folder [EphemeralVolume](#)/(secara/scratch default).

Anda dapat mentransfer data ke volume root atau sistem penyimpanan bersama yang dilampirkan ke klaster, seperti Amazon FSx, Amazon EFS, atau Amazon EBS. Perhatikan bahwa transfer data ke penyimpanan jarak jauh dapat dikenakan biaya tambahan.

Setelah menyimpan data, lanjutkan [Menghentikan dan memulai node kepala klaster](#).

## Menghentikan dan memulai node kepala klaster

1. Verifikasi tidak ada pekerjaan yang berjalan di klaster.

Saat menggunakan Slurm penjadwal:

- Jika `batch--no-requeue` opsi tidak ditentukan, pekerjaan yang berjalan akan diminta.
- Jika `--no-requeue` opsi ditentukan, menjalankan pekerjaan gagal.

2. Meminta penghentian armada komputasi klaster:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
 "status": "STOP_REQUESTED",
```

```
...
}
```

3. Tunggu hingga status armada komputasi adalah `STOPPED`:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
 "status": "STOPPED",
 ...
}
```

4. Untuk update manual dengan OS reboot atau contoh restart, Anda dapat menggunakan AWS Management Console atau AWS CLI. Berikut ini adalah contoh penggunaan AWS CLI.

```
Retrieve head node instance id
$ pcluster describe-cluster --cluster-name cluster-name --status STOP_REQUESTED
{
 "headNode": {
 "instanceId": "i-1234567890abcdef0",
 ...
 },
 ...
}
stop and start the instance
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
 "StoppingInstances": [
 {
 "CurrentState": {
 "Name": "stopping"
 ...
 },
 "InstanceId": "i-1234567890abcdef0",
 "PreviousState": {
 "Name": "running"
 ...
 }
 }
]
}
$ aws ec2 start-instances --instance-ids 1234567890abcdef0
{
 "StartingInstances": [
```

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

## 5. Mulai armada komputasi cluster:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
 "status": "START_REQUESTED",
 ...
}
```

## Sistem Operasi)

AWS ParallelCluster mendukung Amazon Linux 2, CentOS 7, Ubuntu 22.04, Ubuntu 2004, Red Hat Enterprise Linux 8 (RHEL8), Rocky 8, Red Hat Enterprise Linux 9 (RHEL9), Rocky 9. AWS ParallelCluster menawarkan AMI pra-bangun untuk sistem operasi tertentu, untuk detail lebih lanjut tentang AMI yang disediakan oleh AWS ParallelCluster referensi. [Bagian Image](#)

## Pertimbangan Sistem Operasi

### Ubuntu 22.04

Ubuntu 2204 membutuhkan kunci yang lebih aman untuk ssh dan tidak mendukung kunci RSA secara default. Harap buat kunci ed25519 dan gunakan itu untuk pembuatan cluster.

Ubuntu 2204 tidak dapat diperbarui ke kernel terbaru karena tidak ada klien Fsx untuk kernel itu.

### RHEL 8

RedHat Enterprise Linux 8.7 (rhel8) ditambahkan mulai versi 3.6.0. AWS ParallelCluster Jika Anda mengonfigurasi klaster untuk menggunakan rhel8, biaya sesuai permintaan untuk jenis instans apa pun lebih tinggi daripada saat Anda mengonfigurasi klaster untuk menggunakan sistem operasi lain yang didukung.

Untuk informasi selengkapnya tentang harga, lihat [Harga Sesuai Permintaan](#) dan [Bagaimana Red Hat Enterprise Linux di Amazon EC2 ditawarkan](#) dan diberi harga? .

## Berbatu 8

AWS ParallelCluster 3.8.0 mendukung Rocky Linux 8, tetapi AMI Rocky Linux 8 yang sudah dibuat sebelumnya (untuk arsitektur x86 dan ARM) tidak tersedia. AWS ParallelCluster 3.8.0 mendukung pembuatan cluster dengan Rocky Linux 8 menggunakan AMI khusus menggunakan properti.

[CustomAmi](#) Untuk informasi lebih lanjut tentang membangun AMI kustom, lihat [AWS ParallelCluster Kustomisasi AMI](#).

[Untuk membangun AMI kustom Anda dari basis Rocky Linux 8 AMI, Anda dapat mempertimbangkan untuk berlangganan AMI Rocky Linux 8 yang tersedia di Marketplace. AWS](#) Pastikan untuk meninjau harga dan biaya berlangganan untuk Rocky Linux 8 AMI di AWS Marketplace. Atau Anda juga dapat menggunakan AMI [Rocky Linux 8 resmi sebagai AMI](#) dasar Anda.

## Centos7

[Gdrcopy](#) telah menghapus Centos7 dari matriks dukungan OS mereka. Ini berarti bahwa gdrcopy 2.3.1 adalah versi terbaru yang mendukung OS ini. Anda harus menyematkan versi NVIDIA dan gdrcopy untuk Centos7, karena versi driver open source NVIDIA terbaru (OpenRM yaitu 535.129.03+) tidak kompatibel dengan versi gdrcopy ini. Mulai ParallelCluster 3.8.0 dan seterusnya, Centos7 AMI resmi kami akan dirilis dengan gdrcopy 2.3.1 dan driver NVIDIA 535.129.03.

## Rocky9

AWS ParallelCluster 3.9.0 mendukung Rocky Linux 9, tetapi AMI Rocky Linux 9 yang sudah dibuat sebelumnya (untuk arsitektur x86 dan ARM) tidak tersedia. AWS ParallelCluster 3.9.0 mendukung pembuatan cluster dengan Rocky Linux 9 menggunakan AMI khusus menggunakan properti.

[CustomAmi](#) Untuk informasi selengkapnya tentang membuat AMI kustom, lihat [kustomisasi AWS ParallelCluster AMI](#). Untuk membangun AMI kustom Anda dari basis Rocky Linux 9 AMI, Anda juga dapat menggunakan AMI [Rocky Linux 9 resmi sebagai AMI](#) dasar Anda. Kustom Rocky Linux 9 AMI build mungkin gagal jika AMI dasar tidak memiliki kernel terbaru. Untuk memutakhirkan kernel sebelum membangun AMI:

- [Luncurkan instance menggunakan id AMI rocky9 dari sini: https://rockylinux.org/cloud-images/](https://rockylinux.org/cloud-images/)
- ssh ke dalam instance dan jalankan perintah berikut: `sudo yum -y update`
- Buat gambar dari instance untuk digunakan sebagai ParentImage

# Referensi untuk AWS ParallelCluster

AWS ParallelCluster Versi saat ini: 3.7.0.

Topik

- [AWS ParallelClusterPerintah CLI](#)
- [File konfigurasi](#)
- [AWS ParallelClusterReferensi API](#)
- [AWS ParallelClusterAPI pustaka Python](#)

## AWS ParallelClusterPerintah CLI

`pcluster` adalah perintah AWS ParallelCluster CLI utama. Anda gunakan `pcluster` untuk meluncurkan dan mengelola kluster HPC di AWS Cloud dan untuk membuat dan mengelola gambar AMI kustom.

`pcluster3-config-converter` digunakan untuk mengkonversi konfigurasi cluster dalam AWS ParallelCluster versi 2 format ke AWS ParallelCluster versi 3 format.

```
pcluster [-h] (build-image | configure |
 create-cluster | dcv-connect |
 delete-cluster | delete-cluster-instances | delete-image |
 describe-cluster | describe-cluster-instances |
 describe-compute-fleet | describe-image |
 export-cluster-logs | export-image-logs |
 get-cluster-log-events | get-cluster-stack-events |
 get-image-log-events | get-image-stack-events |
 list-cluster-log-streams | list-clusters |
 list-images | list-image-log-streams | list-official-images |
 ssh | update-cluster |
 update-compute-fleet | version) ...
pcluster3-config-converter [-h] [-t CLUSTER_TEMPLATE]
 [-c CONFIG_FILE]
 [--force-convert]
 [-o OUTPUT_FILE]
```

Topik

- [pcluster](#)



- [pcluster3-config-converter](#)

## pcluster

`pcluster` adalah perintah AWS ParallelCluster CLI utama. Anda gunakan `pcluster` untuk meluncurkan dan mengelola cluster HPC di AWS Cloud.

`pcluster` menulis log perintah Anda ke `pcluster.log.#` file di `/home/user/.parallelcluster/`. Untuk informasi selengkapnya, lihat [pclusterLog CLI](#).

Untuk menggunakannya `pcluster`, Anda harus memiliki peran IAM dengan [izin](#) yang diperlukan untuk menjalankannya.

```
pcluster [-h]
```

### Argumen

#### `pcluster` *command*

Pilihan yang memungkinkan: [build-image](#)[configure](#)[create-cluster](#)[dcv-connect](#)[delete-cluster](#)[delete-cluster-instances](#)[delete-image](#)[describe-cluster](#)[describe-cluster-instances](#)[describe-compute-fleet](#)[describe-image](#)[export-cluster-log](#)[export-image-log](#)[get-cluster-log-events](#)[get-cluster-stack-events](#)[get-image-log-events](#)[get-image-stack-events](#)[list-clusters](#)[list-cluster-log-streams](#)[list-images](#)[list-image-log-streams](#)[list-official-images](#)[ssh](#)[update-cluster](#)[update-compute-fleet](#)[version](#)

### Sub-perintah:

#### Topik

- [pcluster build-image](#)
- [pcluster configure](#)
- [pcluster create-cluster](#)
- [pcluster dcv-connect](#)
- [pcluster delete-cluster](#)
- [pcluster delete-cluster-instances](#)

- [pcluster delete-image](#)
- [pcluster describe-cluster](#)
- [pcluster describe-cluster-instances](#)
- [pcluster describe-compute-fleet](#)
- [pcluster describe-image](#)
- [pcluster export-cluster-logs](#)
- [pcluster export-image-logs](#)
- [pcluster get-cluster-log-events](#)
- [pcluster get-cluster-stack-events](#)
- [pcluster get-image-log-events](#)
- [pcluster get-image-stack-events](#)
- [pcluster list-clusters](#)
- [pcluster list-cluster-log-streams](#)
- [pcluster list-images](#)
- [pcluster list-image-log-streams](#)
- [pcluster list-official-images](#)
- [pcluster ssh](#)
- [pcluster update-cluster](#)
- [pcluster update-compute-fleet](#)
- [pcluster version](#)

## pcluster build-image

Buat AWS ParallelCluster gambar kustom di Wilayah yang ditentukan.

```
pcluster build-image [-h]
 --image-configuration IMAGE_CONFIGURATION
 --image-id IMAGE_ID
 [--debug]
 [--dryrun DRYRUN]
 [--query QUERY]
 [--region REGION]
 [--rollback-on-failure ROLLBACK_ON_FAILURE]
```

```
[--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
[--validation-failure-level {INFO,WARNING,ERROR}]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster build-image`.

### **--image-configuration, -c *IMAGE\_CONFIGURATION***

Menentukan file konfigurasi gambar sebagai dokumen YAMAL.

### **--image-id, -i *IMAGE\_ID***

Menentukan id dari gambar yang akan dibangun.

### **--debug**

Aktifkan logging debug.

### **--dryrun *DRYRUN***

Kapan `true`, perintah melakukan validasi tanpa membuat sumber daya apa pun. Anda dapat menggunakan ini untuk memvalidasi konfigurasi gambar. (Default ke.) `false`

### **--query *QUERY***

JMESPath query untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan pengaturan [Wilayah](#) dalam file konfigurasi gambar, variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

Kapan `true`, secara otomatis memulai rollback tumpukan gambar pada kegagalan. (Default ke.) `false`

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS* ...]**

Mengidentifikasi satu atau lebih validator konfigurasi untuk ditekan.

Format: (ALL|type:[A-Za-z0-9]+)

## --validation-failure-level {INFO,WARNING,ERROR}

Menentukan tingkat validasi minimum yang akan menyebabkan penciptaan gagal. (Default ke.)  
ERROR

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```
$ pcluster build-image --image-configuration image-config.yaml --image-id custom-
alinux2-image
{
 "image": {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-alinux2-image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.2"
 }
}
```

### Warning

`pcluster build-image` menggunakan VPC default. Jika VPC default telah dihapus, mungkin dengan menggunakan AWS Control Tower atau AWS Landing Zone, maka ID subnet harus ditentukan dalam file konfigurasi gambar. Untuk informasi lebih lanjut, lihat [SubnetId](#).

## pcluster configure

Memulai wizard konfigurasi interaktif untuk AWS ParallelCluster versi 3. Untuk informasi selengkapnya, lihat [Konfigurasi dan buat cluster dengan antarmuka baris AWS ParallelCluster perintah](#).

```
pcluster configure [-h]
 --config CONFIG
 [--debug]
 [--region REGION]
```

## Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster configure`.

### **--config *CONFIG***

Jalur untuk menampilkan file konfigurasi yang dihasilkan.

### **--debug**

Aktifkan logging debug.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah harus ditentukan, menggunakan pengaturan [Wilayah](#) dalam file konfigurasi gambar, variabel `AWS_DEFAULT_REGION` lingkungan, region pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

## **pcluster create-cluster**

Menciptakan sebuah AWS ParallelCluster cluster.

```
pcluster create-cluster [-h]
 --cluster-configuration CLUSTER_CONFIGURATION
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--query QUERY]
 [--region REGION]
 [--rollback-on-failure ROLLBACK_ON_FAILURE]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]

 [--validation-failure-level {INFO,WARNING,ERROR}]
```

## Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster create-cluster`.

### **--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

Menentukan file konfigurasi cluster YANG.

**--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster yang akan dibuat.

Nama harus dimulai dengan karakter alfabet. Nama dapat memiliki hingga 60 karakter. Jika Slurm akuntansi diaktifkan, nama dapat memiliki hingga 40 karakter.

Karakter yang valid: a-z, A-Z, 0-9, dan - (tanda hubung).

**--debug**

Mengaktifkan logging debug.

**--dryrun *DRYRUN***

Kapan `true`, perintah melakukan validasi tanpa membuat sumber daya apa pun. Anda dapat menggunakan ini untuk memvalidasi konfigurasi cluster. (Default ke.) `false`

**--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan [Region](#) pengaturan dalam file konfigurasi cluster, variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

**--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

Kapan `true`, secara otomatis memulai rollback tumpukan cluster pada kegagalan. (Default ke.) `true`

**--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS* ...]**

Mengidentifikasi satu atau lebih validator konfigurasi untuk ditekan.

Format: (ALL|jenis:[A-Za-z0-9]+)

**--validation-failure-level {INFO,WARNING,ERROR}**

Menentukan tingkat validasi minimum yang akan menyebabkan penciptaan gagal. (Default ke.) `ERROR`

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster create-cluster -c cluster-config.yaml -n cluster-v3
```

```
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

## pcluster dcv-connect

Izin untuk terhubung ke node kepala melalui sesi interaktif dengan menggunakan NICE DCV.

```
pcluster dcv-connect [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--key-path KEY_PATH]
 [--region REGION]
 [--show-url]
```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster dcv-connect`.

### --cluster-name, -n *CLUSTER\_NAME*

Menentukan nama cluster.

### --debug

Mengaktifkan logging debug.

### --key-path *KEY\_PATH*

Menentukan jalur kunci SSH yang akan digunakan untuk koneksi.

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

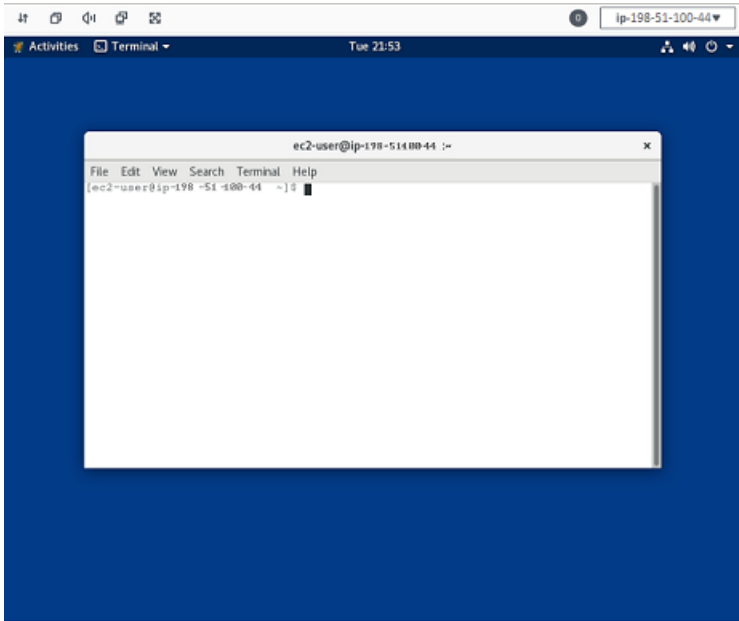
Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

## --show-url

Mencetak URL yang akan digunakan untuk koneksi DCV dan keluar.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster dcv-connect -n cluster-3Dcv -r us-east-1 --key-path /home/user/.ssh/key.pem
```



## pcluster delete-cluster

Memulai penghapusan cluster.

```
pcluster delete-cluster [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster delete-cluster`.



**--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

**--debug**

Mengaktifkan logging debug.

**--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster delete-cluster -n cluster-v3
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

## **pcluster delete-cluster-instances**

Memulai penghentian paksa dari semua node komputasi cluster. Ini tidak bekerja dengan AWS Batch cluster.

```
pcluster delete-cluster-instances [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--force FORCE]
 [--query QUERY]
```

```
[--region REGION]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster delete-cluster-instances`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--debug**

Mengaktifkan logging debug.

### **--force *FORCE***

Kapan `true`, memaksa penghapusan dengan mengabaikan kesalahan validasi. (Default ke.)  
`false`

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

```
$ pcluster delete-cluster-instances -n cluster-v3
```

## **pcluster delete-image**

Mulai penghapusan gambar kustom AWS ParallelCluster .

```
pcluster delete-image [-h]
 --image-id IMAGE_ID
 [--debug]
 [--force FORCE]
 [--query QUERY]
 [--region REGION]
```

## Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster delete-image`.

### **--image-id, -i *IMAGE\_ID***

Menentukan id dari gambar yang akan dihapus.

### **--debug**

Mengaktifkan logging debug.

### **--force *FORCE***

Kapan `true`, memaksa penghapusan jika ada instance yang menggunakan AMI atau jika AMI dibagikan. (Default ke.) `false`

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster delete-image --image-id custom-alinux2-image
{
 "image": {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "DELETE_IN_PROGRESS",
 "region": "us-east-1",
 "version": "3.1.4"
 }
}
```

## **pcluster describe-cluster**

Dapatkan informasi rinci tentang cluster.

```
pcluster describe-cluster [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster describe-cluster`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--debug**

Mengaktifkan logging debug.

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

Jelaskan detail cluster:

```
$ pcluster describe-cluster -n cluster-v3
{
 "creationTime": "2022-07-12T17:19:16.101Z",
 "headNode": {
 "launchTime": "2022-07-12T17:22:21.000Z",
 "instanceId": "i-1234567890abcdef0",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "running",
 "privateIpAddress": "192.0.2.0.196"
```

```

},
"loginNodes": {
 "status": "active",
 "address": "8af2145440569xyz.us-east-1.amazonaws.com",
 "scheme": "internet-facing|internal",
 "healthyNodes": 3,
 "unhealthyNodes": 0
},
"version": "3.1.4",
"clusterConfiguration": {
 "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
},
"tags": [
 {
 "value": "3.1.4",
 "key": "parallelcluster:version"
 }
],
"cloudFormationStackStatus": "CREATE_COMPLETE",
"clusterName": "cluster-v3",
"computeFleetStatus": "RUNNING",
"cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
"lastUpdatedTime": "2022-07-12T17:19:16.101Z",
"region": "us-east-1",
"clusterStatus": "CREATE_COMPLETE"
}

```

Gunakan `describe-cluster` untuk mengambil konfigurasi cluster:

```

$ curl -o - - $(pcluster describe-cluster -n cluster-v3 --query clusterConfiguration.url
| xargs echo)
Region: us-east-1
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: adpc
 Iam:
 S3Access:

```

```

- BucketName: cluster-v3-bucket
 KeyName: logs
 EnableWriteAccess: true
Scheduling:
 Scheduler: slurm
 SlurmQueues:
- Name: queue1
 ComputeResources:
- Name: t2micro
 InstanceType: t2.micro
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
- subnet-021345abcdef6789

```

## pcluster describe-cluster-instances

Jelaskan contoh dalam sebuah cluster.

```

pcluster describe-cluster-instances [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--next-token NEXT_TOKEN]
 [--node-type {HeadNode,ComputeNode,LoginNode}]
 [--query QUERY]
 [--queue-name QUEUE_NAME]
 [--region REGION]

```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster describe-cluster-instances`.

### --cluster-name, -n *CLUSTER\_NAME*

Menentukan nama cluster.

### --debug

Mengaktifkan logging debug.

**--next-token** *NEXT\_TOKEN*

Menentukan token yang akan digunakan untuk permintaan paginasi.

**--node-type** {HeadNode,ComputeNode,LoginNode}

Menentukan jenis node untuk daftar. Nilai yang didukung adalah HeadNode, ComputeNode dan LoginNode. Jika parameter ini tidak ditentukan, contoh HeadNode, ComputeNode dan LoginNode contoh dijelaskan.

**--query** *QUERY*

Menentukan query JMESPath untuk melakukan pada output.

**--queue-name** *QUEUE\_NAME*

Menentukan nama antrian ke daftar. Jika parameter ini tidak ditentukan, contoh di semua antrian dijelaskan.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel AWS\_DEFAULT\_REGION lingkungan, region pengaturan di [default] bagian ~/.aws/config file, atau --region parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster describe-cluster-instances -n cluster-v3
{
 "instances": [
 {
 "launchTime": "2022-07-12T17:22:21.000Z",
 "instanceId": "i-1234567890abcdef0",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "running",
 "nodeType": "HeadNode",
 "privateIpAddress": "192.0.2.0.196"
 },
 {
 "launchTime": "2022-07-12T17:37:42.000Z",
 "instanceId": "i-021345abcdef6789",
 "queueName": "queue1",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
```

```
 "state": "pending",
 "nodeType": "ComputeNode",
 "privateIpAddress": "192.0.2.0.196"
 }
]
}
```

## **pcluster describe-compute-fleet**

Jelaskan status armada komputasi.

```
pcluster describe-compute-fleet [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster describe-compute-fleet`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--debug**

Mengaktifkan logging debug.

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
```



```
"status": "RUNNING",
"lastStatusUpdatedTime": "2022-07-12T17:24:26.000Z"
}
```

## pcluster describe-image

Dapatkan informasi rinci tentang gambar.

```
pcluster describe-image [-h]
 --image-id IMAGE_ID
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster describe-image`.

### --image-id, -i *IMAGE\_ID*

Menentukan ID gambar.

### --debug

Mengaktifkan logging debug.

### --query *QUERY*

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```
$ pcluster describe-image --image-id custom-linux2-image
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678-v1-do-not-delete.../configs/image-
config.yaml" }
}
```

```

},
"imageId": "custom-alinux2-image",
"creationTime": "2022-04-05T20:23:07.000Z"
"imageBuildStatus": "BUILD_COMPLETE",
"region": "us-east-1",
"ec2AmiInfo": {
 "amiName": "custom-alinux2-image 2022-04-05T19-55-22.518Z",
 "amiId": "ami-1234abcd5678efgh",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.268-205.500.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.14.2-1.amzn2.x86_64, dcv-2021.3.11591-1.el7.x86_64, slurm-21-08-6-1",
 "state": "AVAILABLE",
"tags": [
 {
 "value": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
 "key": "Ec2ImageBuilderArn"
 },
 {
 "value": "parallelcluster-1234abcd5678efgh-v1-do-not-delete",
 "key": "parallelcluster:s3_bucket"
 },
 {
 "value": "custom-alinux2-image",
 "key": "parallelcluster:image_name"
 },
 {
 "value": "available",
 "key": "parallelcluster:build_status"
 },
 {
 "value": "s3://parallelcluster-1234abcd5678efgh-v1-do-not-delete/
parallelcluster/3.1.2/images/custom-alinux2-image-1234abcd5678efgh/configs/image-
config.yaml",
 "key": "parallelcluster:build_config"
 },
 {
 "value": "EC2 Image Builder",
 "key": "CreatedBy"
 },
 {
 "value": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-alinux2-image",
 "key": "parallelcluster:build_log"
 }
]
}

```

```
 },
 {
 "value": "4.14.268-205.500.amzn2.x86_64",
 "key": "parallelcluster:kernel_version"
 },
 {
 "value": "arn:aws:imagebuilder:us-east-1:444455556666:image/amazon-linux-2-
x86/2022.3.16/1",
 "key": "parallelcluster:parent_image"
 },
 {
 "value": "3.1.2",
 "key": "parallelcluster:version"
 },
 {
 "value": "0.5.14",
 "key": "parallelcluster:munge_version"
 },
 {
 "value": "21-08-6-1",
 "key": "parallelcluster:slurm_version"
 },
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 {
 "value": "alinux2-image",
 "key": "parallelcluster:image_id"
 },
 {
 "value": "3.2.3",
 "key": "parallelcluster:pmix_version"
 },
 {
 "value": "parallelcluster/3.7.0/images/alinux2-image-abcd1234efgh56781234",
 "key": "parallelcluster:s3_image_dir"
 },
 {
 "value": "1.14.2-1.amzn2.x86_64",
 "key": "parallelcluster:efa_version"
 },
 {
 "value": "alinux2",
```

```

 "key": "parallelcluster:os"
 },
 {
 "value": "aws-parallelcluster-cookbook-3.1.2",
 "key": "parallelcluster:bootstrap_file"
 },
 {
 "value": "1.8.23-10.amzn2.1.x86_64",
 "key": "parallelcluster:sudo_version"
 },
 {
 "value": "2.10.8-5.amzn2.x86_64",
 "key": "parallelcluster:lustre_version"
 }
],
"architecture": "x86_64"
},
"version": "3.1.2"
}

```

## pcluster export-cluster-logs

Ekspor log cluster ke tar.gz arsip lokal dengan melewati Amazon S3 Bucket.

```

pcluster export-cluster-logs [-h]
 --bucket BUCKET_NAME
 --cluster-name CLUSTER_NAME
 [--bucket-prefix BUCKET_PREFIX]
 [--debug]
 [--end-time END_TIME]
 [--filters FILTER [FILTER ...]]
 [--keep-s3-objects KEEP_S3_OBJECTS]
 [--output-file OUTPUT_FILE]
 [--region REGION]
 [--start-time START_TIME]

```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster export-cluster-logs`.

**--bucket** *BUCKET\_NAME*

Menentukan nama bucket Amazon S3 untuk mengekspor data log klaster ke. Itu harus di Wilayah yang sama dengan cluster.

**Note**

Anda harus menambahkan izin ke kebijakan bucket Amazon S3 untuk CloudWatch memberikan akses. Untuk informasi selengkapnya, lihat [Menetapkan izin pada bucket Amazon S3](#) di CloudWatch Panduan Pengguna Log.

**--cluster-name, -n** *CLUSTER\_NAME*

Menentukan nama cluster.

**--bucket-prefix** *BUCKET\_PREFIX*

Menentukan jalur di bucket Amazon S3 tempat data log yang diekspor akan disimpan.

Secara default, awalan ember adalah:

```
cluster-name-logs-202209061743.tar.gz
```

*202209061743* adalah waktu saat ini dalam %Y%m%d%H%M format.

**--debug**

Mengaktifkan logging debug.

**--end-time** *END\_TIME*

Menentukan akhir rentang waktu untuk mengumpulkan peristiwa log, dinyatakan dalam format ISO 8601 (YYYY-MM-DDThh:mm:ssZ, misalnya 2021-01-01T20:00:00Z '). Acara dengan stempel waktu yang sama dengan atau lebih lambat dari waktu ini tidak termasuk. Elemen waktu (misalnya menit dan detik) dapat dihilangkan. Nilai default adalah waktu ketika ini.

**--filters** *FILTER* [*FILTER* ...]

Menentukan filter untuk log. Format: Name=a,Values=1 Name=b,Values=2,3. Filter yang didukung adalah:

private-dns-name

Menentukan bentuk singkat dari nama DNS pribadi dari instance (misalnya). ip-10-0-0-101

## node-type

Menentukan jenis node, satu-satunya nilai yang diterima untuk filter ini adalah `HeadNode`.

### **--keep-s3-objects** *KEEP\_S3\_OBJECTS*

Jika `true`, objek yang diekspor ekspor ke Amazon S3 disimpan. (Default ke.) `false`

### **--output-file** *OUTPUT\_FILE*

Menentukan path file untuk menyimpan arsip log ke. Jika ini disediakan, maka log disimpan secara lokal. Jika tidak, mereka diunggah ke Amazon S3 dengan URL yang dikembalikan dalam output. Defaultnya adalah mengunggah ke Amazon S3.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

### **--start-time** *START\_TIME*

Menentukan awal rentang waktu, dinyatakan dalam format ISO 8601 (`YYYY-MM-DDThh:mm:ssZ`, misalnya). `2021-01-01T20:00:00Z` Peristiwa log dengan stempel waktu yang sama dengan waktu ini atau lebih lambat dari waktu ini disertakan. Jika tidak ditentukan, defaultnya adalah waktu cluster dibuat.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
{
 "url": "https://cluster-v3-bucket..."
}
```

## **pcluster export-image-logs**

Ekspor log tumpukan pembuat gambar ke `tar.gz` arsip lokal dengan melewati Amazon S3 Bucket.

```
pcluster export-image-logs [-h]
 --bucket BUCKET
 --image-id IMAGE_ID
 [--bucket-prefix BUCKET_PREFIX]
 [--debug]
 [--end-time END_TIME]
```

```
[--keep-s3-objects KEEP_S3_OBJECTS]
[--output-file OUTPUT_FILE]
[--region REGION]
[--start-time START_TIME]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster export-image-logs`.

### **--bucket *BUCKET\_NAME***

Menentukan nama bucket Amazon S3 untuk mengekspor log build image ke. Itu harus di Wilayah yang sama dengan gambar.

#### Note

Anda harus menambahkan izin ke kebijakan bucket Amazon S3 untuk CloudWatch memberikan akses. Untuk informasi selengkapnya, lihat [Menetapkan izin pada bucket Amazon S3](#) di CloudWatch Panduan Pengguna Log.

### **--image-id, -i *IMAGE\_ID***

ID gambar yang lognya akan diekspor.

### **--bucket-prefix *BUCKET\_PREFIX***

Menentukan jalur di bucket Amazon S3 tempat data log yang diekspor akan disimpan.

Secara default, awalan ember adalah:

```
ami-id-logs-202209061743.tar.gz
```

*202209061743* adalah waktu saat ini dalam `%Y%m%d%H%M` format.

### **--debug**

Mengaktifkan logging debug.

### **--end-time *END\_TIME***

Menentukan akhir rentang waktu untuk mengumpulkan peristiwa log, dinyatakan dalam format ISO 8601 (YYYY-MM-DDThh:mm:ssZ, misalnya `2021-01-01T20:00:00Z` '). Acara dengan

stempel waktu yang sama dengan atau lebih lambat dari waktu ini tidak termasuk. Elemen waktu (misalnya menit dan detik) dapat dihilangkan. Nilai default adalah waktu ketika ini.

### **--keep-s3-objects** *KEEP\_S3\_OBJECTS*

Jika `true`, objek yang diekspor ekspor ke Amazon S3 disimpan. (Default ke.) `false`

### **--output-file** *OUTPUT\_FILE*

Menentukan path file untuk menyimpan arsip log ke. Jika ini disediakan, maka log disimpan secara lokal. Jika tidak, mereka diunggah ke Amazon S3 dengan URL yang dikembalikan dalam output. Defaultnya adalah mengunggah ke Amazon S3.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/ .aws/config` file, atau `--region` parameter.

### **--start-time** *START\_TIME*

Menentukan awal rentang waktu, dinyatakan dalam format ISO 8601 (`YYYY-MM-DDThh:mm:ssZ`, misalnya). `2021-01-01T20:00:00Z` Peristiwa log dengan stempel waktu yang sama dengan waktu ini atau lebih lambat dari waktu ini disertakan. Jika tidak ditentukan, defaultnya adalah waktu cluster dibuat.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster export-image-logs --bucket image-v3-bucket --image-id ami-1234abcd5678efgh
{
 "url": "https://image-v3-bucket..."
}
```

## **pcluster get-cluster-log-events**

Ambil peristiwa yang terkait dengan aliran log.

```
pcluster get-cluster-log-events [-h]
 --cluster-name CLUSTER_NAME
 --log-stream-name LOG_STREAM_NAME
 [--debug]
 [--end-time END_TIME]
```



```
[--limit LIMIT]
[--next-token NEXT_TOKEN]
[--query QUERY]
[--region REGION]
[--start-from-head START_FROM_HEAD]
[--start-time START_TIME]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster get-cluster-log-events`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--log-stream-name *LOG\_STREAM\_NAME***

Menentukan nama aliran log. Anda dapat menggunakan `list-cluster-log-streams` perintah untuk mengambil aliran log yang terkait dengan peristiwa atau peristiwa.

### **--debug**

Mengaktifkan logging debug.

### **--end-time *END\_TIME***

Menentukan akhir rentang waktu, dinyatakan dalam format ISO 8601 (YYYY-MM-DDThh:mm:ssZ, misalnya). 2021-01-01T20:00:00Z Acara dengan stempel waktu yang sama dengan atau lebih lambat dari waktu ini tidak termasuk.

### **--limit *LIMIT***

Menentukan jumlah maksimum peristiwa log dikembalikan. Jika nilai tidak ditentukan, maksimum adalah sebanyak peristiwa log yang dapat muat dalam ukuran respons 1 MB, hingga 10.000 peristiwa log.

### **--next-token *NEXT\_TOKEN***

Menentukan token yang akan digunakan untuk permintaan paginasi.

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

**--start-from-head *START\_FROM\_HEAD***

Jika nilainya `true`, peristiwa log paling awal dikembalikan terlebih dahulu. Jika nilainya `false`, peristiwa log terbaru dikembalikan terlebih dahulu. (Default ke.) `false`

**--start-time *START\_TIME***

Menentukan awal rentang waktu, dinyatakan dalam format ISO 8601 (`YYYY-MM-DDThh:mm:ssZ`, misalnya). `2021-01-01T20:00:00Z` Acara dengan stempel waktu yang sama dengan waktu ini atau lebih lambat dari waktu ini disertakan.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster get-cluster-log-events \
 -c cluster-v3 \
 -r us-east-1 \
 --log-stream-name ip-198-51-100-44.i-1234567890abcdef0.clustermgtd \
 --limit 3
{
 "nextToken": "f/36966906399261933213029082268132291405859205452101451780/s",
 "prevToken": "b/36966906399239632467830551644990755687586557090595471362/s",
 "events": [
 {
 "message": "2022-07-12 19:16:53,379 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Performing node maintenance actions",
 "timestamp": "2022-07-12T19:16:53.379Z"
 },
 {
 "message": "2022-07-12 19:16:53,380 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Following nodes are currently in replacement: (x0) []",
 "timestamp": "2022-07-12T19:16:53.380Z"
 },
 {
 "message": "2022-07-12 19:16:53,380 -
[slurm_plugin.clustermgtd:_terminate_orphaned_instances] - INFO - Checking for
orphaned instance",
 "timestamp": "2022-07-12T19:16:53.380Z"
 }
]
}
```

```
}
]
}
```

## **pcluster get-cluster-stack-events**

Ambil peristiwa yang terkait dengan tumpukan untuk cluster yang ditentukan.

### Note

Mulai versi 3.6.0, AWS ParallelCluster menggunakan tumpukan bersarang untuk membuat sumber daya yang terkait dengan antrian dan sumber daya komputasi. `GetClusterStackEventsAPI` dan `pcluster get-cluster-stack-events` perintah hanya mengembalikan peristiwa tumpukan utama cluster. Anda dapat melihat peristiwa tumpukan cluster, termasuk yang terkait dengan antrian dan sumber daya komputasi, di konsol. CloudFormation

```
pcluster get-cluster-stack-events [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster get-cluster-stack-events`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--debug**

Mengaktifkan logging debug.

### **--next-token *NEXT\_TOKEN***

Menentukan token yang akan digunakan untuk permintaan paginasi.

**--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster get-cluster-stack-events \
 -n cluster-v3 \
 -r us-east-1 \
 --query "events[0]"
{
 "eventId": "1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "physicalResourceId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "resourceStatus": "CREATE_COMPLETE",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "stackName": "cluster-v3",
 "logicalResourceId": "cluster-v3",
 "resourceType": "AWS::CloudFormation::Stack",
 "timestamp": "2022-07-12T18:29:12.140Z"
}
```

**pcluster get-image-log-events**

Ambil peristiwa yang terkait dengan pembuatan gambar.

```
pcluster get-image-log-events [-h]
 --image-id IMAGE_ID
 --log-stream-name LOG_STREAM_NAME
 [--debug]
 [--end-time END_TIME]
 [--limit LIMIT]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

```
[--start-from-head START_FROM_HEAD]
[--start-time START_TIME]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster get-image-log-events`.

### **--image-id, -i *IMAGE\_ID***

Menentukan Id dari gambar.

### **--log-stream-name *LOG\_STREAM\_NAME***

Menentukan nama aliran log. Anda dapat menggunakan `list-image-log-streams` perintah untuk mengambil aliran log yang terkait dengan peristiwa atau peristiwa.

### **--debug**

Mengaktifkan logging debug.

### **--end-time *END\_TIME***

Menentukan akhir rentang waktu, dinyatakan dalam format ISO 8601 (YYYY-MM-DDThh:mm:ssZ, misalnya). 2021-01-01T20:00:00Z Acara dengan stempel waktu yang sama dengan atau lebih lambat dari waktu ini tidak termasuk.

### **--limit *LIMIT***

Menentukan jumlah maksimum peristiwa log dikembalikan. Jika nilai tidak ditentukan, maksimum adalah sebanyak peristiwa log yang dapat muat dalam ukuran respons 1 MB, hingga 10.000 peristiwa log.

### **--next-token *NEXT\_TOKEN***

Menentukan token yang akan digunakan untuk permintaan paginasi.

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

**--start-from-head** *START\_FROM\_HEAD*

Jika nilainya `true`, peristiwa log paling awal dikembalikan terlebih dahulu. Jika nilainya `false`, peristiwa log terbaru dikembalikan terlebih dahulu. (Default ke.) `false`

**--start-time** *START\_TIME*

Menentukan awal rentang waktu, dinyatakan dalam format ISO 8601 (YYYY-MM-DDThh:mm:ssZ, misalnya). 2021-01-01T20:00:00Z Acara dengan stempel waktu yang sama dengan atau lebih lambat dari waktu ini disertakan.

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```
$ pcluster get-image-log-events --image-id custom-alinux2-image --region us-east-1 --
log-stream-name 3.1.2/1 --limit 3
{
 "nextToken": "f/36778317771100849897800729464621464113270312017760944178/s",
 "prevToken": "b/36778317766952911290874033560295820514557716777648586800/s",
 "events": [
 {
 "message": "ExecuteBash: FINISHED EXECUTION",
 "timestamp": "2022-04-05T22:13:26.633Z"
 },
 {
 "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-1234abcd-56ef-78gh-90ij-abcd1234efgh/3.1.2/1",
 "timestamp": "2022-04-05T22:13:26.741Z"
 },
 {
 "message": "TOE has completed execution successfully",
 "timestamp": "2022-04-05T22:13:26.819Z"
 }
]
}
```

**pcluster get-image-stack-events**

Ambil peristiwa yang terkait dengan tumpukan untuk build gambar yang ditentukan.

```
pcluster get-image-stack-events [-h]
 --image-id IMAGE_ID
 [--debug]
```

```

[--next-token NEXT_TOKEN]
[--query QUERY]
[--region REGION]

```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster get-image-stack-events`.

### **--image-id, -i *IMAGE\_ID***

Menentukan ID gambar.

### **--debug**

Mengaktifkan logging debug.

### **--next-token *NEXT\_TOKEN***

Menentukan token yang akan digunakan untuk permintaan paginasi.

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```

$ pcluster get-image-stack-events --image-id custom-alinux2-image --region us-east-1 --
query "events[0]"
{
 "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
 "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "Resource creation Initiated",
 "resourceProperties": "{\"InfrastructureConfigurationArn\":
\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\", \"ImageRecipeArn

```

```

\":"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-custom-alinux2-image/3.1.2\","DistributionConfigurationArn
\":"arn:aws:imagebuilder:us-east-1:123456789012:distribution-
configuration/parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh",
"EnhancedImageMetadataEnabled\":"false",\ "Tags\":"{\ "parallelcluster:image_name\":"
"custom-alinux2-image\","parallelcluster:image_id\":"custom-alinux2-image"}",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/custom-alinux2-
image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "stackName": "custom-alinux2-image",
 "logicalResourceId": "ParallelClusterImage",
 "resourceType": "AWS::ImageBuilder::Image",
 "timestamp": "2022-04-05T21:39:24.725Z"
}

```

## pcluster list-clusters

Ambil daftar cluster yang ada.

```

pcluster list-clusters [-h]
 [--cluster-status {CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
 DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
 UPDATE_COMPLETE,UPDATE_FAILED}]
 [{CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
 DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
 UPDATE_COMPLETE,UPDATE_FAILED} ...]
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]

```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster list-clusters`.

```

--cluster-status {CREATE_IN_PROGRESS, CREATE_FAILED, CREATE_COMPLETE,
DELETE_IN_PROGRESS, DELETE_FAILED, UPDATE_IN_PROGRESS, UPDATE_COMPLETE,
UPDATE_FAILED} [{CREATE_IN_PROGRESS, CREATE_FAILED, CREATE_COMPLETE,
DELETE_IN_PROGRESS, DELETE_FAILED, UPDATE_IN_PROGRESS, UPDATE_COMPLETE,
UPDATE_FAILED} ...]

```

Menentukan daftar status cluster untuk memfilter untuk. (Default ke.) `all`



**--debug**

Mengaktifkan logging debug.

**--next-token** *NEXT\_TOKEN*

Menentukan token yang akan digunakan untuk permintaan paginasi.

**--query** *QUERY*

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster list-clusters
{
 "clusters": [
 {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_COMPLETE"
 }
]
}
```

**pcluster list-cluster-log-streams**

Ambil daftar aliran log yang terkait dengan cluster.

```
pcluster list-cluster-log-streams [-h]
 --cluster-name CLUSTER_NAME
 [--filters FILTERS [FILTERS ...]]
 [--next-token NEXT_TOKEN] [--debug]
```

```
[--query QUERY]
[--region REGION]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster list-cluster-log-streams`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--debug**

Mengaktifkan logging debug.

### **--filters *FILTERS* [*FILTERS* ...]**

Menentukan filter untuk aliran log. Format: `Name=a,Values=1 Name=b,Values=2,3`. Filter yang didukung adalah:

`private-dns-name`

Menentukan bentuk singkat dari nama DNS pribadi dari instance (misalnya). `ip-10-0-0-101`  
`node-type`

Menentukan jenis node, satu-satunya nilai yang diterima untuk filter ini adalah `HeadNode`.

### **--next-token *NEXT\_TOKEN***

Menentukan token yang akan digunakan untuk permintaan paginasi.

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster list-cluster-log-streams \
```

```
-n cluster-v3 \
-r us-east-1 \
--query 'LogStreams[*].LogStreamName'
[
 "ip-172-31-58-205.i-1234567890abcdef0.cfn-init",
 "ip-172-31-58-205.i-1234567890abcdef0.chef-client",
 "ip-172-31-58-205.i-1234567890abcdef0.cloud-init",
 "ip-172-31-58-205.i-1234567890abcdef0.clustermgtd",
 "ip-172-31-58-205.i-1234567890abcdef0.slurmctld",
 "ip-172-31-58-205.i-1234567890abcdef0.supervisord",
 "ip-172-31-58-205.i-1234567890abcdef0.system-messages"
]
```

## pcluster list-images

Ambil daftar gambar kustom yang ada.

```
pcluster list-images [-h]
 --image-status {AVAILABLE,PENDING,FAILED}
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster list-images`.

### --image-status {AVAILABLE,PENDING,FAILED}

Filter gambar yang dikembalikan berdasarkan status yang disediakan.

### --debug

Mengaktifkan logging debug.

### --next-token **NEXT\_TOKEN**

Menentukan token yang akan digunakan untuk permintaan paginasi.

### --query **QUERY**

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```
$ pcluster list-images --image-status AVAILABLE
{
 "images": [
 {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "BUILD_COMPLETE",
 "ec2AmiInfo": {
 "amiId": "ami-1234abcd5678efgh"
 },
 "region": "us-east-1",
 "version": "3.1.2"
 }
]
}
```

**pcluster list-image-log-streams**

Ambil daftar aliran log yang terkait dengan gambar.

```
pcluster list-image-log-streams [-h]
 --image-id IMAGE_ID
 [--next-token NEXT_TOKEN] [--debug]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

**-h, --help**

Menampilkan teks bantuan untuk `pcluster list-image-log-streams`.

**--image-id, -i *IMAGE\_ID***

Menentukan ID gambar.

**--debug**

Mengaktifkan logging debug.

**--next-token** *NEXT\_TOKEN*

Menentukan token yang akan digunakan untuk permintaan paginasi.

**--query** *QUERY*

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```
$ pcluster list-image-log-streams --image-id custom-linux2-image --region us-east-1 --
query 'LogStreams[*].LogStreamName'
[
 "3.0.0/1",
 "3.1.2/1"
]
```

**pcluster list-official-images**

Jelaskan AWS ParallelCluster AMI resmi.

```
pcluster list-official-images [-h]
 [--architecture ARCHITECTURE]
 [--debug]
 [--os OS]
 [--query QUERY]
 [--region REGION]
```

Argumen bernama

**-h, --help**

Menampilkan teks bantuan untuk `pcluster list-official-images`.

**--architecture** *ARCHITECTURE*

Menentukan arsitektur yang akan digunakan untuk menyaring hasil. Jika parameter ini tidak ditentukan, semua arsitektur dikembalikan.

**--debug**

Mengaktifkan logging debug.

**--os** *OS*

Menentukan sistem operasi yang akan digunakan untuk menyaring hasil. Jika parameter ini tidak ditentukan, semua sistem operasi dikembalikan.

**--query** *QUERY*

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan pengaturan [Wilayah](#) dalam file konfigurasi gambar, variabel `AWS_DEFAULT_REGION` lingkungan, region pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.2:

```
$ pcluster list-official-images
{
 "images": [
 {
 "amiId": "ami-015cfefb4e0d6306b2",
 "os": "ubuntu2004",
 "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-x86_64-202202261505
2022-02-26T15-08-34.759Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-036f23237ce49d25b",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-x86_64-202202261505
2022-02-26T15-08-17.558Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
],
}
```

```
{
 "amiId": "ami-09e5327e694d89ef4",
 "os": "ubuntu2004",
 "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.736Z",
 "version": "3.1.2",
 "architecture": "arm64"
},
{
 "amiId": "ami-0b9b0874c35f626ae",
 "os": "alinux2",
 "name": "aws-parallelcluster-3.1.2-amzn2-hvm-x86_64-202202261505
2022-02-26T15-08-31.311Z",
 "version": "3.1.2",
 "architecture": "x86_64"
},
{
 "amiId": "ami-0bf6d01f398f3737e",
 "os": "centos7",
 "name": "aws-parallelcluster-3.1.2-centos7-hvm-x86_64-202202261505
2022-02-26T15-08-25.001Z",
 "version": "3.1.2",
 "architecture": "x86_64"
},
{
 "amiId": "ami-0d0de4f95f56374bc",
 "os": "alinux2",
 "name": "aws-parallelcluster-3.1.2-amzn2-hvm-arm64-202202261505
2022-02-26T15-08-46.088Z",
 "version": "3.1.2",
 "architecture": "arm64"
},
{
 "amiId": "ami-0ebf7bc54b8740dc6",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.293Z",
 "version": "3.1.2",
 "architecture": "arm64"
}
]
```

## pcluster ssh

Menjalankan ssh perintah dengan nama pengguna cluster dan alamat IP yang telah diisi sebelumnya. Argumen arbitrer ditambahkan ke akhir ssh baris perintah.

```
pcluster ssh [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--region REGION]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster ssh`.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster untuk terhubung ke.

### **--debug**

Mengaktifkan logging debug.

### **--dryrun *DRYRUN***

Kapan `true`, mencetak baris perintah yang akan dijalankan dan keluar. (Default ke.) `false`

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh:

```
$ pcluster ssh --cluster-name mycluster -i ~/.ssh/id_rsa
```

Menjalankan ssh perintah dengan nama pengguna dan alamat IP cluster yang sudah terisi sebelumnya:

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```



## pcluster update-cluster

Memperbarui cluster yang ada agar sesuai dengan pengaturan file konfigurasi yang ditentukan.

```
pcluster update-cluster [-h]
 --cluster-configuration CLUSTER_CONFIGURATION
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--force-update FORCE_UPDATE]
 [--query QUERY]
 [--region REGION]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster update-cluster`.

### **--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

Menentukan file konfigurasi cluster YANG.

### **--cluster-name, -n *CLUSTER\_NAME***

Menentukan nama cluster.

### **--debug**

Mengaktifkan logging debug.

### **--dryrun *DRYRUN***

Kapan `true`, melakukan validasi tanpa memperbarui cluster dan membuat sumber daya apa pun. Ini dapat digunakan untuk memvalidasi konfigurasi gambar dan memperbarui persyaratan. (Default ke.) `false`

### **--force-update *FORCE\_UPDATE***

Kapan `true`, memaksa pembaruan dengan mengabaikan kesalahan validasi pembaruan. (Default ke.) `false`

### **--query *QUERY***

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan [Region](#) pengaturan dalam file konfigurasi cluster, variabel `AWS_DEFAULT_REGION` lingkungan, region pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

**--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

Mengidentifikasi satu atau lebih validator konfigurasi untuk ditekan.

Format: `(ALL|type:[A-Za-z0-9]+)`

**--validation-failure-level *{INFO,WARNING,ERROR}***

Menentukan tingkat kegagalan validasi dilaporkan untuk update.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster update-cluster -c cluster-config.yaml -n cluster-v3 -r us-east-1
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "UPDATE_IN_PROGRESS"
 },
 "changeSet": [
 {
 "parameter": "HeadNode.Iam.S3Access",
 "requestedValue": {
 "BucketName": "pc-beta-test",
 "KeyName": "output",
 "EnableWriteAccess": false
 }
 },
 {
 "parameter": "HeadNode.Iam.S3Access",
 "currentValue": {
 "BucketName": "pcluster-east-test-bucket",
 "KeyName": "logs",
 "EnableWriteAccess": true
 }
 }
]
}
```

```

 }
 }
]
}

```

## pcluster update-compute-fleet

Memperbarui status armada komputasi cluster.

```

pcluster update-compute-fleet [-h]
 --cluster-name CLUSTER_NAME
 --status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

 [--debug]
 [--query QUERY]
 [--region REGION]

```

Argumen bernama

### -h, --help

Menampilkan teks bantuan untuk `pcluster update-compute-fleet`.

### --cluster-name, -n *CLUSTER\_NAME*

Menentukan nama cluster.

### --status {START\_REQUESTED,STOP\_REQUESTED,ENABLED,DISABLED}

Menentukan status diterapkan ke armada komputasi cluster. Status `START_REQUESTED` dan `STOP_REQUESTED` sesuai dengan penjadwal slurm sementara status `ENABLED` dan `DISABLED` sesuai dengan penjadwal. AWS Batch

### --debug

Mengaktifkan logging debug.

### --query *QUERY*

Menentukan query JMESPath untuk melakukan pada output.

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

Menentukan Wilayah AWS untuk menggunakan. Wilayah AWS Harus ditentukan, menggunakan variabel `AWS_DEFAULT_REGION` lingkungan, `region` pengaturan di `[default]` bagian `~/.aws/config` file, atau `--region` parameter.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster update-compute-fleet -n cluster-v3 --status STOP_REQUESTED
{
 "status": "STOP_REQUESTED",
 "lastStatusUpdateTime": "2022-07-12T20:19:47.653Z"
}
```

## pcluster version

Menampilkan versi AWS ParallelCluster.

```
pcluster version [-h] [--debug]
```

Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster version`.

### **--debug**

Mengaktifkan logging debug.

Contoh menggunakan AWS ParallelCluster versi 3.1.4:

```
$ pcluster version
{
 "version": "3.1.4"
}
```

## pcluster3-config-converter

Membaca file konfigurasi AWS ParallelCluster versi 2 dan menulis file konfigurasi AWS ParallelCluster versi 3.

```
pcluster3-config-converter [-h]
 [-t CLUSTER_TEMPLATE]
 [-c CONFIG_FILE]
 [--force-convert]
```

```
[-o OUTPUT_FILE]
```

## Argumen bernama

### **-h, --help**

Menampilkan teks bantuan untuk `pcluster3-config-converter`.

### **-t CLUSTER\_TEMPLATE, --cluster-template CLUSTER\_TEMPLATE**

Menentukan [\[cluster\]bagian](#) dari file konfigurasi untuk mengkonversi. Jika tidak ditentukan script akan mencari parameter [cluster-template](#) di [\[global\]bagian](#) atau akan mencari `[cluster default]`

### **-c CONFIG\_FILE, --config-file CONFIG\_FILE**

Menentukan AWS ParallelCluster versi 2 file konfigurasi yang akan dibaca.

### **--force-convert**

Mengaktifkan konversi meskipun satu atau lebih pengaturan tidak didukung dan tidak disarankan.

### **-o OUTPUT\_FILE, --output-file OUTPUT\_FILE**

Menentukan AWS ParallelCluster versi 3 file konfigurasi yang akan ditulis. Jika parameter ini tidak ditentukan, konfigurasi ditulis ke stdout.

#### Note

`pcluster3-config-converter` Perintah ditambahkan dalam AWS ParallelCluster versi 3.0.1.

## File konfigurasi

AWS ParallelCluster menggunakan file YAMM 1.1 untuk parameter konfigurasi.

Versi AWS ParallelCluster 3 saat ini: 3.7.0.

### Topik

- [File konfigurasi cluster](#)

- [Membangun file konfigurasi gambar](#)

## File konfigurasi cluster

AWS ParallelCluster versi 3 menggunakan file konfigurasi terpisah untuk mengontrol definisi infrastruktur cluster dan definisi AMI kustom. Semua file konfigurasi menggunakan file YAMAL 1.1. Informasi terperinci untuk masing-masing file konfigurasi ini ditautkan di bawah ini. Untuk beberapa contoh konfigurasi, lihat [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example\\_configs](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs).

Objek ini digunakan untuk konfigurasi cluster AWS ParallelCluster versi 3.

Versi AWS ParallelCluster 3 saat ini: 3.7.0.

### Topik

- [Properti file konfigurasi cluster](#)
- [Bagian Imds](#)
- [Bagian Image](#)
- [Bagian HeadNode](#)
- [Bagian Scheduling](#)
- [Bagian SharedStorage](#)
- [Bagian Iam](#)
- [Bagian LoginNodes](#)
- [Bagian Monitoring](#)
- [Bagian Tags](#)
- [Bagian AdditionalPackages](#)
- [Bagian DirectoryService](#)
- [Bagian DeploymentSettings](#)

## Properti file konfigurasi cluster

### Region(Opsional,String)

Menentukan Wilayah AWS untuk cluster. Misalnya, us-east-2.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## CustomS3Bucket(Opsional,String)

Menentukan nama bucket Amazon S3 yang dibuat untuk Akun AWS menyimpan sumber daya yang digunakan oleh cluster, seperti file konfigurasi klaster. AWS ParallelCluster memelihara satu bucket Amazon S3 di setiap Wilayah AWS tempat Anda membuat cluster. Secara default, ember Amazon S3 ini diberi nama. `parallelcluster-hash-v1-D0-NOT-DELETE`

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan. Jika Anda memaksa pembaruan, nilai baru akan diabaikan dan nilai lama akan digunakan.](#)

## AdditionalResources(Opsional,String)

Mendefinisikan AWS CloudFormation template tambahan untuk diluncurkan bersama dengan cluster. Template tambahan ini digunakan untuk membuat sumber daya yang berada di luar cluster tetapi merupakan bagian dari siklus hidup cluster.

Nilai harus berupa URL HTTPS ke templat publik, dengan semua parameter yang disediakan.

Tidak ada nilai default.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## Bagian **Imds**

(Opsional) Menentukan konfigurasi layanan metadata instance global (IMDS).

[Imds:](#)

[ImdsSupport:](#) *string*

### **Imds**properti

#### ImdsSupport(Opsional,String)

Menentukan versi IMDS yang didukung di node cluster. Nilai yang didukung adalah `v1.0` dan `v2.0`. Nilai default-nya adalah `v2.0`.

Jika `ImdsSupport` disetel ke `v1.0`, IMDSv1 dan IMDSv2 didukung.

Jika `ImdsSupport` disetel ke `v2.0`, hanya IMDSv2 yang didukung.

Untuk informasi selengkapnya, lihat [Menggunakan IMDSv2](#) di Panduan Pengguna EC2 untuk instance Linux.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

**Note**

Dimulai dengan AWS ParallelCluster 3.7.0, nilai `ImdsSupport` defaultnya adalah `v2.0`. Kami menyarankan Anda mengatur `v2.0` dan `ImdsSupport` mengganti `IMDSv1` dengan `IMDSv2` dalam panggilan tindakan kustom Anda.

Support for [Imds/ImdsSupport](#) ditambahkan dengan AWS ParallelCluster versi 3.3.0.

## Bagian Image

(Wajib) Mendefinisikan sistem operasi untuk cluster.

Image:

`Os`: *string*

`CustomAmi`: *string*

### Imageproperti

`Os`(Diperlukan,String)

Menentukan sistem operasi yang akan digunakan untuk cluster. Nilai yang didukung adalah `linux2`,`centos7`,`ubuntu2204`,`ubuntu2004`,`rhel8`,`rocky8`,`rhel9`,`rocky9`.

**Note**

RedHat Enterprise Linux 8.7 (`rhel8`) ditambahkan mulai AWS ParallelCluster versi 3.6.0. Jika Anda mengonfigurasi klaster untuk digunakan `rhel`, biaya sesuai permintaan untuk jenis instans apa pun lebih tinggi daripada saat Anda mengonfigurasi klaster untuk menggunakan sistem operasi lain yang didukung. Untuk informasi selengkapnya tentang harga, lihat [Harga Sesuai Permintaan](#) dan [Bagaimana Red Hat Enterprise Linux di Amazon EC2 ditawarkan](#) dan diberi harga? .

RedHat Enterprise Linux 9 (`rhel9`) ditambahkan mulai AWS ParallelCluster versi 3.9.0.

Selain spesifik yang Wilayah AWS disebutkan dalam tabel berikut yang tidak mendukung `centos7`. Semua Wilayah AWS komersial lainnya mendukung semua sistem operasi berikut.



| Partisi (Wilayah AWS)                                       | <b>alinux2</b> | <b>centos7</b> | <b>ubuntu2204 dan ubuntu2004</b> | <b>rhel8</b> | <b>rhel9</b> |
|-------------------------------------------------------------|----------------|----------------|----------------------------------|--------------|--------------|
| Komersil (Semua Wilayah AWS tidak disebutkan secara khusus) | True           | Benar          | Benar                            | Benar        | True         |
| AWS GovCloud (AS-Timur) (us-gov-east-1 )                    | True           | Salah          | Benar                            | Benar        | True         |
| AWS GovCloud (AS-Barat) (us-gov-west-1 )                    | True           | Salah          | Benar                            | Benar        | True         |
| China (Beijing) (cn-north-1 )                               | True           | Salah          | Benar                            | Benar        | True         |
| China (Ningxia) (cn-northwest-1 )                           | True           | Salah          | Benar                            | Benar        | True         |

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

#### Note

AWS ParallelCluster 3.8.0 mendukung Rocky Linux 8, tetapi AMI Rocky Linux 8 yang sudah dibuat sebelumnya (untuk arsitektur x86 dan ARM) tidak tersedia. AWS ParallelCluster 3.8.0 mendukung pembuatan cluster dengan Rocky Linux 8 menggunakan AMI khusus. Untuk informasi lebih lanjut lihat [Pertimbangan Sistem Operasi](#). AWS ParallelCluster 3.9.0 mendukung Rocky Linux 9, tetapi AMI Rocky Linux 9 yang sudah dibuat sebelumnya (untuk arsitektur x86 dan ARM) tidak tersedia. AWS ParallelCluster 3.9.0 mendukung pembuatan cluster dengan Rocky Linux 9 menggunakan AMI khusus. Untuk informasi lebih lanjut lihat [Operating System Consideratoinis](#).

## CustomAmi(Opsional,String)

Menentukan ID AMI kustom yang akan digunakan untuk node head dan compute, bukan AMI default. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Kustomisasi AMI](#).

Jika AMI kustom memerlukan izin tambahan untuk peluncurannya, izin ini harus ditambahkan ke kebijakan pengguna dan node kepala.

Misalnya, jika AMI kustom memiliki snapshot terenkripsi yang terkait dengannya, kebijakan tambahan berikut diperlukan baik dalam kebijakan pengguna maupun node kepala:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

Untuk membangun AMI kustom RedHat Enterprise Linux, Anda harus mengkonfigurasi OS untuk menginstal paket yang disediakan oleh repositori RHUI (AWS):`rhel-<version>-baseos-rhui-rpms`, `rhel-<version>-appstream-rhui-rpms` dan `codeready-builder-for-rhel-<version>-rhui-rpms` Selain itu, repositori pada AMI khusus harus berisi `kernel-devel` paket pada versi yang sama dengan versi kernel yang sedang berjalan. `kernel`.

Keterbatasan yang diketahui:

- Hanya RHEL 8.2 dan versi yang lebih baru yang mendukung FSx for Lustre.
- RHEL 8.7 kernel versi 4.18.0-425.3.1.el8 tidak mendukung FSx for Lustre.
- Hanya RHEL 8.4 dan versi yang lebih baru yang mendukung EFA.

Untuk memecahkan masalah peringatan validasi AMI kustom, lihat. [Memecahkan masalah AMI khusus](#)

[Kebijakan pembaruan](#): Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## Bagian **HeadNode**

(Wajib) Menentukan konfigurasi untuk node kepala.

### HeadNode:

InstanceType: *string*

### Networking:

SubnetId: *string*

ElasticIp: *string/boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

Proxy:

HttpProxyAddress: *string*

DisableSimultaneousMultithreading: *boolean*

Ssh:

KeyName: *string*

AllowedIps: *string*

LocalStorage:

RootVolume:

Size: *integer*

Encrypted: *boolean*

VolumeType: *string*

Iops: *integer*

Throughput: *integer*

DeleteOnTermination: *boolean*

EphemeralVolume:

MountDir: *string*

SharedStorageType: *string*

Dcv:

Enabled: *boolean*

Port: *integer*

AllowedIps: *string*

CustomActions:

OnNodeStart:

Sequence:

- Script: *string*

```

 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 Iam:
 InstanceRole: string
 InstanceProfile: string
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
 Ims:
 Secured: boolean
 Image:
 CustomAmi: string

```

## HeadNodeproperti

InstanceType(Diperlukan,String)

Menentukan jenis contoh untuk node kepala.

Menentukan jenis instans Amazon EC2 yang digunakan untuk node kepala. Arsitektur tipe instance harus sama dengan arsitektur yang digunakan untuk Slurm [InstanceType](#) pengaturan AWS Batch [InstanceType](#) atau.

**Note**

AWS ParallelCluster tidak mendukung jenis instance berikut untuk HeadNode pengaturan.

- hpc6id

Jika Anda menentukan jenis instans p4d atau jenis instance lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan, Anda harus mengatur [ElasticIp](#) `true` untuk menyediakan akses publik. AWS IP publik hanya dapat ditetapkan ke instance yang diluncurkan dengan antarmuka jaringan tunggal. Untuk kasus ini, kami menyarankan Anda menggunakan [gateway NAT](#) untuk menyediakan akses publik ke node komputasi cluster. Untuk informasi selengkapnya, lihat [Menetapkan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`DisableSimultaneousMultithreading`(Opsional, Boolean)

Jika `true`, nonaktifkan hyperthreading pada node kepala. Nilai default-nya adalah `false`.

Tidak semua tipe instance dapat menonaktifkan hyperthreading. Untuk daftar jenis instans yang mendukung penonaktifan hyperthreading, lihat [inti CPU dan thread untuk setiap inti CPU per jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`SharedStorageType`(Opsional, String)

Menentukan jenis penyimpanan yang digunakan untuk data bersama secara internal. Data yang dibagikan secara internal mencakup data yang AWS ParallelCluster digunakan untuk mengelola klaster dan default yang dibagikan `/home` jika tidak ditentukan dalam direktori [Bagian SharedStorage](#) `as Mount` untuk memasang volume sistem file bersama. Untuk detail selengkapnya tentang data bersama internal, lihat [AWS ParallelCluster Direktori internal](#).

Jika `Ebs`, yang merupakan tipe penyimpanan default, node kepala akan mengeksport sebagian volume akarnya sebagai direktori bersama untuk node komputasi dan node login menggunakan NFS.

Jika Efs, Parallelcluster akan membuat sistem file EFS untuk digunakan untuk data internal bersama dan. /home

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

#### Note

Saat kluster keluar, tipe penyimpanan EBS dapat menghadirkan hambatan kinerja saat node kepala berbagi data dari volume root dengan node komputasi menggunakan ekspor NFS. Dengan menggunakan EFS, Anda dapat menghindari ekspor NFS saat kluster Anda mengurangi skala dan menghindari kemacetan kinerja yang terkait dengannya. Disarankan untuk memilih EBS untuk potensi baca/tulis maksimal untuk file kecil dan proses instalasi. Pilih EFS untuk skala.

## Networking

(Wajib) Mendefinisikan konfigurasi jaringan untuk node kepala.

### Networking:

SubnetId: *string*

ElasticIp: *string/boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

Proxy:

HttpProxyAddress: *string*

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Networkingproperti

SubnetId(Diperlukan,String)

Menentukan ID dari subnet yang ada di mana untuk menyediakan node kepala.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## ElasticIp(Opsional,String)

Membuat atau menetapkan alamat IP Elastis ke node kepala. Nilai yang didukung adalah `true`, `false`, atau ID dari alamat IP Elastis yang ada. Nilai default-nya `false`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## SecurityGroups(Opsional,[String])

Daftar id grup keamanan Amazon VPC yang akan digunakan untuk node kepala. Ini menggantikan grup keamanan yang AWS ParallelCluster dibuat jika properti ini tidak disertakan.

Verifikasi bahwa grup keamanan dikonfigurasi dengan benar untuk [SharedStorage](#) sistem Anda.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## AdditionalSecurityGroups(Opsional,[String])

Daftar id grup keamanan Amazon VPC tambahan yang akan digunakan untuk node kepala.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## Proxy(Opsional)

Menentukan pengaturan proxy untuk node kepala.

[Proxy:](#)

[HttpProxyAddress](#): *string*

## HttpProxyAddress(Opsional,String)

Mendefinisikan server proxy HTTP atau HTTPS, biasanya `https://x.x.x.x:8080`.

Tidak ada nilai default.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Ssh

(Opsional) Mendefinisikan konfigurasi untuk akses SSH ke node kepala.

[Ssh:](#)

```
KeyName: string
AllowedIps: string
```

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## Sshproperty

KeyName(Opsional,String)

Menamai key pair Amazon EC2 yang ada untuk mengaktifkan akses SSH ke node kepala.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

AllowedIps(Opsional,String)

Menentukan rentang IP berformat CIDR atau id daftar awalan untuk koneksi SSH ke node kepala.  
Nilai default-nya 0.0.0.0/0.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## LocalStorage

(Opsional) Mendefinisikan konfigurasi penyimpanan lokal untuk node kepala.

```
LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 DeleteOnTermination: boolean
 EphemeralVolume:
 MountDir: string
```

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## LocalStorageproperty

RootVolume(Diperlukan)

Menentukan penyimpanan volume root untuk node kepala.



**RootVolume:****Size:** *integer***Encrypted:** *boolean***VolumeType:** *string***Iops:** *integer***Throughput:** *integer***DeleteOnTermination:** *boolean*

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**Size(Opsional,Integer)**

Menentukan ukuran volume akar simpul kepala di gibibytes (GiB). Ukuran default berasal dari AMI. Menggunakan ukuran yang berbeda mengharuskan AMI mendukung `growroot`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**Encrypted(Opsional,Boolean)**

Menentukan apakah volume root dienkrpsi. Nilai default-nya adalah `true`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**VolumeType(Opsional,String)**

Menentukan jenis [volume Amazon EBS](#). Nilai yang didukung adalah `gp2gp3`, `io1`, `io2`, `sc1`, `st1`, dan `standard`. Nilai default-nya adalah `gp3`.

Untuk informasi selengkapnya, lihat [Jenis volume Amazon EBS](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**Iops(Opsional,Integer)**

Mendefinisikan jumlah IOPS untuk `io1`, `io2`, dan `gp3` jenis volume.

Nilai default, nilai yang didukung, dan `volume_iops` `volume_size` rasio bervariasi menurut `VolumeType` dan `Size`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`VolumeType = io1`

Default Iops = 100

Nilai yang didukung Iops = 100—64000 †

SizeRasio maksimum Iops = 50 IOPS per GiB. 5000 IOPS membutuhkan Size setidaknya 100 GiB.

VolumeType = io2

Default Iops = 100

Nilai yang didukung Iops = 100—64000 (256000 untuk io2 volume Block Express) †

SizeRasio maksimum Iops = 500 IOPS per GiB. 5000 IOPS membutuhkan Size minimal 10 GiB.

VolumeType = gp3

Default Iops = 3000

Nilai yang didukung Iops = 3000—16000

SizeRasio maksimum Iops = 500 IOPS per GiB. 5000 IOPS membutuhkan Size minimal 10 GiB.

† IOPS maksimum dijamin hanya pada [Instans yang dibangun di atas Sistem Nitro yang](#) disediakan dengan lebih dari 32.000 IOPS. Instans lain menjamin hingga 32.000 IOPS. io1Volume yang lebih lama mungkin tidak mencapai performa penuh kecuali Anda [memodifikasi volume](#). io2 Volume Block Express mendukung Iops nilai hingga 256000 pada jenis R5b instans. Untuk informasi selengkapnya, lihat [io2 Memblokir volume Express](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

Throughput(Opsional,Integer)

Mendefinisikan throughput untuk tipe gp3 volume, dalam MiB/s. Pengaturan ini hanya valid jika VolumeType adalah gp3. Nilai default-nya adalah 125. Nilai yang didukung: 125—1000 MiB/s

Rasio Throughput to Iops bisa tidak lebih dari 0,25. Throughput maksimum 1000 MiB/s mengharuskan Iops pengaturan setidaknya 4000.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## DeleteOnTermination(Opsional,Boolean)

Menentukan apakah volume root harus dihapus ketika node kepala dihentikan. Nilai default-nya adalah `true`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## EphemeralVolume(Opsional)

Menentukan rincian untuk setiap volume toko contoh. Untuk informasi selengkapnya, lihat [Volume penyimpanan instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

```
EphemeralVolume:
MountDir: string
```

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## MountDir(Opsional,String)

Menentukan direktori mount untuk volume toko contoh. Nilai default-nya `/scratch`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Dcv

(Opsional) Mendefinisikan pengaturan konfigurasi untuk server NICE DCV yang berjalan di node kepala.

Untuk informasi selengkapnya, lihat [Connect ke node kepala melalui NICE DCV](#).

```
Dcv:
Enabled: boolean
Port: integer
AllowedIps: string
```

### Important

Secara default, pengaturan port NICE DCV terbuka untuk semua AWS ParallelCluster alamat IPv4. Namun, Anda dapat terhubung ke port NICE DCV hanya jika Anda memiliki URL untuk sesi NICE DCV dan terhubung ke sesi NICE DCV dalam waktu 30 detik sejak URL

dikembalikan. `pcluster dcv-connect` Gunakan `AllowedIps` pengaturan untuk lebih membatasi akses ke port NICE DCV dengan rentang IP berformat CIDR, dan gunakan `Port` pengaturan untuk mengatur port yang tidak standar.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Dcvproperty

`Enabled`(Diperlukan, Boolean)

Menentukan apakah NICE DCV diaktifkan pada node kepala. Nilai default-nya adalah `false`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### Note

NICE DCV secara otomatis menghasilkan sertifikat yang ditandatangani sendiri yang digunakan untuk mengamankan lalu lintas antara klien NICE DCV dan server NICE DCV yang berjalan di node kepala. Untuk mengonfigurasi sertifikat Anda sendiri, lihat [Sertifikat HTTPS NICE DCV](#).

`Port`(Opsional, Integer)

Menentukan port untuk NICE DCV. Nilai default-nya adalah 8443.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`AllowedIps`(Opsional, Direkomendasikan, String)

Menentukan rentang IP berformat CIDR untuk koneksi ke NICE DCV. Pengaturan ini hanya digunakan saat AWS ParallelCluster membuat grup keamanan. Nilai defaultnya adalah `0.0.0.0/0`, yang memungkinkan akses dari alamat internet apa pun.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## CustomActions

(Opsional) Menentukan skrip kustom untuk berjalan pada node kepala.

```
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
```

## CustomActions properti

### OnNodeStart(Opsional)

Menentukan skrip tunggal atau urutan skrip untuk dijalankan pada node kepala sebelum tindakan bootstrap penyebaran node dimulai. Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

### Sequence(Opsional)

Daftar skrip untuk dijalankan. AWS ParallelCluster menjalankan skrip dalam urutan yang sama seperti yang tercantum dalam file konfigurasi, dimulai dengan yang pertama.

### Script(Diperlukan,String)

Menentukan file yang akan digunakan. Jalur file dapat dimulai dengan `https://` atau `s3://`.

Args(Opsional,[String])

Daftar argumen untuk diteruskan ke skrip.

Script(Diperlukan,String)

Menentukan file yang akan digunakan untuk skrip tunggal. Jalur file dapat dimulai dengan `https://` atau `ataus3://`.

Args(Opsional,[String])

Daftar argumen untuk diteruskan ke skrip tunggal.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

OnNodeConfigured(Opsional)

Menentukan skrip tunggal atau urutan skrip untuk dijalankan pada node kepala setelah tindakan bootstrap node selesai. Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

Sequence(Opsional)

Menentukan daftar skrip untuk menjalankan.

Script(Diperlukan,String)

Menentukan file yang akan digunakan. Jalur file dapat dimulai dengan `https://` atau `ataus3://`.

Args(Opsional,[String])

Daftar argumen untuk diteruskan ke skrip.

Script(Diperlukan,String)

Menentukan file yang akan digunakan untuk skrip tunggal. Jalur file dapat dimulai dengan `https://` atau `ataus3://`.

Args(Opsional,[String])

Daftar argumen untuk diteruskan ke skrip tunggal.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

OnNodeUpdated(Opsional)

Menentukan skrip tunggal atau urutan skrip untuk berjalan pada node kepala setelah tindakan pembaruan node selesai. Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

## Sequence(Optional)

Menentukan daftar skrip untuk menjalankan.

`Script(Diperlukan,String)`

Menentukan file yang akan digunakan. Jalur file dapat dimulai dengan `https://` atau `s3://`.

`Args(Optional,[String])`

Daftar argumen untuk diteruskan ke skrip.

`Script(Diperlukan,String)`

Menentukan file yang akan digunakan untuk skrip tunggal. Jalur file dapat dimulai dengan `https://` atau `s3://`.

`Args(Optional,[String])`

Daftar argumen untuk diteruskan ke skrip tunggal.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### Note

`OnNodeUpdate` ditambahkan dimulai dengan AWS ParallelCluster 3.4.0.

`Sequenced` ditambahkan dimulai dengan AWS ParallelCluster versi 3.6.0. Saat Anda menentukan `Sequence`, Anda dapat membuat daftar beberapa skrip untuk tindakan kustom. AWS ParallelCluster terus mendukung konfigurasi tindakan khusus dengan satu skrip, tanpa menyertakan `Sequence`.

AWS ParallelCluster tidak mendukung termasuk skrip tunggal dan `Sequence` untuk tindakan kustom yang sama.

## Iam

(Optional) Menentukan peran instance atau profil instance yang akan digunakan pada node kepala untuk mengganti peran instance default atau profil instance untuk cluster.

`Iam:`

`InstanceRole: string`

```
InstanceProfile: string
S3Access:
- BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
AdditionalIamPolicies:
- Policy: string
```

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## Iamproperti

### InstanceProfile(Optional,String)

Menentukan profil contoh untuk mengganti profil instance node kepala default. Anda tidak dapat menentukan InstanceProfile dan InstanceRole sekaligus. Formatnya adalah `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

Jika ini ditentukan, AdditionalIamPolicies pengaturan S3Access dan tidak dapat ditentukan.

Kami menyarankan Anda menentukan salah satu atau kedua AdditionalIamPolicies pengaturan S3Access dan karena fitur yang ditambahkan AWS ParallelCluster sering memerlukan izin baru.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### InstanceRole(Optional,String)

Menentukan peran instance untuk mengganti peran instance head node default. Anda tidak dapat menentukan InstanceProfile dan InstanceRole sekaligus. Formatnya adalah `arn:Partition:iam::Account:role/RoleName`.

Jika ini ditentukan, AdditionalIamPolicies pengaturan S3Access dan tidak dapat ditentukan.

Kami menyarankan Anda menentukan salah satu atau kedua AdditionalIamPolicies pengaturan S3Access dan karena fitur yang ditambahkan AWS ParallelCluster sering memerlukan izin baru.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)



## S3Access

### S3Access(Opsional)

Menentukan ember. Ini digunakan untuk membuat kebijakan untuk memberikan akses yang ditentukan ke bucket.

Jika ini ditentukan, InstanceRole pengaturan InstanceProfile dan tidak dapat ditentukan.

Kami menyarankan Anda menentukan salah satu atau kedua AdditionalIamPolicies pengaturan S3Access dan karena fitur yang ditambahkan AWS ParallelCluster sering memerlukan izin baru.

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

#### BucketName(Diperlukan,String)

Nama ember.

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

#### KeyName(Opsional,String)

Kunci untuk ember. Nilai defaultnya adalah "\*".

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

#### EnableWriteAccess(Opsional,Boolean)

Menunjukkan apakah akses tulis diaktifkan untuk bucket. Nilai default-nya adalah false.

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

## AdditionalIamPolicies

### AdditionalIamPolicies(Opsional)

Menentukan daftar Nama Sumber Daya Amazon (ARN) kebijakan IAM untuk Amazon EC2. Daftar ini dilampirkan ke peran root yang digunakan untuk node kepala selain izin yang diperlukan oleh AWS ParallelCluster.

Nama kebijakan IAM dan ARN-nya berbeda. Nama tidak bisa digunakan.

Jika ini ditentukan, InstanceRole pengaturan InstanceProfile dan tidak dapat ditentukan.

Kami menyarankan Anda menggunakan AdditionalIamPolicies karena AdditionalIamPolicies ditambahkan ke izin yang AWS ParallelCluster diperlukan, dan InstanceRole harus menyertakan semua izin yang diperlukan. Izin yang diperlukan sering berubah dari rilis ke rilis saat fitur ditambahkan.

Tidak ada nilai default.

AdditionalIamPolicies:

- Policy: *string*

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

Policy(Opsional,[String])

Daftar kebijakan IAM.

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

## Imsds

(Opsional) Menentukan properti misalnya layanan metadata (IMDS). Untuk informasi selengkapnya, lihat [Cara kerja layanan metadata instans versi 2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Imsds:

Secured: *boolean*

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## Imsdsproperti

Secured(Opsional,Boolean)

Jika `true`, membatasi akses ke IMDS node kepala (dan kredensyal profil instance) ke subset pengguna super.

Jika `false`, setiap pengguna di node kepala memiliki akses ke IMDS node kepala.

Pengguna berikut diizinkan mengakses IMDS node kepala:

- pengguna root
- pengguna administratif cluster (secara `pc-cluster-admin` default)
- pengguna default khusus sistem operasi (`ec2-user` di Amazon Linux 2 dan RedHat, `ubuntu` di Ubuntu 18.04, `centos` di CentOS 7)

Nilai default-nya `true`.

`defaultPengguna` bertanggung jawab untuk memastikan kluster memiliki izin yang dibutuhkan untuk berinteraksi dengan AWS sumber daya. Jika Anda menonaktifkan akses IMDS default pengguna, tidak AWS ParallelCluster dapat mengelola node komputasi dan berhenti bekerja. Jangan menonaktifkan akses IMDS default pengguna.

Ketika pengguna diberikan akses ke IMDS node kepala, mereka dapat menggunakan izin yang disertakan dalam profil [instance node kepala](#). Misalnya, mereka dapat menggunakan izin ini untuk meluncurkan instans EC2 atau untuk membaca kata sandi untuk domain AD yang cluster dikonfigurasi untuk digunakan untuk otentikasi.

Untuk membatasi akses IMDS, AWS ParallelCluster mengelola rantai `iptables`

Pengguna cluster dengan `sudo` akses dapat secara selektif mengaktifkan atau menonaktifkan akses ke IMDS head node untuk pengguna individu lainnya, termasuk default pengguna, dengan menjalankan perintah:

```
$ sudo /opt/parallelcluster/scripts/imds/imds-access.sh --allow <USERNAME>
```

Anda dapat menonaktifkan akses IMDS pengguna dengan `--deny` opsi untuk perintah ini.

Jika Anda tanpa sadar menonaktifkan akses IMDS default pengguna, Anda dapat mengembalikan izin dengan menggunakan opsi `--allow`

#### Note

Setiap kustomisasi `iptables` atau `ip6tables` aturan dapat mengganggu mekanisme yang digunakan untuk membatasi akses IMDS pada node kepala.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Image

(Opsional) Mendefinisikan gambar kustom untuk node kepala.

```
Image:
 CustomAmi: string
```

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Imageproperti

CustomAmi(Opsional,String)

Menentukan ID AMI kustom yang akan digunakan untuk node kepala, bukan AMI default. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Kustomisasi AMI](#).

Jika AMI kustom memerlukan izin tambahan untuk peluncurannya, izin ini harus ditambahkan ke kebijakan pengguna dan node kepala.

Misalnya, jika AMI kustom memiliki snapshot terenkripsi yang terkait dengannya, kebijakan tambahan berikut diperlukan baik dalam kebijakan pengguna maupun node kepala:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

Untuk memecahkan masalah peringatan validasi AMI kustom, lihat [Memecahkan masalah AMI khusus](#)

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## Bagian **Scheduling**

(Wajib) Mendefinisikan penjadwal pekerjaan yang digunakan dalam kluster dan contoh komputasi yang dikelola oleh penjadwal pekerjaan. Anda dapat menggunakan AWS Batch penjadwal Slurm atau penjadwal. Masing-masing mendukung serangkaian pengaturan dan properti yang berbeda.

Topik

- [Schedulingproperty](#)
- [AwsBatchQueues](#)
- [SlurmQueues](#)
- [SlurmSettings](#)

### Scheduling:

Scheduler: `slurm`

ScalingStrategy: `string`

### SlurmSettings:

MungeKeySecretArn: `string`

ScaledownIdleTime: `integer`

QueueUpdateStrategy: `string`

EnableMemoryBasedScheduling: `boolean`

CustomSlurmSettings: `[dict]`

CustomSlurmSettingsIncludeFile: `string`

### Database:

Uri: `string`

UserName: `string`

PasswordSecretArn: `string`

DatabaseName: `string`

### Dns:

DisableManagedDns: `boolean`

HostedZoneId: `string`

UseEc2Hostnames: `boolean`

### SlurmQueues:

- Name: `string`

### ComputeSettings:

#### LocalStorage:

RootVolume:

Size: `integer`

```
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
EphemeralVolume:
 MountDir: string
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
CapacityType: string
AllocationStrategy: string
JobExclusiveAllocation: boolean
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
HealthChecks:
 Gpu:
 Enabled: boolean
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
 Proxy:
 HttpProxyAddress: string
ComputeResources:
 - Name: string
 InstanceType: string
 Instances:
 - InstanceType: string
 MinCount: integer
 MaxCount: integer
 DynamicNodePriority: integer
 StaticNodePriority: integer
 SpotPrice: float
 DisableSimultaneousMultithreading: boolean
```

```
SchedulableMemory: integer
HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
Iam:
 InstanceProfile: string
 InstanceRole: string
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
```

Image:CustomAmi: *string*Scheduling:Scheduler: awsbatchAwsBatchQueues:- Name: *string*CapacityType: *string*Networking:SubnetIds:- *string*AssignPublicIp: *boolean*SecurityGroups:- *string*AdditionalSecurityGroups:- *string*ComputeResources: # this maps to a Batch compute environment (initially we support only 1)- Name: *string*InstanceTypes:- *string*MinvCpus: *integer*DesiredvCpus: *integer*MaxvCpus: *integer*SpotBidPercentage: *float***Scheduling**properti**Scheduler**(Diperlukan,**String**)

Menentukan jenis scheduler yang digunakan. Nilai yang didukung adalah slurm dan awsbatch.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

**Note**

awsbatchhanya mendukung sistem alinux2 operasi dan x86\_64 platform.



## ScalingStrategy(Opsional,String)

Memungkinkan Anda memilih bagaimana skala node Slurm dinamis. Nilai yang didukung adalah `all-or-nothing`, `greedy-all-or-nothing` dan `best-effort`. Nilai defaultnya adalah `all-or-nothing`.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### Note

Strategi penskalaan hanya berlaku untuk node yang akan dilanjutkan oleh Slurm, bukan untuk node yang akhirnya sudah berjalan.

- `all-or-nothing` Strategi ini secara ketat mengikuti `all-or-nothing-approach`, yang bertujuan menghindari instance idle di akhir proses penskalaan. Ini beroperasi atas `all-or-nothing` dasar, yang berarti itu baik skala sepenuhnya atau tidak sama sekali. Ketahuilah bahwa mungkin ada biaya tambahan karena instans yang diluncurkan sementara, ketika pekerjaan membutuhkan lebih dari 500 node atau menjangkau beberapa sumber daya komputasi. Strategi ini memiliki throughput terendah di antara tiga Strategi Penskalaan yang mungkin. Waktu penskalaan tergantung pada jumlah pekerjaan yang dikirimkan per eksekusi program resume Slurm. Selain itu, Anda tidak dapat menskalakan jauh melampaui batas akun RunInstances sumber daya default per eksekusi, yaitu 1000 instance secara default. Detail selengkapnya dapat ditemukan di dokumentasi [AWS pelambatan API EC2](#)
- `greedy-all-or-nothing` Mirip dengan `all-or-nothing` strategi, ini bertujuan untuk menghindari instance idle pasca-penskalaan. Strategi ini memungkinkan penskalaan berlebih sementara selama proses penskalaan untuk mencapai throughput yang lebih tinggi daripada `all-or-nothing` pendekatan tetapi juga dilengkapi dengan batas penskalaan yang sama yaitu 1000 instance sesuai batas akun sumber daya. RunInstances
- `best-effort` Strategi ini memprioritaskan throughput yang tinggi, bahkan jika itu berarti bahwa beberapa contoh mungkin menganggur di akhir proses penskalaan. Ini mencoba untuk mengalokasikan node sebanyak yang diminta oleh pekerjaan, tetapi ada kemungkinan tidak memenuhi seluruh permintaan. Berbeda dengan strategi lainnya, pendekatan upaya terbaik dapat mengakumulasi lebih banyak contoh daripada RunInstances batas standar, dengan biaya memiliki sumber daya yang menganggur di sepanjang eksekusi proses penskalaan ganda.

Setiap strategi dirancang untuk memenuhi kebutuhan penskalaan yang berbeda, memungkinkan Anda memilih salah satu yang memenuhi persyaratan dan kendala spesifik Anda.

## AwsBatchQueues

(Opsional) Pengaturan AWS Batch antrian. Hanya satu antrian yang didukung. Jika [Scheduler](#) diatur ke `awsbatch`, bagian ini diperlukan. Untuk informasi selengkapnya tentang `awsbatch` penjadwal, lihat [penyiapan jaringan](#) dan [AWS Batch \(awsbatch\)](#).

### AwsBatchQueues:

- Name: *string*
- CapacityType: *string*
- Networking:
  - SubnetIds:
    - *string*
  - AssignPublicIp: *boolean*
  - SecurityGroups:
    - *string*
  - AdditionalSecurityGroups:
    - *string*
- ComputeResources: # this maps to a Batch compute environment (initially we support only 1)
  - Name: *string*
  - InstanceTypes:
    - *string*
  - MinvCpus: *integer*
  - DesiredvCpus: *integer*
  - MaxvCpus: *integer*
  - SpotBidPercentage: *float*

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

### AwsBatchQueuesproperti

#### Name(Diperlukan,**String**)

Nama AWS Batch antrian.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

#### CapacityType(Opsional,**String**)

Jenis sumber daya komputasi yang digunakan AWS Batch antrian. Nilai yang didukung adalah `ONDEMAND`, `SPOT` atau `CAPACITY_BLOCK`. Nilai default-nya adalah `ONDEMAND`.

**Note**

Jika disetel CapacityType ke SPOT, akun Anda harus berisi peran AWSServiceRoleForEC2Spot terkait layanan. Anda dapat membuat peran ini menggunakan AWS CLI perintah berikut.

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

Untuk informasi selengkapnya, lihat [Peran terkait layanan untuk permintaan Instans Spot](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## Networking

(Wajib) Mendefinisikan konfigurasi jaringan untuk AWS Batch antrian.

**Networking:****SubnetIds:**

- *string*

**AssignPublicIp:** *boolean*

**SecurityGroups:**

- *string*

**AdditionalSecurityGroups:**

- *string*

## Networkingproperti

### SubnetIds(Diperlukan, [String])

Menentukan ID dari subnet yang ada untuk menyediakan AWS Batch antrian di. Saat ini hanya satu subnet yang didukung.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## **AssignPublicIp**(Opsional,**String**)

Membuat atau menetapkan alamat IP publik ke node dalam AWS Batch antrian. Nilai yang didukung adalah `true` dan `false`. Default tergantung pada subnet yang Anda tentukan.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## **SecurityGroups**(Opsional,**[String]**)

Daftar grup keamanan yang digunakan AWS Batch antrian. Jika Anda tidak menentukan grup keamanan, AWS ParallelCluster buat grup keamanan baru.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## **AdditionalSecurityGroups**(Opsional,**[String]**)

Daftar grup keamanan yang digunakan AWS Batch antrian.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## **ComputeResources**

(Wajib) Mendefinisikan `ComputeResources` konfigurasi untuk AWS Batch antrian.

```
ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
- Name: string
 InstanceTypes:
 - string
 MinvCpus: integer
 DesiredvCpus: integer
 MaxvCpus: integer
 SpotBidPercentage: float
```

## **ComputeResources**properti

### **Name**(Diperlukan,**String**)

Nama lingkungan komputasi AWS Batch antrian.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## InstanceTypes(Diperlukan,[String])

Array lingkungan AWS Batch komputasi dari tipe instance. Semua tipe instance harus menggunakan x86\_64 arsitektur.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## MinvCpus(Opsional,Integer)

Jumlah minimum vCPU yang dapat digunakan oleh lingkungan AWS Batch komputasi.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## DesiredVcpus(Opsional,Integer)

Jumlah vCPU yang diinginkan di lingkungan AWS Batch komputasi. AWS Batch menyesuaikan nilai ini antara MinvCpus dan MaxvCpus berdasarkan permintaan dalam antrian pekerjaan.

[Kebijakan pembaruan: Pengaturan ini tidak dianalisis selama pembaruan.](#)

## MaxvCpus(Opsional,Integer)

Jumlah maksimum vCPU untuk lingkungan AWS Batch komputasi. Anda tidak dapat mengatur ini ke nilai yang lebih rendah dari DesiredVcpus.

[Kebijakan pembaruan: Pengaturan ini tidak dapat dikurangi selama pembaruan.](#)

## SpotBidPercentage(Opsional,Float)

Persentase maksimum harga On-Demand untuk jenis instans yang dapat dicapai oleh harga Instans Spot EC2 sebelum instans diluncurkan. Nilai defaultnya adalah 100 (100%). Rentang yang didukung adalah 1 -100.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## SlurmQueues

(Opsional) Pengaturan untuk Slurm antrian. Jika [Scheduler](#) diatur ke `slurm`, bagian ini diperlukan.

### SlurmQueues:

- Name: *string*
- ComputeSettings:
  - LocalStorage:
  - RootVolume:

```
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 EphemeralVolume:
 MountDir: string
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
CapacityType: string
AllocationStrategy: string
JobExclusiveAllocation: boolean
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
HealthChecks:
 Gpu:
 Enabled: boolean
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
 Proxy:
 HttpProxyAddress: string
ComputeResources:
 - Name: string
 InstanceType: string
 Instances:
 - InstanceType: string
 MinCount: integer
 MaxCount: integer
 DynamicNodePriority: integer
 StaticNodePriority: integer
 SpotPrice: float
```

```
DisableSimultaneousMultithreading: boolean
SchedulableMemory: integer
HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
Iam:
 InstanceProfile: string
 InstanceRole: string
S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
AdditionalIamPolicies:
```

```
- Policy: string
Image:
 CustomAmi: string
```

Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, nilai baru dapat ditambahkan selama pembaruan atau armada komputasi harus dihentikan saat menghapus nilai yang ada.

## SlurmQueuesproperti

### Name(Diperlukan,String)

Nama Slurm antrian.

#### Note

Ukuran cluster dapat berubah selama pembaruan. Untuk informasi selengkapnya, lihat [Ukuran dan pembaruan kapasitas cluster](#)

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## CapacityReservationTarget

#### Note

CapacityReservationTargetditambahkan dengan AWS ParallelCluster versi 3.3.0.

CapacityReservationTarget:

```
CapacityReservationId: string
CapacityReservationResourceGroupArn: string
```

Menentukan reservasi kapasitas Sesuai Permintaan untuk sumber daya komputasi antrian.

### CapacityReservationId(Opsional,String)


ID reservasi kapasitas yang ada untuk menargetkan sumber daya komputasi antrian. Id dapat merujuk ke [ODCR](#) atau [Blok Kapasitas untuk ML](#).

Reservasi harus menggunakan platform yang sama dengan yang digunakan instans.

Misalnya, jika instans Anda berjalanrhe18, reservasi kapasitas Anda harus berjalan di



platform Red Hat Enterprise Linux. Untuk informasi selengkapnya, lihat [Platform yang didukung](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

 Note

Jika Anda menyertakan [Instances](#) dalam konfigurasi cluster, Anda harus mengecualikan `CapacityReservationId` pengaturan tingkat antrian ini dari konfigurasi.

### **CapacityReservationResourceGroupArn**(Opsional,**String**)

Nama Sumber Daya Amazon (ARN) dari grup sumber daya yang berfungsi sebagai grup reservasi kapasitas terkait layanan untuk sumber daya komputasi antrian. AWS ParallelCluster mengidentifikasi dan menggunakan reservasi kapasitas yang paling tepat dari kelompok sumber daya berdasarkan kondisi berikut:

- Jika `PlacementGroup` diaktifkan di [SlurmQueues/Networking](#) atau [SlurmQueuesComputeResources/Networking](#), AWS ParallelCluster pilih grup sumber daya yang menargetkan jenis instance dan `PlacementGroup` untuk sumber daya komputasi, jika sumber daya komputasi ada.

`PlacementGroup` harus menargetkan salah satu jenis instance yang didefinisikan dalam [ComputeResources](#).

- Jika `PlacementGroup` tidak diaktifkan di [SlurmQueues/Networking](#) atau [SlurmQueuesComputeResources/Networking](#), AWS ParallelCluster pilih grup sumber daya yang hanya menargetkan jenis instance sumber daya komputasi, jika sumber daya komputasi ada.

Grup sumber daya harus memiliki setidaknya satu ODCR untuk setiap jenis instans yang dicadangkan di Availability Zone di semua sumber daya komputasi dan Availability Zone antrian. Untuk informasi selengkapnya, lihat [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#).

Untuk informasi selengkapnya tentang beberapa persyaratan konfigurasi subnet, lihat [Networking/SubnetIds](#).

**Note**

Beberapa Availability Zones ditambahkan dalam AWS ParallelCluster versi 3.4.0.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**CapacityType**(Opsional,**String**)

Jenis sumber daya komputasi yang digunakan Slurm antrian. Nilai yang didukung adalah ONDEMAND atau SPOT. Nilai default-nya adalah ONDEMAND.

**Note**

Jika Anda menyetel CapacityType ke SPOT, akun Anda harus memiliki peran AWSServiceRoleForEC2Spot terkait layanan. Anda dapat membuat peran ini dengan menggunakan AWS CLI perintah berikut.

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

Untuk informasi selengkapnya, lihat [Peran terkait layanan untuk permintaan Instans Spot](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**AllocationStrategy**(Opsional,**String**)

Tentukan strategi alokasi untuk semua sumber daya komputasi yang ditentukan dalam.

[Instances](#)

Nilai yang valid: lowest-price | capacity-optimized

Default: lowest-price

**lowest-price**

- Jika Anda menggunakan CapacityType = ONDEMAND, Armada EC2 menggunakan harga untuk menentukan pesanan dan meluncurkan instans harga terendah terlebih dahulu.

- Jika Anda menggunakannya `CapacityType = SPOT`, Armada EC2 meluncurkan instans dari kumpulan Instans Spot harga terendah yang memiliki kapasitas yang tersedia. Jika kolam kehabisan kapasitas sebelum memenuhi kapasitas yang Anda butuhkan, Armada EC2 memenuhi permintaan Anda dengan meluncurkan instans untuk Anda. Secara khusus, Armada EC2 meluncurkan instans dari kumpulan Instans Spot harga terendah yang memiliki kapasitas yang tersedia. Armada EC2 mungkin meluncurkan Instans Spot dari beberapa kumpulan yang berbeda.
- Jika Anda mengatur `CapacityType = CAPACITY_BLOCK`, tidak ada strategi alokasi, sehingga `AllocationStrategy` parameter tidak dapat dikonfigurasi.

### **capacity-optimized**

- Jika Anda mengatur `CapacityType = ONDEMAND`, `capacity-optimized` tidak tersedia.
- Jika Anda menyetel `CapacityType = SPOT`, EC2 Fleet meluncurkan instans dari kumpulan Instans Spot dengan kapasitas optimal untuk jumlah instans yang akan diluncurkan.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau `QueueUpdateStrategy` harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

#### Note

`AllocationStrategy` didukung mulai AWS ParallelCluster versi 3.3.0.

### **JobExclusiveAllocation**(Opsional, **String**)

Jika diatur ke `true`, `OverSubscribe` bendera Slurm partisi diatur ke `EXCLUSIVE`. Ketika `OverSubscribe = EXCLUSIVE`, pekerjaan di partisi memiliki akses eksklusif ke semua node yang dialokasikan. Untuk informasi selengkapnya, lihat [EKSKLUSIF](#) dalam Slurm dokumentasi.

Nilai yang valid: `true` | `false`

Default: `false`

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

#### Note

`JobExclusiveAllocation` didukung mulai AWS ParallelCluster versi 3.7.0.

## **CustomSlurmSettings**(Opsional,**Dict**)

Mendefinisikan pengaturan konfigurasi Slurm partisi kustom (antrian).

Menentukan kamus parameter Slurm konfigurasi kustom pasangan kunci-nilai yang berlaku untuk antrian (partisi).

Setiap pasangan kunci-nilai yang terpisah `Param1: Value1`, seperti, ditambahkan secara terpisah ke akhir baris konfigurasi Slurm partisi dalam format. `Param1=Value1`

Anda hanya dapat menentukan parameter Slurm konfigurasi yang tidak dicantumkan dalam daftar penolakan. `CustomSlurmSettings` Untuk informasi tentang parameter Slurm konfigurasi daftar penolakan, lihat. [Terdaftar penolakanSlurmparameter konfigurasi untukCustomSlurmSettings](#)

AWS ParallelCluster hanya memeriksa apakah parameter ada dalam daftar penolakan. AWS ParallelCluster tidak memvalidasi sintaks atau Slurm semantik parameter konfigurasi kustom Anda. Anda bertanggung jawab untuk memvalidasi parameter Slurm konfigurasi kustom Anda. Parameter Slurm konfigurasi kustom yang tidak valid dapat menyebabkan kegagalan Slurm daemon yang dapat menyebabkan kegagalan pembuatan dan pembaruan cluster.

Untuk informasi selengkapnya tentang cara menentukan parameter Slurm konfigurasi kustom AWS ParallelCluster, lihat [Slurmkustomisasi konfigurasi](#).

Untuk informasi selengkapnya tentang parameter Slurm konfigurasi, lihat [slurm.conf](#) di dokumentasi. Slurm

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### Note

`CustomSlurmSettings` didukung dimulai dengan AWS ParallelCluster versi 3.6.0.

## **Tags**(Opsional, [String])

Daftar pasangan nilai kunci tag. [ComputeResource](#) tag mengganti tag duplikat yang ditentukan dalam [Bagian Tags](#) atau di `SlurmQueues/`. Tags

## **Key**(Opsional,**String**)

Kunci tanda.

## **Value**(Opsional,**String**)

Nilai tanda.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## **HealthChecks**(Opsional)

Tentukan pemeriksaan kesehatan node komputasi pada semua sumber daya komputasi dalam antrian.

## **Gpu**(Opsional)

Tentukan pemeriksaan kesehatan GPU pada semua sumber daya komputasi dalam antrian.

### Note

AWS ParallelCluster tidak mendukung `HealthChecks/Gpu` di node yang menggunakan sistem operasi `alinux2 ARM`. Platform ini tidak mendukung [NVIDIA Data Center GPU Manager \(DCGM\)](#).

## **Enabled**(Opsional,**Boolean**)

Apakah AWS ParallelCluster melakukan pemeriksaan kesehatan GPU pada node komputasi. Nilai default-nya `false`.

## **Gpu** perilaku pemeriksaan kesehatan

- Jika `Gpu/Enabled` disetel ke `true`, AWS ParallelCluster melakukan pemeriksaan kesehatan GPU pada sumber daya komputasi dalam antrian.
- Pemeriksaan `Gpu` kesehatan melakukan pemeriksaan kesehatan GPU pada sumber daya komputasi untuk mencegah pengajuan pekerjaan pada node dengan GPU yang terdegradasi.
- Jika node komputasi gagal dalam pemeriksaan `Gpu` kesehatan, status node komputasi berubah menjadi `DRAIN`. Pekerjaan baru tidak dimulai pada node ini. Pekerjaan yang ada berjalan sampai selesai. Setelah semua pekerjaan berjalan selesai, node komputasi berakhir jika itu adalah node dinamis, dan diganti jika itu adalah node statis.

- Durasi pemeriksaan Gpu kesehatan tergantung pada jenis instans yang dipilih, jumlah GPU dalam instance, dan jumlah target pemeriksaan Gpu kesehatan (setara dengan jumlah target GPU pekerjaan). Misalnya dengan 8 GPU, durasi tipikal kurang dari 3 menit.
- Jika pemeriksaan Gpu kesehatan berjalan pada instance yang tidak didukung, itu keluar dan pekerjaan berjalan pada node komputasi. Misalnya, jika sebuah instance tidak memiliki GPU, atau, jika instance memiliki GPU, tetapi itu bukan GPU NVIDIA, pemeriksaan kesehatan keluar dan pekerjaan berjalan pada node komputasi. Hanya GPU NVIDIA yang didukung.
- Pemeriksaan Gpu kesehatan menggunakan `dcgmi` alat untuk melakukan pemeriksaan kesehatan pada node dan mengambil langkah-langkah berikut:

Ketika pemeriksaan Gpu kesehatan dimulai di simpul:

1. Ini mendeteksi apakah `nvidia-dcgm` dan `nvidia-fabricmanager` layanan berjalan.
2. Jika layanan ini tidak berjalan, pemeriksaan Gpu kesehatan memulainya.
3. Ini mendeteksi apakah mode persistensi diaktifkan.
4. Jika mode persistensi tidak diaktifkan, pemeriksaan Gpu kesehatan mengaktifkannya.

Pada akhir pemeriksaan kesehatan, pemeriksaan Gpu kesehatan mengembalikan layanan dan sumber daya ini ke keadaan awal.

- Jika pekerjaan ditetapkan ke satu set GPU node tertentu, pemeriksaan Gpu kesehatan hanya berjalan pada set tertentu. Jika tidak, pemeriksaan Gpu kesehatan berjalan pada semua GPU di node.
- Jika node komputasi menerima 2 atau lebih permintaan pemeriksaan Gpu kesehatan pada saat yang sama, hanya pemeriksaan kesehatan pertama yang berjalan dan yang lainnya dilewati. Ini juga berlaku untuk pemeriksaan kesehatan yang menargetkan GPU node. Anda dapat memeriksa file log untuk informasi tambahan mengenai situasi ini.
- Log pemeriksaan kesehatan untuk node komputasi tertentu tersedia dalam `/var/log/parallelcluster/slurm_health_check.log` file. File ini tersedia di Amazon CloudWatch, di grup CloudWatch log cluster, tempat Anda dapat menemukan:
  - Detail tentang tindakan yang dijalankan oleh pemeriksaan Gpu kesehatan, termasuk mengaktifkan dan menonaktifkan layanan dan mode persistensi.
  - Pengidentifikasi GPU, ID serial, dan UUID.
  - Output pemeriksaan kesehatan.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**Note**

HealthChecks didukung mulai AWS ParallelCluster versi 3.6.0.

## Networking

(Wajib) Mendefinisikan konfigurasi jaringan untuk Slurm antrian.

```
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
 Proxy:
 HttpProxyAddress: string
```

Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.

### Networking properti

#### SubnetIds (Diperlukan, [String])

ID subnet yang ada tempat Anda menyediakan Slurm antrian.

Jika Anda mengonfigurasi jenis instance di [SlurmQueues/ComputeResources/InstanceType](#), Anda hanya dapat menentukan satu subnet.

Jika Anda mengonfigurasi jenis instance di [SlurmQueues/ComputeResources/Instances](#), Anda dapat menentukan subnet tunggal atau beberapa subnet.

Jika Anda menggunakan beberapa subnet, semua subnet yang ditentukan untuk antrian harus berada dalam VPC yang sama, dengan setiap subnet di Availability Zone (AZ) terpisah.


Misalnya, Anda mendefinisikan subnet-1 dan subnet-2 untuk antrian Anda.

subnet-1 dan tidak subnet-2 bisa keduanya berada di AZ-1.

subnet-1 bisa di AZ-1 dan subnet-2 bisa di AZ-2.

Jika Anda mengonfigurasi hanya satu jenis instans dan ingin menggunakan beberapa subnet, tentukan jenis instans Anda sebagai Instances pengganti. InstanceType

Misalnya, definisikan `ComputeResources/Instances/InstanceType=instance.type` bukannya `ComputeResources/InstanceType=instance.type`.

 Note

Elastic Fabric Adapter (EFA) tidak didukung pada zona ketersediaan yang berbeda.

Penggunaan beberapa Availability Zone dapat menyebabkan peningkatan latensi jaringan penyimpanan dan menambahkan biaya transfer data antar-AZ. Misalnya, ini dapat terjadi ketika sebuah instance mengakses penyimpanan file yang terletak di AZ yang berbeda. Untuk informasi selengkapnya, lihat [Transfer Data dalam hal yang sama Wilayah AWS](#).


Pembaruan cluster untuk mengubah dari penggunaan subnet tunggal ke beberapa subnet:

- Misalkan definisi subnet cluster didefinisikan dengan subnet tunggal dan sistem file FSx for Lustre yang AWS ParallelCluster dikelola. Kemudian, Anda tidak dapat memperbarui cluster ini dengan definisi subnet ID yang diperbarui secara langsung. Untuk membuat pembaruan cluster, Anda harus terlebih dahulu mengubah sistem file terkelola ke sistem file eksternal. Untuk informasi selengkapnya, lihat [Ubah penyimpanan AWS ParallelCluster terkelola ke penyimpanan eksternal](#).
- Misalkan definisi subnet cluster didefinisikan dengan subnet tunggal dan sistem file Amazon EFS eksternal jika target pemasangan EFS tidak ada untuk semua AZ untuk beberapa subnet yang ditentukan untuk ditambahkan. Kemudian, Anda tidak dapat memperbarui cluster ini dengan definisi subnet ID yang diperbarui secara langsung. Untuk membuat pembaruan cluster atau membuat cluster, Anda harus terlebih dahulu membuat semua target pemasangan untuk semua AZ untuk beberapa subnet yang ditentukan.




Availability Zone dan reservasi kapasitas cluster yang ditentukan dalam [CapacityReservationResourceGroupArn](#):

- Anda tidak dapat membuat klaster jika tidak ada tumpang tindih antara kumpulan tipe instans dan zona ketersediaan yang dicakup oleh grup sumber daya reservasi kapasitas yang ditentukan dan kumpulan tipe instans dan zona ketersediaan yang ditentukan untuk antrian.
- Anda dapat membuat klaster jika ada tumpang tindih sebagian antara kumpulan tipe instans dan zona ketersediaan yang dicakup oleh grup sumber daya reservasi kapasitas yang ditentukan dan kumpulan tipe instans dan zona ketersediaan yang ditentukan untuk antrian. AWS ParallelCluster mengirim pesan peringatan tentang tumpang tindih sebagian untuk kasus ini.
- Untuk informasi selengkapnya, lihat [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#).

 Note

Beberapa Availability Zones ditambahkan dalam AWS ParallelCluster versi 3.4.0.

 Warning

Peringatan ini berlaku untuk semua versi 3.xy sebelum AWS ParallelCluster versi 3.3.1. AWS ParallelCluster versi 3.3.1 tidak terpengaruh jika parameter ini diubah.

Untuk AWS ParallelCluster 3 versi sebelum versi 3.3.1:

Mengubah parameter ini dan memperbarui cluster membuat sistem file FSx for Lustre dikelola baru dan menghapus sistem file FSx for Lustre dikelola yang ada tanpa melestarikan data yang ada. Hal ini mengakibatkan hilangnya data. Sebelum Anda melanjutkan, pastikan Anda mencadangkan data dari sistem file FSx for Lustre yang ada jika Anda ingin menyimpan data. Untuk informasi selengkapnya, lihat [Bekerja dengan cadangan di Panduan Pengguna](#) FSx for Lustre.

Jika nilai subnet baru ditambahkan, [Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan](#).

Jika nilai subnet dihapus, [Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel](#) agar pengaturan ini diubah untuk pembaruan.

## **AssignPublicIp**(Opsional,**String**)

Membuat atau menetapkan alamat IP publik ke node dalam Slurm antrian. Nilai yang didukung adalah `true` dan `false`. Subnet yang Anda tentukan menentukan nilai default. Subnet dengan IP publik default untuk menetapkan alamat IP publik.

Jika Anda menentukan jenis `p4d` atau `hpc6id` instance, atau jenis instance lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan, Anda harus mengatur [HeadNode/Networking/ElasticIp](#) `true` untuk menyediakan akses publik. AWS IP publik hanya dapat ditetapkan ke instance yang diluncurkan dengan antarmuka jaringan tunggal. Untuk kasus ini, kami menyarankan Anda menggunakan [gateway NAT](#) untuk menyediakan akses publik ke node komputasi cluster. Dalam hal ini, atur `AssignPublicIp` ke `false`. Untuk informasi selengkapnya tentang alamat IP, lihat [Menetapkan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## **SecurityGroups**(Opsional,**[String]**)

Daftar grup keamanan yang akan digunakan untuk Slurm antrian. Jika tidak ada grup keamanan yang ditentukan, AWS ParallelCluster buat grup keamanan untuk Anda.

Verifikasi bahwa grup keamanan dikonfigurasi dengan benar untuk [SharedStorage](#) sistem Anda.

### Warning

Peringatan ini berlaku untuk semua 3. *x*. AWS ParallelCluster versi *y* sebelum versi 3.3.0. AWS ParallelCluster versi 3.3.0 tidak terpengaruh jika parameter ini diubah.

Untuk AWS ParallelCluster 3 versi sebelum versi 3.3.0:

Mengubah parameter ini dan memperbarui cluster membuat sistem file FSx for Lustre terkelola baru dan menghapus sistem file FSx for Lustre terkelola yang ada tanpa melestarikan data yang ada. Hal ini mengakibatkan hilangnya data. Pastikan untuk mencadangkan data dari sistem file FSx for Lustre yang ada jika Anda ingin menyimpan data. Untuk informasi selengkapnya, lihat [Bekerja dengan cadangan di Panduan Pengguna](#) FSx for Lustre.

**⚠ Warning**

Jika Anda mengaktifkan [EFA](#) untuk instans komputasi, pastikan instans berkemampuan EFA Anda adalah anggota grup keamanan yang memungkinkan semua lalu lintas masuk dan keluar ke dirinya sendiri.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**AdditionalSecurityGroups**(Opsional, **[String]**)

Daftar grup keamanan tambahan yang akan digunakan untuk Slurm antrian.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**PlacementGroup**(Opsional)

Menentukan pengaturan grup penempatan untuk Slurm antrian.

[PlacementGroup](#):

[Enabled](#): *boolean*

[Id](#): *string*

[Name](#): *string*

[Kebijakan pembaruan: Semua node komputasi harus dihentikan untuk penghapusan grup penempatan terkelola. Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**Enabled**(Opsional, **Boolean**)

Menunjukkan apakah grup penempatan digunakan untuk Slurm antrian. Nilai default-nya `false`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**Id**(Opsional, **String**)

Nama grup penempatan untuk grup penempatan klaster yang ada yang digunakan Slurm antrian. Pastikan untuk memberikan nama grup penempatan dan bukan ID.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## Name(Opsional,String)

Nama grup penempatan untuk grup penempatan kluster yang ada yang digunakan Slurm antrian. Pastikan untuk memberikan nama grup penempatan dan bukan ID.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategyharus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### Note

- Jika `PlacementGroup/Enabled` disetel ke `true`, tanpa `Name` atau `Id` didefinisikan, setiap sumber daya komputasi ditetapkan grup penempatan terkelolanya sendiri, kecuali `ComputeResources/Networking/PlacementGroup` didefinisikan untuk mengganti pengaturan ini.
- Dimulai dengan AWS ParallelCluster versi 3.3.0, `SlurmQueuesNetworking/PlacementGroup//Name` ditambahkan sebagai alternatif yang disukai untuk `SlurmQueues//NetworkingPlacementGroup/Id`.

`PlacementGroup/Id` dan `PlacementGroup/Name` setara. Anda dapat menggunakan salah satunya.

Jika Anda menyertakan keduanya `PlacementGroup/Id` dan `PlacementGroup/Name`, AWS ParallelCluster gagal. Anda hanya dapat memilih satu atau yang lain.

Anda tidak perlu memperbarui cluster Anda untuk menggunakan `PlacementGroup/Name`.

## Proxy(Opsional)

Menentukan pengaturan proxy untuk Slurm antrian.

```
Proxy:
 HttpProxyAddress: string
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategyharus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## HttpProxyAddress(Opsional,String)

Mendefinisikan server proxy HTTP atau HTTPS untuk Slurm antrian. Biasanya, itu `https://x.x.x.x:8080`.

Tidak ada nilai default.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## Image

(Opsional) Menentukan gambar yang akan digunakan untuk Slurm antrian. Untuk menggunakan AMI yang sama untuk semua node, gunakan [CustomAmi](#) pengaturan di [Imagebagian](#).

```
Image:
CustomAmi: string
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## ImageProperti

### CustomAmi(Opsional,String)

AMI yang akan digunakan untuk Slurm antrian alih-alih AMI default. Anda dapat menggunakan perintah `pcluster CLI` untuk melihat daftar AMI default.

#### Note

AMI harus didasarkan pada sistem operasi yang sama yang digunakan oleh node kepala.

```
pcluster list-official-images
```

Jika AMI kustom memerlukan izin tambahan untuk peluncurannya, Anda harus menambahkan izin ini ke kebijakan node kepala.

Misalnya, jika AMI kustom memiliki snapshot terenkripsi yang terkait dengannya, kebijakan tambahan berikut diperlukan dalam kebijakan simpul kepala.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>;key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

Untuk memecahkan masalah peringatan validasi AMI kustom, lihat. [Memecahkan masalah AMI khusus](#)

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## ComputeResources

(Wajib) Mendefinisikan ComputeResources konfigurasi untuk Slurm antrian.

### Note

Ukuran cluster dapat berubah selama pembaruan. Untuk informasi selengkapnya, lihat [Ukuran dan pembaruan kapasitas cluster](#)

### ComputeResources:

- Name: *string*
- InstanceType: *string*
- Instances:
  - InstanceType: *string*
  - MinCount: *integer*

```

MaxCount: integer
DynamicNodePriority: integer
StaticNodePriority: integer
SpotPrice: float
DisableSimultaneousMultithreading: boolean
SchedulableMemory: integer
HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string

```

Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, nilai baru dapat ditambahkan selama pembaruan atau armada komputasi harus dihentikan saat menghapus nilai yang ada.

### ComputeResourcesproperti

#### Name(Diperlukan,String)

Nama lingkungan komputasi Slurm antrian. Nama dapat memiliki hingga 25 karakter.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

#### InstanceType(Diperlukan,String)

Jenis instance yang digunakan dalam sumber daya Slurm komputasi ini. Semua jenis instance dalam cluster harus menggunakan arsitektur prosesor yang sama. Contoh dapat menggunakan arm64 arsitektur x86\_64 atau.

Konfigurasi cluster harus mendefinisikan salah satu [InstanceType](#) atau [Instance](#). Jika keduanya didefinisikan, AWS ParallelCluster gagal.

Ketika Anda mendefinisikan `InstanceType`, Anda tidak dapat mendefinisikan beberapa subnet. Jika Anda mengonfigurasi hanya satu jenis instance dan ingin menggunakan beberapa subnet, tentukan tipe instans Anda `Instances` daripada di `InstanceType`. Untuk informasi lebih lanjut, lihat [Networking/SubnetIds](#).

Jika Anda menentukan jenis `p4d` atau `hpc6id` instance, atau jenis instans lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan, Anda harus meluncurkan instance komputasi di subnet pribadi seperti yang dijelaskan dalam [AWS ParallelCluster menggunakan dua subnet](#) AWS IP publik hanya dapat ditetapkan ke instance yang diluncurkan dengan antarmuka jaringan tunggal. Untuk informasi selengkapnya, lihat [Menetapkan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## Instances(Diperlukan)

Menentukan daftar jenis contoh untuk sumber daya komputasi. Untuk menentukan strategi alokasi untuk daftar jenis instance, lihat [AllocationStrategy](#).

Konfigurasi cluster harus mendefinisikan salah satu [InstanceType](#) atau [Instances](#). Jika keduanya didefinisikan, AWS ParallelCluster gagal.

Untuk informasi selengkapnya, lihat [Beberapa alokasi tipe instans dengan Slurm](#).

### [Instances](#):

- `InstanceType`: *string*

### Note

[Dimulai dengan AWS ParallelCluster versi 3.7.0, `EnableMemoryBasedScheduling` dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instans di `Instans`.](#)

Untuk AWS ParallelCluster versi 3.2.0 hingga 3.6. *x*, tidak

`EnableMemoryBasedScheduling` dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instance di [Instance](#).

[Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, nilai baru dapat ditambahkan selama pembaruan atau armada komputasi harus dihentikan saat menghapus nilai yang ada.](#)



## InstanceType(Diperlukan,String)

Jenis instance untuk digunakan dalam sumber daya Slurm komputasi ini. Semua jenis instance dalam cluster harus menggunakan arsitektur prosesor yang sama, baik x86\_64 atau arm64.

Jenis instance yang tercantum di [Instances](#) harus memiliki:

- Jumlah vCPU yang sama, atau, [DisableSimultaneousMultithreading](#) jika diatur `true` ke, jumlah inti yang sama.
- Jumlah akselerator yang sama dari produsen yang sama.
- EFA didukung, jika [Efa/Enabled](#) disetel ke `true`.

Jenis instance yang tercantum dalam [Instances](#) dapat memiliki:

- Jumlah memori yang berbeda.

Dalam hal ini, memori minimum harus ditetapkan sebagai Slurm sumber daya habis pakai.

### Note

[Dimulai dengan AWS ParallelCluster versi 3.7.0, EnableMemoryBasedScheduling dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instans di Instans.](#)

Untuk AWS ParallelCluster versi 3.2.0 hingga 3.6. x, tidak [EnableMemoryBasedScheduling](#) dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instance di [Instance](#).

- Kartu jaringan yang berbeda.

Dalam hal ini, jumlah antarmuka jaringan yang dikonfigurasi untuk sumber daya komputasi ditentukan oleh jenis instance dengan jumlah kartu jaringan terkecil.

- Bandwidth jaringan yang berbeda.
- Ukuran toko instance yang berbeda.

Jika Anda menentukan jenis p4d atau hpc6id instance, atau jenis instans lain yang memiliki beberapa antarmuka jaringan atau kartu antarmuka jaringan, Anda harus meluncurkan instance komputasi di subnet pribadi seperti yang dijelaskan dalam [AWS ParallelCluster menggunakan dua subnet](#) AWS IP publik hanya dapat ditetapkan ke instance

yang diluncurkan dengan antarmuka jaringan tunggal. Untuk informasi selengkapnya, lihat [Menetapkan alamat IPv4 publik selama peluncuran instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

**Note**

Instances didukung dimulai dengan AWS ParallelCluster versi 3.3.0.

### **MinCount**(Opsional, **Integer**)

Jumlah minimum instance yang digunakan sumber daya Slurm komputasi. Default-nya adalah 0.

**Note**

Ukuran cluster dapat berubah selama pembaruan. Untuk informasi selengkapnya, lihat [Ukuran dan pembaruan kapasitas cluster](#)

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### **MaxCount**(Opsional, **Integer**)

Jumlah maksimum instance yang digunakan sumber daya Slurm komputasi. Default-nya adalah 10.

Saat menggunakan `CapacityType = CAPACITY_BLOCK`, `MaxCount` harus sama dengan `MinCount` dan lebih besar dari 0, karena semua instance bagian dari reservasi Blok Kapasitas dikelola sebagai node statis.

Pada waktu pembuatan cluster, head node menunggu semua node statis siap sebelum menandakan keberhasilan pembuatan cluster. Namun, saat menggunakan `CapacityType = CAPACITY_BLOCK`, bagian node dari sumber daya komputasi yang terkait dengan Blok Kapasitas tidak akan dipertimbangkan untuk pemeriksaan ini. Cluster akan dibuat bahkan jika tidak semua Blok Kapasitas yang dikonfigurasi aktif.

**Note**

Ukuran cluster dapat berubah selama pembaruan. Untuk informasi selengkapnya, lihat [Ukuran dan pembaruan kapasitas cluster](#)

**DynamicNodePriority**(Opsional,**Integer**)

Prioritas node dinamis dalam sumber daya komputasi antrian. Prioritas memetakan ke parameter [Weight](#) konfigurasi Slurm node untuk node dinamis sumber daya komputasi. Nilai default-nya adalah 1000.

Slurmmemprioritaskan node dengan Weight nilai terendah terlebih dahulu.

**Warning**

Penggunaan banyak Weight nilai yang berbeda dalam Slurm partisi (antrian) mungkin memperlambat laju penjadwalan pekerjaan dalam antrian.

Dalam AWS ParallelCluster versi yang lebih awal dari versi 3.7.0, node statis dan dinamis diberi bobot default yang sama yaitu 1. Dalam hal ini, Slurm mungkin memprioritaskan node dinamis idle di atas node statis idle karena skema penamaan untuk node statis dan dinamis. Ketika semuanya sama, Slurm menjadwalkan node menurut abjad berdasarkan nama.

**Note**

DynamicNodePriority ditambahkan dalam AWS ParallelCluster versi 3.7.0.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**StaticNodePriority**(Opsional,**Integer**)

Prioritas node statis dalam sumber daya komputasi antrian. Prioritas memetakan ke parameter [Weight](#) konfigurasi Slurm node untuk node statis sumber daya komputasi. Nilai default-nya adalah 1.

Slurmmemprioritaskan node dengan Weight nilai terendah terlebih dahulu.

**⚠ Warning**

Penggunaan banyak `Weight` nilai yang berbeda dalam Slurm partisi (antrian) mungkin memperlambat laju penjadwalan pekerjaan dalam antrian.

**ℹ Note**

`StaticNodePriority` ditambahkan dalam AWS ParallelCluster versi 3.7.0.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**SpotPrice**(Opsional,**Float**)

Harga maksimum yang dibayarkan untuk Instans Spot EC2 sebelum instans diluncurkan. Nilai default adalah harga On-Demand.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau `QueueUpdateStrategy` harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**DisableSimultaneousMultithreading**(Opsional,**Boolean**)

Jika `true`, multithreading pada node dalam Slurm antrian dinonaktifkan. Nilai default-nya adalah `false`.

Tidak semua tipe instance dapat menonaktifkan multithreading. Untuk daftar jenis instans yang mendukung penonaktifan multithreading, lihat [inti CPU dan thread untuk setiap inti CPU per jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)


**SchedulableMemory**(Opsional,**Integer**)

Jumlah memori di MiB yang dikonfigurasi dalam Slurm parameter `RealMemory` untuk node komputasi sumber daya komputasi. Nilai ini adalah batas atas untuk memori node yang tersedia untuk pekerjaan saat [SlurmSettings/EnableMemoryBasedScheduling](#) diaktifkan. Nilai defaultnya adalah 95 persen dari memori yang terdaftar di [Jenis Instans Amazon EC2](#) dan dikembalikan oleh Amazon EC2 API. [DescribeInstanceTypes](#) Pastikan untuk mengonversi nilai yang diberikan dalam GiB ke MiB.

Nilai yang didukung: 1-EC2Memory

EC2Memory adalah memori (dalam MiB) yang terdaftar di Jenis [Instans Amazon EC2](#) dan dikembalikan oleh Amazon EC2 API. [DescribeInstanceTypes](#) Pastikan untuk mengonversi nilai yang diberikan dalam GiB ke MiB.

Opsi ini paling relevan ketika [SlurmSettings/EnableMemoryBasedScheduling](#) diaktifkan. Untuk informasi selengkapnya, lihat [Slurmpenjadwalan berbasis memori](#).

 Note

SchedulableMemory didukung dimulai dengan AWS ParallelCluster versi 3.2.0. Dimulai dengan versi 3.2.0, secara default, AWS ParallelCluster mengonfigurasi RealMemory node Slurm komputasi hingga 95 persen memori yang dikembalikan oleh Amazon EC2 API. DescribeInstanceTypes Konfigurasi ini tidak tergantung pada nilai EnableMemoryBasedScheduling.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## HealthChecks(Opsional)


Tentukan pemeriksaan kesehatan pada sumber daya komputasi.

## Gpu(Opsional)

Tentukan pemeriksaan kesehatan GPU pada sumber daya komputasi.

## Enabled(Opsional, Boolean)

Apakah AWS ParallelCluster melakukan pemeriksaan kesehatan GPU saat menghitung sumber daya dalam antrian. Nilai default-nya false.

 Note

AWS ParallelCluster tidak mendukung HealthChecks/Gpu di node yang menggunakan sistem operasi alinux2 ARM. Platform ini tidak mendukung [NVIDIA Data Center GPU Manager \(DCGM\)](#).

## Gpuperilaku pemeriksaan kesehatan

- Jika `Gpu/Enabled` disetel ke `true`, AWS ParallelCluster lakukan pemeriksaan kesehatan GPU kesehatan pada sumber daya komputasi.
- Pemeriksaan Gpu kesehatan melakukan pemeriksaan kesehatan pada sumber daya komputasi untuk mencegah pengajuan pekerjaan pada node dengan GPU terdegradasi.
- Jika node komputasi gagal dalam pemeriksaan Gpu kesehatan, status node komputasi berubah menjadi `DRAIN`. Pekerjaan baru tidak dimulai pada node ini. Pekerjaan yang ada berjalan sampai selesai. Setelah semua pekerjaan berjalan selesai, node komputasi berakhir jika itu adalah node dinamis, dan diganti jika itu adalah node statis.
- Durasi pemeriksaan Gpu kesehatan tergantung pada jenis instans yang dipilih, jumlah GPU dalam instance, dan jumlah target pemeriksaan Gpu kesehatan (setara dengan jumlah target GPU pekerjaan). Misalnya dengan 8 GPU, durasi tipikal kurang dari 3 menit.
- Jika pemeriksaan Gpu kesehatan berjalan pada instance yang tidak didukung, itu keluar dan pekerjaan berjalan pada node komputasi. Misalnya, jika sebuah instance tidak memiliki GPU, atau, jika instance memiliki GPU, tetapi itu bukan GPU NVIDIA, pemeriksaan kesehatan keluar dan pekerjaan berjalan pada node komputasi. Hanya GPU NVIDIA yang didukung.
- Pemeriksaan Gpu kesehatan menggunakan `dcgmi` alat untuk melakukan pemeriksaan kesehatan pada node dan mengambil langkah-langkah berikut:

Ketika pemeriksaan Gpu kesehatan dimulai di simpul:

1. Ini mendeteksi apakah `nvidia-dcgm` dan `nvidia-fabricmanager` layanan berjalan.
2. Jika layanan ini tidak berjalan, pemeriksaan Gpu kesehatan memulainya.
3. Ini mendeteksi apakah mode persistensi diaktifkan.
4. Jika mode persistensi tidak diaktifkan, pemeriksaan Gpu kesehatan mengaktifkannya.

Pada akhir pemeriksaan kesehatan, pemeriksaan Gpu kesehatan mengembalikan layanan dan sumber daya ini ke keadaan awal.

- Jika pekerjaan ditetapkan ke satu set GPU node tertentu, pemeriksaan Gpu kesehatan hanya berjalan pada set tertentu. Jika tidak, pemeriksaan Gpu kesehatan berjalan pada semua GPU di node.
- Jika node komputasi menerima 2 atau lebih permintaan pemeriksaan Gpu kesehatan pada saat yang sama, hanya pemeriksaan kesehatan pertama yang berjalan dan yang lainnya dilewati. Ini juga berlaku untuk pemeriksaan kesehatan yang menargetkan GPU node. Anda dapat memeriksa file log untuk informasi tambahan mengenai situasi ini.

- Log pemeriksaan kesehatan untuk node komputasi tertentu tersedia dalam `/var/log/parallelcluster/slurm_health_check.log` file. File ini tersedia di Amazon CloudWatch, di grup CloudWatch log cluster, tempat Anda dapat menemukan:
  - Detail tentang tindakan yang dijalankan oleh pemeriksaan Gpu kesehatan, termasuk mengaktifkan dan menonaktifkan layanan dan mode persistensi.
  - Pengidentifikasi GPU, ID serial, dan UUID.
  - Output pemeriksaan kesehatan.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**Note**

HealthChecks didukung mulai AWS ParallelCluster versi 3.6.0.

## Efa(Opsional)

Menentukan pengaturan Elastic Fabric Adapter (EFA) untuk node dalam antrian. Slurm

Efa:

Enabled: *boolean*

GdrSupport: *boolean*

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## Enabled(Opsional, Boolean)

Menentukan bahwa Elastic Fabric Adapter (EFA) diaktifkan. Untuk melihat daftar instans EC2 yang mendukung EFA, lihat [Jenis instans yang didukung](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk informasi selengkapnya, lihat [Elastic Fabric Adapter](#). Kami menyarankan Anda menggunakan cluster [SlurmQueues/Networking/PlacementGroup](#) untuk meminimalkan latensi antar instance.

Nilai default-nya adalah `false`.

**Note**

Elastic Fabric Adapter (EFA) tidak didukung pada zona ketersediaan yang berbeda. Untuk informasi lebih lanjut, lihat [SubnetIds](#).

**Warning**

Jika Anda mendefinisikan grup keamanan khusus [SecurityGroups](#), pastikan instans berkemampuan EFA Anda adalah anggota grup keamanan yang memungkinkan semua lalu lintas masuk dan keluar ke dirinya sendiri.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**GdrSupport**(Opsional,**Boolean**)

(Opsional) Dimulai dengan AWS ParallelCluster versi 3.0.2, pengaturan ini tidak berpengaruh. Dukungan Elastic Fabric Adapter (EFA) untuk GPUDirect RDMA (akses memori langsung jarak jauh) selalu diaktifkan jika didukung oleh jenis instans untuk sumber daya komputasi dan sistem operasi. Slurm

**Note**

AWS ParallelCluster versi 3.0.0 hingga 3.0.1: Support untuk GPUDirect RDMA diaktifkan untuk sumber daya komputasi. Slurm Support untuk GPUDirect RDMA didukung oleh jenis instans tertentu (p4d.24xlarge) pada sistem operasi tertentu ([Osisalinux2](#),,centos7, ubuntu1804 atau). ubuntu2004 Nilai default adalah false.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**CapacityReservationTarget**

[CapacityReservationTarget](#):

[CapacityReservationId](#): *string*

[CapacityReservationResourceGroupArn](#): *string*



Menentukan reservasi kapasitas sesuai permintaan yang akan digunakan untuk sumber daya komputasi.

### **CapacityReservationId**(Opsional,**String**)

ID reservasi kapasitas yang ada untuk menargetkan sumber daya komputasi antrian. Id dapat merujuk ke [ODCR](#) atau [Blok Kapasitas untuk ML](#).

Ketika parameter ini ditentukan pada tingkat sumber daya komputasi, InstanceType adalah opsional, itu akan secara otomatis diambil dari reservasi.

### **CapacityReservationResourceGroupArn**(Opsional,**String**)

Menunjukkan Nama Sumber Daya Amazon (ARN) dari grup sumber daya yang berfungsi sebagai grup reservasi kapasitas yang ditautkan layanan untuk sumber daya komputasi. AWS ParallelCluster mengidentifikasi dan menggunakan reservasi kapasitas yang paling tepat dari grup. Grup sumber daya harus memiliki setidaknya satu ODCR untuk setiap jenis instans yang terdaftar untuk sumber daya komputasi. Untuk informasi selengkapnya, lihat [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#).

- Jika PlacementGroup diaktifkan di [SlurmQueues/Networking](#) atau [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster pilih grup sumber daya yang menargetkan jenis instance dan PlacementGroup untuk sumber daya komputasi jika ada.

PlacementGroupHarus menargetkan salah satu jenis instance yang didefinisikan dalam [ComputeResources](#).

- Jika PlacementGroup tidak diaktifkan di [SlurmQueues/Networking](#) atau [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster pilih grup sumber daya yang hanya menargetkan jenis instance sumber daya komputasi, jika ada.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategyharus disetel agar pengaturan ini diubah untuk pembaruan.](#)

#### Note

CapacityReservationTargetditambahkan dengan AWS ParallelCluster versi 3.3.0.

## Networking

[Networking](#):

**PlacementGroup:****Enabled:** *boolean***Name:** *string*

Kebijakan pembaruan: Semua node komputasi harus dihentikan untuk penghapusan grup penempatan terkelola. Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.

**PlacementGroup**(Opsional)

Menentukan pengaturan grup penempatan untuk sumber daya komputasi.

**Enabled**(Opsional,**Boolean**)

Menunjukkan apakah grup penempatan digunakan untuk sumber daya komputasi.

- Jika disetel ke `true`, tanpa Name didefinisikan, sumber daya komputasi tersebut ditetapkan grup penempatan terkelolanya sendiri, terlepas dari [PlacementGroup](#) pengaturan [SlurmQueues/Networking/](#).
- Jika disetel ke `true`, dengan Name didefinisikan, sumber daya komputasi tersebut ditetapkan grup penempatan bernama, terlepas dari `PlacementGroup` pengaturan [SlurmQueues/Networking/](#).

Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.

**Name**(Opsional,**String**)

Nama grup penempatan untuk grup penempatan kluster yang ada yang digunakan untuk sumber daya komputasi.

Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.

**Note**

- Jika keduanya `PlacementGroup/Enabled` dan `Name` tidak disetel, nilainya masing-masing default ke [PlacementGroup](#) pengaturan [SlurmQueuesNetworking/](#).
- `ComputeResources/Networking/PlacementGroup` ditambahkan dengan AWS ParallelCluster versi 3.3.0.

## **CustomSlurmSettings**(Opsional,**Dict**)

(Opsional) Mendefinisikan pengaturan konfigurasi Slurm node kustom (sumber daya komputasi).

Menentukan kamus parameter Slurm konfigurasi kustom pasangan kunci-nilai yang berlaku untuk Slurm node (sumber daya komputasi).

Setiap pasangan kunci-nilai yang terpisah `Param1: Value1`, seperti, ditambahkan secara terpisah ke akhir baris konfigurasi Slurm node dalam format. `Param1=Value1`

Anda hanya dapat menentukan parameter Slurm konfigurasi yang tidak dicantumkan dalam daftar penolakan. `CustomSlurmSettings` Untuk informasi tentang parameter Slurm konfigurasi daftar penolakan, lihat. [Terdaftar penolakanSlurmparameter konfigurasi untukCustomSlurmSettings](#)

AWS ParallelCluster hanya memeriksa apakah parameter ada dalam daftar penolakan. AWS ParallelCluster tidak memvalidasi sintaks atau Slurm semantik parameter konfigurasi kustom Anda. Anda bertanggung jawab untuk memvalidasi parameter Slurm konfigurasi kustom Anda. Parameter Slurm konfigurasi kustom yang tidak valid dapat menyebabkan kegagalan Slurm daemon yang dapat menyebabkan kegagalan pembuatan dan pembaruan cluster.

Untuk informasi selengkapnya tentang cara menentukan parameter Slurm konfigurasi kustom AWS ParallelCluster, lihat [Slurm kustomisasi konfigurasi](#).

Untuk informasi selengkapnya tentang parameter Slurm konfigurasi, lihat [slurm.conf](#) di dokumentasi. Slurm

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### Note

`CustomSlurmSettings` didukung dimulai dengan AWS ParallelCluster versi 3.6.0.

## **Tags**(Opsional, [String])

Daftar pasangan nilai kunci tag. `ComputeResource` tag mengganti tag duplikat yang ditentukan dalam [Bagian Tags](#) atau [SlurmQueues](#)/. Tags

## **Key**(Opsional,**String**)

Kunci tanda.

## Value(Opsional,String)

Nilai tanda.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## ComputeSettings

(Wajib) Mendefinisikan ComputeSettings konfigurasi untuk Slurm antrian.

### ComputeSettings properti

Menentukan properti ComputeSettings dari node dalam Slurm antrian.

```
ComputeSettings:
 LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 EphemeralVolume:
 MountDir: string
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## LocalStorage(Opsional)

Menentukan properti LocalStorage dari node dalam Slurm antrian.

```
LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 EphemeralVolume:
 MountDir: string
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### **RootVolume**(Opsional)

Menentukan rincian volume root node dalam Slurm antrian.

#### RootVolume:

Size: *integer*

Encrypted: *boolean*

VolumeType: *string*

Iops: *integer*

Throughput: *integer*

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### **Size**(Opsional,**Integer**)

Menentukan ukuran volume root dalam gibibytes (GiB) untuk node dalam antrian. Slurm Ukuran default berasal dari AMI. Menggunakan ukuran yang berbeda mengharuskan AMI mendukung `growroot`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### **Encrypted**(Opsional,**Boolean**)

Jika `true`, volume root node dalam Slurm antrian dienkripsi. Nilai default-nya adalah `false`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### **VolumeType**(Opsional,**String**)

Menentukan [jenis volume Amazon EBS](#) dari node dalam antrian. Slurm Nilai yang didukung adalah `gp2gp3`, `io1`, `io2`, `sc1`, `st1`, dan `standard`. Nilai default-nya adalah `gp3`.

Untuk informasi selengkapnya, lihat [Jenis volume Amazon EBS](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## **Iops**(Opsional,**Boolean**)

Mendefinisikan jumlah IOPS untuk `io1`, `io2`, dan `gp3` jenis volume.

Nilai default, nilai yang didukung, dan `volume_iops` `volume_size` rasio bervariasi menurut `VolumeType` dan `Size`.

### **VolumeType = io1**

Default Iops = 100

Nilai yang didukung Iops = 100—64000 †

`volume_size`Rasio maksimum `volume_iops` = 50 IOPS per GiB. 5000 IOPS membutuhkan `volume_size` setidaknya 100 GiB.

### **VolumeType = io2**

Default Iops = 100

Nilai yang didukung Iops = 100—64000 (256000 untuk `io2` volume Block Express) †

`Size`Rasio maksimum Iops = 500 IOPS per GiB. 5000 IOPS membutuhkan `Size` minimal 10 GiB.

### **VolumeType = gp3**

Default Iops = 3000

Nilai yang didukung Iops = 3000—16000 †

`Size`Rasio maksimum Iops = 500 IOPS per GiB untuk volume dengan IOPS lebih besar dari 3000.

† IOPS maksimum dijamin hanya pada [Instans yang dibangun di atas Sistem Nitro](#) yang juga disediakan dengan lebih dari 32.000 IOPS. Instans lain dapat memiliki hingga 32.000 IOPS. `io1`Volume sebelumnya mungkin tidak mencapai kinerja penuh kecuali Anda [memodifikasi volume](#). `io2` Volume Block Express mendukung `volume_iops` nilai hingga 256000 pada jenis R5b instans. Untuk informasi selengkapnya, lihat [io2Memblokir volume Express](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategyharus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## Throughput(Opsional,Integer)

Mendefinisikan throughput untuk tipe gp3 volume, dalam MiB/s. Pengaturan ini hanya valid jika VolumeType adalah gp3. Nilai default-nya adalah 125. Nilai yang didukung: 125—1000 MiB/s

Rasio Throughput to Iops bisa tidak lebih dari 0,25. Throughput maksimum 1000 MiB/s mengharuskan Iops pengaturan setidaknya 4000.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## EphemeralVolume(Opsional,Boolean)

Menentukan pengaturan untuk volume fana. Volume sementara dibuat dengan menggabungkan semua volume penyimpanan instance ke dalam volume logis tunggal yang diformat dengan sistem file. ext4 Nilai default-nya /scratch. Jika tipe instance tidak memiliki volume penyimpanan instance apa pun, tidak ada volume sementara yang dibuat. Untuk informasi selengkapnya, lihat [Volume penyimpanan instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[EphemeralVolume](#):  
[MountDir](#): *string*

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## MountDir(Opsional,String)

Direktori mount untuk volume fana untuk setiap node dalam antrian. Slurm

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## CustomActions

(Opsional) Menentukan skrip kustom untuk berjalan pada node dalam antrian. Slurm

[CustomActions](#):  
[OnNodeStart](#):  
[Sequence](#):

```
- Script: string
 Args:
 - string
Script: string
Args:
 - string
OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

## CustomActionsProperti

### OnNodeStart(Opsional,String)

Menentukan urutan skrip atau skrip tunggal untuk dijalankan pada node dalam Slurm antrian sebelum tindakan bootstrap penyebaran node dimulai. AWS ParallelCluster tidak mendukung termasuk skrip tunggal dan Sequence untuk tindakan kustom yang sama. Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

### Sequence(Opsional)

Daftar skrip untuk dijalankan.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### Script(Diperlukan,String)

File yang akan digunakan. Jalur file dapat dimulai dengan `https://` atau `s3://`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### Args(Opsional,[String])

Daftar argumen untuk diteruskan ke skrip.



[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**Script**(Diperlukan,**String**)

File yang akan digunakan untuk satu skrip. Jalur file dapat dimulai dengan `https://` atau `s3://`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**Args**(Opsional,**[String]**)

Daftar argumen untuk diteruskan ke skrip tunggal.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**OnNodeConfigured**(Opsional,**String**)

Menentukan urutan skrip atau skrip tunggal untuk dijalankan pada node dalam Slurm antrian setelah semua tindakan bootstrap node selesai. AWS ParallelCluster tidak mendukung termasuk skrip tunggal dan Sequence untuk tindakan kustom yang sama. Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

**Sequence**(Opsional)

Daftar skrip untuk dijalankan.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**Script**(Diperlukan,**String**)

File yang akan digunakan. Jalur file dapat dimulai dengan `https://` atau `s3://`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

**Args**(Opsional,**[String]**)

Daftar argumen untuk diteruskan ke skrip.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### **Script**(Diperlukan,**String**)

File yang akan digunakan untuk satu skrip. Jalur file dapat dimulai dengan `https://` atau `s3://`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

### **Args**(Opsional,**[String]**)

Daftar argumen untuk diteruskan ke skrip tunggal.

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

#### Note

Sequenceditambahkan dimulai dengan AWS ParallelCluster versi 3.6.0. Saat Anda menentukan `Sequence`, Anda dapat membuat daftar beberapa skrip untuk tindakan kustom. AWS ParallelCluster terus mendukung konfigurasi tindakan khusus dengan satu skrip, tanpa menyertakan `Sequence`.

AWS ParallelCluster tidak mendukung termasuk skrip tunggal dan `Sequence` untuk tindakan kustom yang sama.

## **Iam**

(Opsional) Mendefinisikan pengaturan IAM opsional untuk antrian. Slurm

```
Iam:
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
```

`InstanceProfile`: *string*

`InstanceRole`: *string*

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## IamProperti

### InstanceProfile(Opsional,String)

Menentukan profil contoh untuk mengganti peran contoh default atau profil contoh untuk antrian. Slurm Anda tidak dapat menentukan keduanya InstanceProfile dan InstanceRole. Formatnya adalah `arn:${Partition}:iam::${Account}:instance-profile/${InstanceProfileName}`.

Jika ini ditentukan, AdditionalIamPolicies pengaturan S3Access dan tidak dapat ditentukan.

Kami menyarankan Anda menentukan salah satu atau kedua AdditionalIamPolicies pengaturan S3Access dan karena fitur yang ditambahkan AWS ParallelCluster sering memerlukan izin baru.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### InstanceRole(Opsional,String)

Menentukan peran contoh untuk mengganti peran contoh default atau profil contoh untuk antrian. Slurm Anda tidak dapat menentukan keduanya InstanceProfile dan InstanceRole. Formatnya adalah `arn:${Partition}:iam::${Account}:role/${RoleName}`.

Jika ini ditentukan, AdditionalIamPolicies pengaturan S3Access dan tidak dapat ditentukan.

Kami menyarankan Anda menentukan salah satu atau kedua AdditionalIamPolicies pengaturan S3Access dan karena fitur yang ditambahkan AWS ParallelCluster sering memerlukan izin baru.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### S3Access(Opsional)

Menentukan ember untuk Slurm antrian. Ini digunakan untuk membuat kebijakan untuk memberikan akses yang ditentukan ke bucket dalam Slurm antrian.

Jika ini ditentukan, InstanceRole pengaturan InstanceProfile dan tidak dapat ditentukan.

Kami menyarankan Anda menentukan salah satu atau kedua AdditionalIamPolicies pengaturan S3Access dan karena fitur yang ditambahkan AWS ParallelCluster sering memerlukan izin baru.

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

#### **BucketName**(Diperlukan,**String**)

Nama ember.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

#### **KeyName**(Opsional,**String**)

Kunci untuk ember. Nilai default-nya adalah \*.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

#### **EnableWriteAccess**(Opsional,**Boolean**)

Menunjukkan apakah akses tulis diaktifkan untuk bucket.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

#### **AdditionalIamPolicies**(Opsional)

Menentukan daftar Nama Sumber Daya Amazon (ARN) kebijakan IAM untuk Amazon EC2. Daftar ini dilampirkan ke peran root yang digunakan untuk Slurm antrian selain izin yang diperlukan oleh AWS ParallelCluster

Nama kebijakan IAM dan ARN-nya berbeda. Nama tidak bisa digunakan.

Jika ini ditentukan, InstanceRole pengaturan InstanceProfile dan tidak dapat ditentukan.

Kami menyarankan Anda menggunakan AdditionalIamPolicies karena AdditionalIamPolicies ditambahkan ke izin yang AWS ParallelCluster diperlukan, dan

InstanceRole harus menyertakan semua izin yang diperlukan. Izin yang diperlukan sering berubah dari rilis ke rilis saat fitur ditambahkan.

Tidak ada nilai default.

AdditionalIamPolicies:

- Policy: *string*

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

**Policy**(Diperlukan,**[String]**)

Daftar kebijakan IAM.

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

## SlurmSettings

(Opsional) Mendefinisikan pengaturan untuk Slurm itu berlaku untuk seluruh cluster.

SlurmSettings:

ScaledownIdleTime: *integer*

QueueUpdateStrategy: *string*

EnableMemoryBasedScheduling: *boolean*

CustomSlurmSettings: *[dict]*

CustomSlurmSettingsIncludeFile: *string*

Database:

Uri: *string*

UserName: *string*

PasswordSecretArn: *string*

Dns:

DisableManagedDns: *boolean*

HostedZoneId: *string*

UseEc2Hostnames: *boolean*

## SlurmSettingsProperti

**ScaledownIdleTime**(Opsional,**Integer**)

Mendefinisikan jumlah waktu (dalam menit) bahwa tidak ada pekerjaan dan Slurm node berakhir.

Nilai default-nya adalah 10.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### **MungeKeySecretArn**(Opsional,**String**)

Amazon Resource Name (ARN) dari rahasia Secrets AWS Manager plaintext yang berisi kunci munge yang dikodekan base64 untuk digunakan di kluster Slurm. Kunci munge ini akan digunakan untuk mengautentikasi panggilan RPC antara perintah klien Slurm dan daemon Slurm yang bertindak sebagai server jarak jauh. Jika tidak MungeKeySecretArn disediakan, AWS ParallelCluster akan menghasilkan kunci munge acak untuk cluster.

#### Note

MungeKeySecretArn didukung dimulai dengan AWS ParallelCluster versi 3.8.0.

#### Warning

Jika baru MungeKeySecretArn ditambahkan ke cluster yang ada, tidak ParallelCluster akan mengembalikan Kunci munge sebelumnya jika terjadi Rollback atau ketika kemudian menghapus. MungeKeySecretArn Sebagai gantinya, kunci munge acak baru akan dihasilkan.

Jika AWS ParallelCluster pengguna memiliki izin untuk [DescribeSecret](#) pada sumber rahasia tertentu, MungeKeySecretArn divalidasi. MungeKeySecretArn berlaku jika:

- Rahasia yang ditentukan ada, dan
- Rahasiannya adalah plaintext dan berisi string berencode base64 yang valid, dan
- Kunci munge biner yang diterjemahkan memiliki ukuran antara 256 dan 8192 bit.

Jika kebijakan IAM pengguna pcluster tidak termasuk DescribeSecret, tidak MungeKeySecretArn divalidasi dan pesan peringatan ditampilkan. Untuk informasi selengkapnya, lihat [Kebijakan AWS ParallelCluster pcluster pengguna dasar](#).

Saat Anda memperbarui MungeKeySecretArn, armada komputasi dan semua node login harus dihentikan.

Jika nilai rahasia di ARN rahasia dimodifikasi sementara ARN tetap sama, cluster tidak akan secara otomatis diperbarui dengan kunci munge baru. Untuk menggunakan kunci munge

baru ARN rahasia, Anda harus menghentikan armada komputasi dan login node kemudian menjalankan perintah berikut dari node kepala.

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

Setelah menjalankan perintah, Anda dapat melanjutkan armada komputasi dan node login: node komputasi dan login yang baru disediakan akan secara otomatis mulai menggunakan kunci munge baru.

Untuk menghasilkan kunci munge kustom yang dikodekan base64, Anda dapat menggunakan utilitas [mungekey](#) yang didistribusikan dengan perangkat lunak munge dan kemudian menyandikannya menggunakan utilitas base64 yang umumnya tersedia di OS Anda. Atau, Anda juga menggunakan bash (harap atur parameter bs antara 32 dan 1024)

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

atau Python sebagai berikut:

```
import random
import os
import base64

key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

Kebijakan Pembaruan: **KEBIJAKAN PEMBARUAN BARU DENGAN ARMADA KOMPUTASI DAN NODE LOGIN BERHENTI** (keliru tidak ditambahkan di 3.7.0).

### **QueueUpdateStrategy**(Opsional,**String**)

Menentukan strategi penggantian untuk parameter [SlurmQueues](#) bagian yang memiliki kebijakan pembaruan berikut:

[Kebijakan pembaruan: Armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel agar pengaturan ini diubah untuk pembaruan.](#)

QueueUpdateStrategyNilai hanya digunakan ketika proses pembaruan cluster dimulai.

Nilai yang valid: COMPUTE\_FLEET\_STOP | DRAIN | TERMINATE

Nilai default: `COMPUTE_FLEET_STOP`

## DRAIN

Node dalam antrian dengan nilai parameter yang diubah diatur ke `DRAINING`. Node dalam keadaan ini tidak menerima pekerjaan baru dan menjalankan pekerjaan terus selesai.

Setelah node menjadi `idle` (`DRAINED`), node diganti jika node statis, dan node dihentikan jika node dinamis. Node lain di antrian lain tanpa nilai parameter yang diubah tidak terpengaruh.

Waktu yang dibutuhkan strategi ini untuk mengganti semua node antrian dengan nilai parameter yang diubah tergantung pada beban kerja yang sedang berjalan.

## COMPUTE\_FLEET\_STOP

Nilai default `QueueUpdateStrategy` parameter. Dengan pengaturan ini, memperbarui parameter di bawah [SlurmQueues](#) bagian mengharuskan Anda [menghentikan armada komputasi](#) sebelum melakukan pembaruan kluster:

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

## TERMINATE

Dalam antrian dengan nilai parameter yang diubah, pekerjaan yang berjalan dihentikan dan node segera dimatikan.

Node statis diganti dan node dinamis dihentikan.

Node lain di antrian lain tanpa nilai parameter yang diubah tidak terpengaruh.

[Kebijakan pembaruan: Pengaturan ini tidak dianalisis selama pembaruan.](#)

### Note


`QueueUpdateStrategy` didukung dimulai dengan AWS ParallelCluster versi 3.2.0.

## `EnableMemoryBasedScheduling` (Opsional, `Boolean`)

Jika `true`, penjadwalan berbasis memori diaktifkan di Slurm. Untuk informasi lebih lanjut, lihat [SlurmQueues/ComputeResources/SchedulableMemory](#).




Nilai default-nya adalah `false`.


 Warning

Mengaktifkan penjadwalan berbasis memori berdampak pada cara Slurm penjadwal menangani pekerjaan dan alokasi node.

Untuk informasi selengkapnya, lihat [Slurmpenjadwalan berbasis memori](#).

 Note

`EnableMemoryBasedScheduling` didukung dimulai dengan AWS ParallelCluster versi 3.2.0.

 Note

[Dimulai dengan AWS ParallelCluster versi 3.7.0, `EnableMemoryBasedScheduling` dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instans di `Instans`.](#)

Untuk AWS ParallelCluster versi 3.2.0 hingga 3.6. ~~x~~, tidak

`EnableMemoryBasedScheduling` dapat diaktifkan jika Anda mengonfigurasi beberapa jenis instance di [Instance](#).

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### `CustomSlurmSettings`(Opsional, **[Dict]**)

Mendefinisikan Slurm pengaturan kustom yang berlaku untuk seluruh cluster.

Menentukan daftar kamus Slurm konfigurasi pasangan kunci-nilai yang akan ditambahkan ke akhir file yang menghasilkan. `slurm.conf` AWS ParallelCluster

Setiap kamus dalam daftar muncul sebagai baris terpisah yang ditambahkan ke file Slurm konfigurasi. Anda dapat menentukan parameter sederhana atau kompleks.

Parameter sederhana terdiri dari single key pair, seperti yang ditunjukkan pada contoh berikut:

```
- Param1: 100
```

```
- Param2: "SubParam1,SubParam2=SubValue2"
```

Contoh yang diberikan dalam Slurm konfigurasi:

```
Param1=100
Param2=SubParam1,SubParam2=SubValue2
```

Parameter Slurm konfigurasi kompleks terdiri dari beberapa nilai kunci yang dipisahkan ruang, pasangan seperti yang ditunjukkan pada contoh berikut:

```
- NodeName: test-nodes[1-10]
 CPUs: 4
 RealMemory: 4196
 ... # other node settings
- NodeSet: test-nodeset
 Nodes: test-nodes[1-10]
 ... # other nodeset settings
- PartitionName: test-partition
 Nodes: test-nodeset
 ... # other partition settings
```

Contoh, dirender dalam Slurm konfigurasi:

```
NodeName=test-nodes[1-10] CPUs=4 RealMemory=4196 ... # other node settings
NodeSet=test-nodeset Nodes=test-nodes[1-10] ... # other nodeset settings
PartitionName=test-partition Nodes=test-nodeset ... # other partition settings
```

#### Note

SlurmNode kustom tidak boleh berisi `-st-` atau `-dy-` pola dalam nama mereka. Pola-pola ini dicadangkan untuk node yang dikelola oleh AWS ParallelCluster.

Jika Anda menentukan parameter Slurm konfigurasi kustom di `CustomSlurmSettings`, Anda tidak boleh menentukan parameter Slurm konfigurasi kustom untuk `CustomSlurmSettingsIncludeFile`.


Anda hanya dapat menentukan parameter Slurm konfigurasi yang tidak dicantumkan dalam daftar penolakan. `CustomSlurmSettings` Untuk informasi tentang parameter Slurm konfigurasi daftar penolakan, lihat. [Terdaftar penolakanSlurmparameter konfigurasi untukCustomSlurmSettings](#)

AWS ParallelCluster hanya memeriksa apakah parameter ada dalam daftar penolakan. AWS ParallelCluster tidak memvalidasi sintaks atau Slurm semantik parameter konfigurasi kustom Anda. Anda bertanggung jawab untuk memvalidasi parameter Slurm konfigurasi kustom Anda. Parameter Slurm konfigurasi kustom yang tidak valid dapat menyebabkan kegagalan Slurm daemon yang dapat menyebabkan kegagalan pembuatan dan pembaruan cluster.

Untuk informasi selengkapnya tentang cara menentukan parameter Slurm konfigurasi kustom AWS ParallelCluster, lihat [Slurm kustomisasi konfigurasi](#).

Untuk informasi selengkapnya tentang parameter Slurm konfigurasi, lihat [slurm.conf](#) di dokumentasi. Slurm

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

 Note

CustomSlurmSettings didukung dimulai dengan AWS ParallelCluster versi 3.6.0.


### **CustomSlurmSettingsIncludeFile**(Opsional, **String**)

Mendefinisikan Slurm pengaturan kustom yang berlaku untuk seluruh cluster.

Menentukan Slurm file kustom yang terdiri dari parameter Slurm konfigurasi kustom yang akan ditambahkan pada akhir `slurm.conf` file yang AWS ParallelCluster menghasilkan.

Anda harus menyertakan jalur ke file. Jalan bisa dimulai dengan `https://` atau `s3://`.

Jika Anda menentukan parameter Slurm konfigurasi kustom untuk `CustomSlurmSettingsIncludeFile`, Anda tidak boleh menentukan parameter Slurm konfigurasi kustom untuk `CustomSlurmSettings`.

 Note

SlurmNode kustom tidak boleh berisi `-st-` atau `-dy-` pola dalam nama mereka. Pola-pola ini dicadangkan untuk node yang dikelola oleh AWS ParallelCluster.

Anda hanya dapat menentukan parameter Slurm konfigurasi yang tidak dicantumkan dalam daftar penolakan. `CustomSlurmSettingsIncludeFile` Untuk informasi tentang parameter

Slurm konfigurasi daftar penolakan, lihat [Terdaftar penolakanSlurmparameter konfigurasi untukCustomSlurmSettings](#)

AWS ParallelCluster hanya memeriksa apakah parameter ada dalam daftar penolakan. AWS ParallelCluster tidak memvalidasi sintaks atau Slurm semantik parameter konfigurasi kustom Anda. Anda bertanggung jawab untuk memvalidasi parameter Slurm konfigurasi kustom Anda. Parameter Slurm konfigurasi kustom yang tidak valid dapat menyebabkan kegagalan Slurm daemon yang dapat menyebabkan kegagalan pembuatan dan pembaruan cluster.

Untuk informasi selengkapnya tentang cara menentukan parameter Slurm konfigurasi kustom AWS ParallelCluster, lihat [Slurmkustomisasi konfigurasi](#).

Untuk informasi selengkapnya tentang parameter Slurm konfigurasi, lihat [slurm.conf](#) di dokumentasi. Slurm

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

#### Note

CustomSlurmSettingsdidukung dimulai dengan AWS ParallelCluster versi 3.6.0.

## Database

(Opsional) Mendefinisikan pengaturan untuk mengaktifkan Slurm Akuntansi pada cluster. Untuk informasi selengkapnya, lihat [Slurmakuntansi dengan AWS ParallelCluster](#).

#### Database:

Uri: *string*

UserName: *string*

PasswordSecretArn: *string*

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## Databaseproperti

### Uri(Diperlukan,String)

Alamat ke server database yang digunakan sebagai backend untuk Slurm akuntansi. URI ini harus diformat sebagai `host:port` dan tidak boleh berisi skema, seperti `mysql://`. Host dapat

berupa alamat IP atau nama DNS yang dapat diselesaikan oleh node kepala. Jika port tidak disediakan, AWS ParallelCluster gunakan port MySQL default 3306.

AWS ParallelCluster bootstraps database Slurm akuntansi ke cluster dan harus mengakses database.

Database harus dapat dijangkau sebelum hal berikut terjadi:

- Sebuah cluster dibuat.
- Slurmakuntansi diaktifkan dengan pembaruan cluster.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### **UserName**(Diperlukan,**String**)

Identitas yang Slurm digunakan untuk terhubung ke database, menulis log akuntansi, dan melakukan query. Pengguna harus memiliki izin baca dan tulis pada database.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### **PasswordSecretArn**(Diperlukan,**String**)

Nama Sumber Daya Amazon (ARN) dari AWS Secrets Manager rahasia yang berisi kata sandi teks biasa `UserName`. Kata sandi ini digunakan bersama dengan `UserName` dan Slurm akuntansi untuk mengautentikasi pada server database.

#### Note

Saat membuat rahasia menggunakan AWS Secrets Manager konsol pastikan untuk memilih “Jenis rahasia lainnya”, pilih teks biasa, dan hanya sertakan teks kata sandi dalam rahasia.

Untuk informasi lebih lanjut tentang cara menggunakan AWS Secrets Manager untuk membuat rahasia, lihat [Buat AWS Secrets Manager Rahasia](#)

Jika pengguna memiliki izin untuk [DescribeSecret](#), `PasswordSecretArn` divalidasi.

`PasswordSecretArnInvalid` jika rahasia yang ditentukan ada. Jika kebijakan IAM pengguna tidak disertakan `DescribeSecret`, `PasswordSecretArn` tidak divalidasi dan pesan peringatan

ditampilkan. Untuk informasi selengkapnya, lihat [Kebijakan AWS ParallelCluster `pcluster` pengguna dasar](#).

Saat Anda memperbarui `PasswordSecretArn`, armada komputasi harus dihentikan. Jika nilai rahasia berubah, dan ARN rahasia tidak berubah, cluster tidak diperbarui secara otomatis dengan kata sandi database baru. Untuk memperbarui cluster untuk nilai rahasia baru, Anda harus menjalankan perintah berikut dari dalam node kepala setelah armada komputasi dihentikan.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

#### Warning

Kami menyarankan Anda hanya mengubah kata sandi database ketika armada komputasi dihentikan untuk menghindari hilangnya data akuntansi.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### **DatabaseName**(Opsional,**String**)

Nama database pada server database (didefinisikan oleh parameter `Uri`) yang akan digunakan untuk Akuntansi Slurm.

Nama database mungkin berisi huruf kecil, angka dan garis bawah. Nama mungkin tidak lebih dari 64 karakter.

Parameter ini memetakan ke `StorageLoc` parameter [`slurmdbd.conf`](#).

Jika tidak `DatabaseName` disediakan, `ParallelCluster` akan menggunakan nama cluster untuk menentukan nilai untuk `StorageLoc`.

Memperbarui `DatabaseName` diperbolehkan, dengan pertimbangan berikut:

- Jika database dengan nama `DatabaseName` belum ada di server database, `slurmdbd` akan membuatnya. Ini akan menjadi tanggung jawab Anda untuk mengkonfigurasi ulang database baru sesuai kebutuhan (misalnya menambahkan entitas akuntansi — cluster, akun, pengguna, asosiasi, QOS, dll.).
- Jika database dengan nama `DatabaseName` sudah ada di server database, `slurmdbd` akan menggunakannya untuk fungsionalitas Akuntansi Slurm.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### Note

Databaseditambahkan dimulai dengan rilis 3.3.0.

## Dns

(Opsional) Mendefinisikan pengaturan untuk Slurm itu berlaku untuk seluruh cluster.

### Dns:

[DisableManagedDns](#): *boolean*

[HostedZoneId](#): *string*

[UseEc2Hostnames](#): *boolean*

## DnsProperti

### **DisableManagedDns**(Opsional,**Boolean**)

Jika `true`, entri DNS untuk cluster tidak dibuat dan nama Slurm node tidak dapat diselesaikan.

Secara default, AWS ParallelCluster buat zona host Route 53 tempat node terdaftar saat diluncurkan. Nilai default-nya adalah `false`. Jika `DisableManagedDns` disetel ke `true`, zona yang dihosting tidak dibuat oleh AWS ParallelCluster.

Untuk mempelajari cara menggunakan pengaturan ini untuk menyebarkan cluster di subnet tanpa akses internet, lihat. [AWS ParallelCluster dalam subnet tunggal tanpa akses internet](#)

### Warning

Sistem resolusi nama diperlukan agar cluster dapat beroperasi dengan benar. Jika `DisableManagedDns` diatur ke `true`, Anda harus memberikan sistem resolusi nama. Untuk menggunakan DNS default EC2, atur `UseEc2Hostnames` ke `true` Atau, konfigurasi resolver DNS Anda sendiri dan pastikan bahwa nama node terdaftar saat instance diluncurkan. Misalnya, Anda dapat melakukan ini dengan mengonfigurasi [CustomActions/OnNodeStart](#).

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### **HostedZoneId**(Opsional,**String**)

Mendefinisikan ID zona yang dihosting Route 53 kustom yang akan digunakan untuk resolusi nama DNS untuk cluster. Saat disediakan, AWS ParallelCluster mendaftarkan node cluster di zona host yang ditentukan dan tidak membuat zona host terkelola.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### **UseEc2Hostnames**(Opsional,**Boolean**)

Jika `true`, node komputasi cluster dikonfigurasi dengan nama host EC2 default. `SlurmNodeHostName` ini juga diperbarui dengan informasi ini. Nilai default-nya `false`.

Untuk mempelajari cara menggunakan pengaturan ini untuk menyebarkan cluster di subnet tanpa akses internet, lihat. [AWS ParallelCluster dalam subnet tunggal tanpa akses internet](#)

#### Note

Catatan ini tidak relevan dimulai dengan AWS ParallelCluster versi 3.3.0.

Untuk versi yang AWS ParallelCluster didukung sebelum 3.3.0:

Ketika `UseEc2Hostnames` diatur ke `true`, file konfigurasi Slurm diatur dengan skrip AWS ParallelCluster `prolog` dan `epilog`:

- `prolog` berjalan untuk menambahkan info node ke `/etc/hosts` node komputasi ketika setiap pekerjaan dialokasikan.
- `epilog` berjalan untuk membersihkan konten yang ditulis oleh `prolog`.

Untuk menambahkan kustom `prolog` atau `epilog` skrip, tambahkan masing-masing ke `/opt/slurm/etc/pcluster/epilog.d/` folder `/opt/slurm/etc/pcluster/prolog.d/` atau.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Bagian **SharedStorage**

(Opsional) Pengaturan penyimpanan bersama untuk cluster.



AWS ParallelCluster [mendukung baik menggunakan Amazon EBS, FSx untuk ONTAP, dan FSx untuk volume penyimpanan bersama OpenZFS, Amazon EFS dan FSx for Lustre sistem file penyimpanan bersama, atau File Cache.](#)

Di SharedStorage bagian ini, Anda dapat menentukan penyimpanan eksternal atau terkelola:

- Penyimpanan eksternal mengacu pada volume atau sistem file yang ada yang Anda kelola. AWS ParallelCluster tidak membuat atau menghapusnya.
- AWS ParallelCluster penyimpanan terkelola mengacu pada volume atau sistem file yang AWS ParallelCluster dibuat dan dapat dihapus.

Untuk [kuota penyimpanan bersama](#) dan informasi selengkapnya tentang mengonfigurasi penyimpanan bersama, lihat [Penyimpanan bersama](#) di Menggunakan. AWS ParallelCluster

#### Note

Jika AWS Batch digunakan sebagai penjadwal, FSx for Lustre hanya tersedia di node kepala cluster.

#### SharedStorage:

- MountDir: *string*  
Name: *string*  
StorageType: Ebs  
EbsSettings:
  - VolumeType: *string*
  - Iops: *integer*
  - Size: *integer*
  - Encrypted: *boolean*
  - KmsKeyId: *string*
  - SnapshotId: *string*
  - Throughput: *integer*
  - VolumeId: *string*
  - DeletionPolicy: *string*
- MountDir: *string*  
Name: *string*  
StorageType: Efs

EfsSettings:Encrypted: *boolean*KmsKeyId: *string*EncryptionInTransit: *boolean*IamAuthorization: *boolean*PerformanceMode: *string*ThroughputMode: *string*ProvisionedThroughput: *integer*FileSystemId: *string*DeletionPolicy: *string*

- MountDir: *string*

Name: *string*StorageType: FsxLustreFsxLustreSettings:StorageCapacity: *integer*DeploymentType: *string*ImportedFileChunkSize: *integer*DataCompressionType: *string*ExportPath: *string*ImportPath: *string*WeeklyMaintenanceStartTime: *string*AutomaticBackupRetentionDays: *integer*CopyTagsToBackups: *boolean*DailyAutomaticBackupStartTime: *string*PerUnitStorageThroughput: *integer*BackupId: *string*KmsKeyId: *string*FileSystemId: *string*AutoImportPolicy: *string*DriveCacheType: *string*StorageType: *string*DeletionPolicy: *string*DataRepositoryAssociations:

- Name: *string*

BatchImportMetaDataOnCreate: *boolean*DataRepositoryPath: *string*FileSystemPath: *string*ImportedFileChunkSize: *integer*AutoExportPolicy: *string*AutoImportPolicy: *string*

- MountDir: *string*

Name: *string*StorageType: FsxOntapFsxOntapSettings:

```

 VolumeId: string
- MountDir: string
 Name: string
 StorageType: FsxOpenZfs
 FsxOpenZfsSettings:
 VolumeId: string
- MountDir: string
 Name: string
 StorageType: FileCache
 FileCacheSettings:
 FileCacheId: string

```

## SharedStorage memperbarui kebijakan

- Untuk EBS terkelola/eksternal, EFS terkelola, dan FSx Lustre terkelola, kebijakan pemutakhiran adalah [Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, armada komputasi harus dihentikan atau QueueUpdateStrategy harus disetel untuk menambahkan nilai baru; armada komputasi harus dihentikan saat menghapus nilai yang ada.](#)
- Untuk EFS eksternal, FSx Lustre, FSx ONTAP, fsX dan File Cache, kebijakan OpenZfs pemutakhiran adalah, [Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## SharedStorage properti

MountDir(Diperlukan,String)

Jalur tempat penyimpanan bersama dipasang.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

Name(Diperlukan,String)

Nama penyimpanan bersama. Anda menggunakan nama ini saat memperbarui pengaturan.

### Warning

Jika Anda menentukan penyimpanan bersama AWS ParallelCluster terkelola, dan Anda mengubah nilainya `Name`, penyimpanan dan data bersama terkelola yang ada akan dihapus dan penyimpanan bersama terkelola baru dibuat. Mengubah nilai `Name` dengan pembaruan cluster sama dengan mengganti penyimpanan bersama terkelola yang ada

dengan yang baru. Pastikan Anda mencadangkan data Anda sebelum mengubah Name jika Anda perlu menyimpan data dari penyimpanan bersama yang ada.

Kebijakan pembaruan: Untuk pengaturan nilai daftar ini, armada komputasi harus dihentikan atau `QueueUpdateStrategy` harus disetel untuk menambahkan nilai baru; armada komputasi harus dihentikan saat menghapus nilai yang ada.

`StorageType`(Diperlukan,String)

Jenis penyimpanan bersama. Nilai yang didukung adalah `Ebs`, `Efs`, `FsxLustre`, `FsxOntap`, dan `FsxOpenZfs`.

Lihat informasi selengkapnya di [FsxLustreSettings](#), [FsxOntapSettings](#), dan [FsxOpenZfsSettings](#).

**Note**

Jika Anda menggunakan AWS Batch sebagai penjadwal, FSx for Lustre hanya tersedia di node kepala cluster.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## EbsSettings

(Opsional) Pengaturan untuk volume Amazon EBS.

EbsSettings:

VolumeType: *string*

Iops: *integer*

Size: *integer*

Encrypted: *boolean*

KmsKeyId: *string*

SnapshotId: *string*

VolumeId: *string*

Throughput: *integer*

DeletionPolicy: *string*

Raid:

Type: *string*

NumberOfVolumes: *integer*

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## **EbsSettings**properti

Ketika [DeletionPolicy](#) diatur ke `Delete`, volume terkelola, dengan datanya, dihapus jika cluster dihapus atau jika volume dihapus dengan pembaruan cluster.

Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#) di Menggunakan AWS ParallelCluster.

## **VolumeType**(Opsional,String)

Menentukan jenis [volume Amazon EBS](#). Nilai yang didukung adalah `gp2gp3`, `io1`, `io2`, `sc1`, `st1`, dan `standard`. Nilai default-nya adalah `gp3`.

Untuk informasi selengkapnya, lihat [Jenis volume Amazon EBS](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## **Iops**(Opsional,Integer)

Mendefinisikan jumlah IOPS untuk `io1`, `io2`, dan `gp3` jenis volume.

Nilai default, nilai yang didukung, dan `volume_iops` `volume_size` rasio bervariasi menurut `VolumeType` dan `Size`.

`VolumeType = io1`

Default Iops = 100

Nilai yang didukung Iops = 100—64000 †

Maksimum `volume_iops` terhadap `volume_size` rasio = 50 IOPS untuk setiap GiB. 5000 IOPS membutuhkan `volume_size` setidaknya 100 GiB.

`VolumeType = io2`

Default Iops = 100

Nilai yang didukung Iops = 100—64000 (256000 untuk `io2` volume Block Express) †

SizeRasio maksimum Iops = 500 IOPS untuk setiap GiB. 5000 IOPS membutuhkan `Size` minimal 10 GiB.

`VolumeType = gp3`

Default Iops = 3000

Nilai yang didukung Iops = 3000—16000

SizeRatio maksimum Iops = 500 IOPS untuk setiap GiB. 5000 IOPS membutuhkan Size minimal 10 GiB.

† IOPS maksimum dijamin hanya pada [Instans yang dibangun di atas Sistem Nitro yang](#) disediakan dengan lebih dari 32.000 IOPS. Instans lain menjamin hingga 32.000 IOPS. Kecuali Anda [memodifikasi volume](#), io1 volume sebelumnya mungkin tidak mencapai kinerja penuh. io2 Volume Block Express mendukung volume\_iops nilai hingga 256000 pada jenis R5b instans. Untuk informasi selengkapnya, lihat [io2Memblokir volume Express](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

Size(Opsional,Integer)

Menentukan ukuran volume dalam gibibytes (GiB). Nilai defaultnya adalah 35.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

Encrypted(Opsional,Boolean)

Menentukan apakah volume dienkrpsi. Nilai default-nya adalah true.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

KmsKeyId(Opsional,String)

Menentukan AWS KMS kunci kustom untuk digunakan untuk enkripsi. Pengaturan ini mengharuskan Encrypted pengaturan diatur ke true.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

SnapshotId(Opsional,String)

Menentukan ID snapshot Amazon EBS jika Anda menggunakan snapshot sebagai sumber volume.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

VolumeId(Opsional,String)

Menentukan ID volume Amazon EBS. Ketika ini ditentukan untuk sebuah EbsSettings instance, hanya MountDir parameter yang juga dapat ditentukan.

Volume harus dibuat di Availability Zone yang sama dengan HeadNode.

**Note**

Beberapa Availability Zones ditambahkan dalam AWS ParallelCluster versi 3.4.0.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

**Throughput(Opsional,Integer)**

Throughput, dalam MiB/s untuk penyediaan volume, dengan maksimum 1.000 MiB/s.

Pengaturan ini hanya valid jika VolumeType adalah gp3. Rentang yang didukung adalah 125 hingga 1000, dengan nilai default 125.

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

**DeletionPolicy(Opsional,String)**

Menentukan apakah volume harus dipertahankan, dihapus, atau snapshotted ketika cluster dihapus atau volume dihapus. Nilai yang didukung adalah Delete, Retain, dan Snapshot. Nilai default-nya adalah Delete.

Ketika [DeletionPolicy](#) diatur ke Delete, volume terkelola, dengan datanya, dihapus jika cluster dihapus atau jika volume dihapus dengan pembaruan cluster.

Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#).

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

**Note**

DeletionPolicy didukung dimulai dengan AWS ParallelCluster versi 3.2.0.

**Raid**

(Opsional) Mendefinisikan konfigurasi volume RAID.

**Raid:**

Type: *string*

NumberOfVolumes: *integer*

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## Raidproperti

Type(Diperlukan,String)

Mendefinisikan jenis array RAID. Nilai yang didukung adalah "0" (bergaris) dan "1" (dicerminkan).

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

NumberOfVolumes(Opsional,Integer)

Mendefinisikan jumlah volume Amazon EBS yang akan digunakan untuk membuat array RAID. Rentang nilai yang didukung adalah 2-5. Nilai default (ketika Raid pengaturan didefinisikan) adalah 2.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## EfsSettings

(Opsional) Pengaturan untuk sistem file Amazon EFS.

```
EfsSettings:
 Encrypted: boolean
 KmsKeyId: string
 EncryptionInTransit: boolean
 IamAuthorization: boolean
 PerformanceMode: string
 ThroughputMode: string
 ProvisionedThroughput: integer
 FileSystemId: string
 DeletionPolicy: string
```

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## EfsSettingsproperti

Ketika [DeletionPolicy](#) diatur ke `Delete`, sistem file terkelola, dengan datanya, dihapus jika cluster dihapus, atau jika sistem file dihapus dengan pembaruan cluster.

Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#) di Menggunakan AWS ParallelCluster.

Encrypted(Opsional,Boolean)

Menentukan apakah sistem file Amazon EFS dienkripsi. Nilai default-nya adalah `false`.



[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`KmsKeyId`(Opsional,String)

Menentukan AWS KMS kunci kustom untuk digunakan untuk enkripsi. Pengaturan ini mengharuskan `Encrypted` pengaturan diatur ke `true`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`EncryptionInTransit`(Opsional,Boolean)

Jika disetel ke `true`, sistem file Amazon EFS dipasang menggunakan Transport Layer Security (TLS). Secara default, ini diatur ke `false`.

**Note**

Jika AWS Batch digunakan sebagai penjadwal, `EncryptionInTransit` tidak didukung.

**Note**

`EncryptionInTransit` ditambahkan dimulai dengan AWS ParallelCluster versi 3.4.0.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`IamAuthorization`(Opsional,Boolean)

Jika disetel ke `true`, Amazon EFS diautentikasi dengan menggunakan identitas IAM sistem. Secara default, ini diatur ke `false`.

**Note**

Jika `IamAuthorization` diatur ke `true`, juga `EncryptionInTransit` harus disetel ke `true`.

**Note**

Jika AWS Batch digunakan sebagai penjadwal, `IamAuthorization` tidak didukung.

**Note**

IamAuthorizationditambahkan dimulai dengan AWS ParallelCluster versi 3.4.0.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**PerformanceMode(Opsional,String)**

Menentukan mode kinerja sistem file Amazon EFS. Nilai yang didukung adalah `generalPurpose` dan `maxIO`. Nilai default-nya adalah `generalPurpose`. Untuk informasi selengkapnya, lihat [Mode performa](#) dalam Panduan Pengguna Amazon Elastic File System.

Kami merekomendasikan mode `generalPurpose` kinerja untuk sebagian besar sistem file.

Sistem file yang menggunakan mode `maxIO` kinerja dapat menskalakan ke tingkat throughput agregat dan operasi per detik yang lebih tinggi. Namun, ada trade-off latensi yang sedikit lebih tinggi untuk sebagian besar operasi file.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**ThroughputMode(Opsional,String)**

Menentukan mode throughput sistem file Amazon EFS. Nilai yang didukung adalah `bursting` dan `provisioned`. Nilai default-nya adalah `bursting`. Kapan `provisioned` digunakan, `ProvisionedThroughput` harus ditentukan.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**ProvisionedThroughput(Diperlukan kapan `ThroughputModeprovisioned`,Integer)**

Mendefinisikan throughput yang disediakan (dalam MIB/s) dari sistem file Amazon EFS, diukur dalam MIB/s. Ini sesuai dengan [ProvisionedThroughputInMibbps](#) parameter di Referensi API Amazon EFS.

Jika Anda menggunakan parameter ini, Anda harus mengatur `ThroughputMode` ke `provisioned`.

Rentang yang didukung adalah 1 -1024. Untuk meminta kenaikan batas, hubungi AWS Support.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## FileSystemId(Opsional,String)

Mendefinisikan ID sistem file Amazon EFS untuk sistem file yang ada.

Jika klaster dikonfigurasi untuk menjangkau beberapa Availability Zone, Anda harus menentukan target pemasangan sistem file di setiap Availability Zone yang digunakan oleh cluster.

Ketika ini ditentukan, hanya `MountDir` dapat ditentukan. Tidak ada yang lain yang `EfsSettings` dapat ditentukan.

Jika Anda mengatur opsi ini, berikut ini harus benar untuk sistem file yang Anda tentukan:

- Sistem file memiliki target pemasangan yang ada di setiap Zona Ketersediaan cluster, dengan lalu lintas NFS masuk dan keluar diizinkan dari dan. `HeadNode ComputeNodes` Beberapa zona ketersediaan dikonfigurasi dalam [Penjadwalan SlurmQueues//Jaringan/SubnetIds](#).

Untuk memastikan lalu lintas diperbolehkan antara cluster dan sistem file, Anda dapat melakukan salah satu hal berikut:

- Konfigurasi grup keamanan target pemasangan untuk memungkinkan lalu lintas ke dan dari CIDR atau daftar awalan subnet cluster.

### Note

AWS ParallelCluster memvalidasi bahwa port terbuka dan bahwa CIDR atau daftar awalan dikonfigurasi. AWS ParallelCluster tidak memvalidasi konten blok CIDR atau daftar awalan.

- Tetapkan grup keamanan khusus untuk node cluster dengan menggunakan [SlurmQueues/Networking/SecurityGroups](#) dan [HeadNode/Networking/SecurityGroups](#). Grup keamanan khusus harus dikonfigurasi untuk memungkinkan lalu lintas antara cluster dan sistem file.

### Note

Jika semua node cluster menggunakan grup keamanan khusus, AWS ParallelCluster hanya memvalidasi bahwa port terbuka. AWS ParallelCluster tidak memvalidasi bahwa sumber dan tujuan dikonfigurasi dengan benar.

**⚠ Warning**

EFS hanya OneZone didukung jika semua node komputasi dan node kepala berada di Availability Zone yang sama. EFS hanya OneZone dapat memiliki satu target pemasangan.

**ℹ Note**

Beberapa Availability Zones ditambahkan dalam AWS ParallelCluster versi 3.4.0.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**DeletionPolicy(Optional,String)**

Menentukan apakah sistem file harus dipertahankan atau dihapus ketika sistem file dihapus dari cluster atau cluster dihapus. Nilai yang di-support adalah Delete dan Retain. Nilai default-nya adalah Delete.

Ketika [DeletionPolicy](#) diatur keDelete, sistem file terkelola, dengan datanya, dihapus jika cluster dihapus, atau jika sistem file dihapus dengan pembaruan cluster.

Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#).

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**ℹ Note**

DeletionPolicydidukung dimulai dengan AWS ParallelCluster versi 3.3.0.

**FsxLustreSettings****ℹ Note**

Anda harus menentukan FsxLustreSettings FsxLustre apakah ditentukan untuk [StorageType](#).

(Opsional) Pengaturan untuk sistem file FSx for Lustre.

#### FsxLustreSettings:

StorageCapacity: *integer*  
DeploymentType: *string*  
ImportedFileChunkSize: *integer*  
DataCompressionType: *string*  
ExportPath: *string*  
ImportPath: *string*  
WeeklyMaintenanceStartTime: *string*  
AutomaticBackupRetentionDays: *integer*  
CopyTagsToBackups: *boolean*  
DailyAutomaticBackupStartTime: *string*  
PerUnitStorageThroughput: *integer*  
BackupId: *string* # BackupId cannot coexist with some of the fields  
KmsKeyId: *string*  
FileSystemId: *string* # FileSystemId cannot coexist with other fields  
AutoImportPolicy: *string*  
DriveCacheType: *string*  
StorageType: *string*  
DeletionPolicy: *string*

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

#### Note

Jika AWS Batch digunakan sebagai penjadwal, FSx for Lustre hanya tersedia di node kepala cluster.

### FsxLustreSettingsproperti

Ketika DeletionPolicy diatur ke `Delete`, sistem file terkelola, dengan datanya, dihapus jika cluster dihapus, atau jika sistem file dihapus dengan pembaruan cluster.

Untuk informasi selengkapnya, lihat Penyimpanan bersama.

#### StorageCapacity(Diperlukan,Integer)

Menetapkan kapasitas penyimpanan sistem file FSx for Lustre, di GiB.

`StorageCapacity` diperlukan jika Anda membuat sistem file baru. Jangan sertakan `StorageCapacity` jika `BackupId` atau `FileSystemId` ditentukan.

- Untuk SCRATCH\_2, PERSISTENT\_1, dan jenis PERSISTENT\_2 penerapan, nilai yang valid adalah 1200 GiB, 2400 GiB, dan kenaikan 2400 GiB.
- Untuk jenis deployment SCRATCH\_1, nilai yang valid adalah 1200 GiB, 2400 GiB, dan peningkatan sebesar 3600 GiB.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## DeploymentType(Opsional,String)

Menentukan jenis penyebaran sistem file FSx for Lustre. Nilai yang didukung adalah SCRATCH\_1, SCRATCH\_2, PERSISTENT\_1, dan PERSISTENT\_2. Nilai default-nya adalah SCRATCH\_2.

Pilih SCRATCH\_1 dan jenis SCRATCH\_2 penerapan saat Anda membutuhkan penyimpanan sementara dan pemrosesan data jangka pendek. Jenis SCRATCH\_2 penyebaran menyediakan enkripsi data transit dan kapasitas throughput burst yang lebih tinggi daripada SCRATCH\_1

Pilih jenis PERSISTENT\_1 penerapan untuk penyimpanan jangka panjang dan untuk beban kerja yang berfokus pada throughput yang tidak sensitif terhadap latensi. PERSISTENT\_1 mendukung enkripsi data dalam perjalanan. Ini tersedia di semua Wilayah AWS tempat FSx for Lustre tersedia.

Pilih jenis PERSISTENT\_2 penerapan untuk penyimpanan jangka panjang dan untuk beban kerja sensitif latensi yang memerlukan tingkat IOPS dan throughput tertinggi. PERSISTENT\_2 mendukung penyimpanan SSD dan menawarkan lebih tinggi PerUnitStorageThroughput (hingga 1000 MB/s/Tib). PERSISTENT\_2 tersedia dalam jumlah terbatas Wilayah AWS. Untuk informasi selengkapnya tentang jenis penerapan dan daftar Wilayah AWS tempat PERSISTENT\_2 tersedia, lihat [Opsi penerapan sistem file untuk FSx for Lustre di Panduan Pengguna Amazon FSx for Lustre](#).

[Enkripsi data dalam perjalanan diaktifkan secara otomatis saat Anda mengakses SCRATCH\\_2, PERSISTENT\\_1, atau sistem file jenis PERSISTENT\\_2 penyebaran dari instans Amazon EC2 yang mendukung fitur ini.](#)

Enkripsi data dalam perjalanan untuk SCRATCH\_2, PERSISTENT\_1, dan jenis PERSISTENT\_2 penerapan didukung saat diakses dari jenis instans yang didukung di support Wilayah AWS. Untuk informasi selengkapnya, lihat [Mengenkripsi data saat transit di Panduan Pengguna Amazon FSx for Lustre](#).

**Note**

Support untuk jenis PERSISTENT\_2 deployment ditambahkan dengan AWS ParallelCluster versi 3.2.0.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**ImportedFileChunkSize(Opsional,Integer)**

Untuk file yang diimpor dari repositori data, nilai ini menentukan jumlah garis dan jumlah maksimum data untuk setiap file (dalam MiB) yang disimpan pada disk fisik tunggal. Jumlah maksimum disk yang suatu file dapat diberi stripe dibatasi oleh jumlah total disk yang membentuk sistem file.

Ukuran potongan default adalah 1.024 MiB (1 GiB) dan dapat mencapai 512.000 MiB (500 GiB). Objek Amazon S3 memiliki ukuran maksimal 5 TB.

**Note**

Parameter ini tidak didukung untuk sistem file yang menggunakan tipe PERSISTENT\_2 penerapan. Untuk petunjuk tentang cara mengonfigurasi asosiasi repositori data, lihat [Menautkan sistem file ke bucket S3](#) di Panduan Pengguna Amazon FSx for Lustre.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**DataCompressionType(Opsional,String)**

Menetapkan konfigurasi kompresi data untuk sistem file FSx for Lustre. Nilai yang didukung adalah LZ4. LZ4 menunjukkan bahwa kompresi data dihidupkan dengan algoritma LZ4. Bila DataCompressionType tidak ditentukan, kompresi data dimatikan saat sistem file dibuat.

Untuk informasi lebih lanjut, lihat [Kompresi data Lustre](#).

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**ExportPath(Opsional,String)**

Jalur di Amazon S3 tempat root sistem file FSx for Lustre Anda diekspor. Pengaturan ini hanya didukung ketika ImportPath parameter ditentukan. Jalur harus menggunakan bucket Amazon

S3 yang sama seperti yang ditentukan dalam. `ImportPath` Anda dapat memberikan awalan opsional yang data baru dan yang diubah akan diekspor dari sistem file FSx for Lustre Anda. Jika `ExportPath` nilai tidak disediakan, FSx for Lustre menetapkan jalur ekspor default, `s3://import-bucket/FSxLustre[creation-timestamp]` Stempel waktu ada dalam format UTC, misalnya `s3://import-bucket/FSxLustre20181105T222312Z`.

Bucket ekspor Amazon S3 harus sama dengan bucket impor yang ditentukan oleh `ImportPath`. Jika Anda hanya menentukan nama bucket, misalnya `s3://import-bucket`, Anda mendapatkan pemetaan 1:1 objek sistem file ke objek bucket Amazon S3. Pemetaan ini berarti bahwa data input di Amazon S3 ditimpa pada ekspor. Jika Anda memberikan awalan kustom di jalur ekspor, seperti `s3://import-bucket/[custom-optional-prefix]`, FSx for Lustre mengeksport konten sistem file Anda ke awalan ekspor tersebut di bucket Amazon S3.

#### Note

Parameter ini tidak didukung untuk sistem file yang menggunakan tipe `PERSISTENT_2` penerapan. Konfigurasi asosiasi repositori data seperti yang dijelaskan dalam [Menautkan sistem file ke bucket S3 di Panduan Pengguna Amazon FSx for Lustre](#).

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`ImportPath`(Opsional,String)

Jalur ke bucket Amazon S3 (termasuk awalan opsional) yang Anda gunakan sebagai repositori data untuk sistem file FSx for Lustre Anda. Root dari sistem file FSx for Lustre Anda akan dipetakan ke root dari bucket Amazon S3 yang Anda pilih. Contoh adalah `s3://import-bucket/optional-prefix`. Jika Anda menentukan prefiks setelah nama bucket Amazon S3, hanya tombol objek dengan prefiks tersebut yang dimuat ke dalam sistem file.

#### Note

Parameter ini tidak didukung untuk sistem file yang menggunakan tipe `PERSISTENT_2` penerapan. Konfigurasi asosiasi repositori data seperti yang dijelaskan dalam [Menautkan sistem file ke bucket S3 di Panduan Pengguna Amazon FSx for Lustre](#).

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)



### WeeklyMaintenanceStartTime(Opsional,String)

Waktu mulai yang disukai untuk melakukan pemeliharaan mingguan. Ini dalam "d:HH:MM" format di zona waktu UTC+0. Untuk format ini, d adalah angka hari kerja dari 1 hingga 7, dimulai dengan Senin dan berakhir dengan hari Minggu. Tanda kutip diperlukan untuk bidang ini.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### AutomaticBackupRetentionDays(Opsional,Integer)

Jumlah hari penyimpanan cadangan otomatis. Mengatur ini menjadi 0 menonaktifkan cadangan otomatis. Rentang yang didukung adalah 0-90. Default-nya adalah 0. Pengaturan ini hanya berlaku untuk digunakan dengan PERSISTENT\_1 dan jenis PERSISTENT\_2 penerapan. Untuk informasi selengkapnya, lihat [Bekerja dengan cadangan](#) di Panduan Pengguna Amazon FSx for Lustre.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### CopyTagsToBackups(Opsional,Boolean)

Jika true, salin tag untuk sistem file FSx for Lustre ke backup. Nilai ini default menjadi false. Jika disetel ke true, semua tag untuk sistem file disalin ke semua backup otomatis dan yang dimulai pengguna di mana pengguna tidak menentukan tag. Jika nilai ini true, dan Anda menentukan satu atau beberapa tag, hanya tag yang ditentukan yang disalin ke cadangan. Jika Anda menentukan satu atau beberapa tag saat membuat cadangan yang dimulai pengguna, tidak ada tag yang disalin dari sistem file, terlepas dari nilai ini. Pengaturan ini hanya berlaku untuk digunakan dengan PERSISTENT\_1 dan jenis PERSISTENT\_2 penerapan.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### DailyAutomaticBackupStartTime(Opsional,String)

Waktu harian yang berulang, dalam HH:MM format. HH adalah jam tanpa empatik dalam sehari (00-23). MM adalah menit empatik nol jam (00-59). Misalnya, 05:00 menentukan jam 5 pagi setiap hari. Pengaturan ini hanya berlaku untuk digunakan dengan PERSISTENT\_1 dan jenis PERSISTENT\_2 penerapan.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### PerUnitStorageThroughput(Diperlukan untuk **PERSISTENT\_1** dan jenis **PERSISTENT\_2** penerapan,Integer)

Menjelaskan jumlah throughput baca dan tulis untuk setiap 1 tebibyte penyimpanan, dalam MB/s/Tib. Kapasitas throughput sistem file dihitung dengan mengalikan kapasitas

penyimpanan sistem file (TiB) dengan (MB/s/TiB). `PerUnitStorageThroughput` Untuk sistem file 2,4 TiB, penyediaan 50 MB/s/TiB menghasilkan 120 MB/s throughput sistem file. `PerUnitStorageThroughput` Anda membayar untuk jumlah throughput yang Anda sediakan. Ini sesuai dengan [PerUnitStorageThroughput](#) properti.

Nilai yang valid:

Penyimpanan SSD PERSISTENT\_1:50, 100, 200 MB/s/Tib.

Penyimpanan HDD PERSISTENT\_1:12, 40 MB/s/Tib.

Penyimpanan SSD PERSISTENT\_2:125, 250, 500, 1000 MB/s/Tib.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`BackupId`(Opsional,String)

Menentukan ID cadangan yang akan digunakan untuk memulihkan sistem file FSx for Lustre dari cadangan yang ada. Ketika `BackupId` pengaturan ditentukan, `AutoImportPolicy`, `DeploymentType`, `ExportPath`, `KmsKeyId`, `ImportPath`, `ImportedFileChunkSizeStorageCapacity`, dan `PerUnitStorageThroughput` pengaturan tidak boleh ditentukan. Pengaturan ini dibaca dari cadangan. Selain itu `AutoImportPolicy`, `ExportPath`, `ImportPath`, dan `ImportedFileChunkSize` pengaturan tidak boleh ditentukan. Ini sesuai dengan [BackupId](#) properti.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`KmsKeyId`(Opsional,String)

ID ID kunci AWS Key Management Service (AWS KMS) yang digunakan untuk mengenkripsi data sistem file FSx for Lustre untuk persisten FSx for Lustre file system saat istirahat. Jika tidak ditentukan, kunci terkelola FSx for Lustre digunakan. Sistem file SCRATCH\_1 dan SCRATCH\_2 FSx for Lustre selalu dienkripsi saat istirahat menggunakan kunci terkelola FSx for Lustre. Untuk informasi selengkapnya, lihat [Mengekripsi](#) di Referensi AWS Key Management Service API.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

`FileSystemId`(Opsional,String)

Menentukan ID dari FSx yang ada untuk sistem file Lustre.

Jika opsi ini ditentukan, hanya `FileSystemId` pengaturan `MountDir` dan di `FsxLustreSettings` yang digunakan. Semua pengaturan lain di `FsxLustreSettings` diabaikan.

**Note**

Jika AWS Batch scheduler digunakan, FSx for Lustre hanya tersedia di node kepala.

**Note**

Sistem file harus dikaitkan dengan grup keamanan yang memungkinkan lalu lintas TCP masuk dan keluar melalui port 988, 1021, 1022, dan 1023.

Pastikan lalu lintas diperbolehkan antara cluster dan sistem file dengan melakukan salah satu hal berikut:

- Konfigurasi grup keamanan sistem file untuk memungkinkan lalu lintas ke dan dari CIDR atau daftar awalan subnet cluster.

**Note**

AWS ParallelCluster memvalidasi bahwa port terbuka dan bahwa CIDR atau daftar awalan dikonfigurasi. AWS ParallelCluster tidak memvalidasi konten blok CIDR atau daftar awalan.

- Tetapkan grup keamanan khusus untuk node cluster dengan menggunakan [SlurmQueues/Networking/SecurityGroups](#) dan [HeadNode/Networking/SecurityGroups](#). Grup keamanan khusus harus dikonfigurasi untuk memungkinkan lalu lintas antara cluster dan sistem file.

**Note**

Jika semua node cluster menggunakan grup keamanan khusus, AWS ParallelCluster hanya memvalidasi bahwa port terbuka. AWS ParallelCluster tidak memvalidasi bahwa sumber dan tujuan dikonfigurasi dengan benar.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## AutoImportPolicy(Opsional,String)

Saat Anda membuat sistem file FSx for Lustre, objek Amazon S3 yang ada akan muncul sebagai daftar file dan direktori. Gunakan properti ini untuk memilih bagaimana FSx for Lustre membuat daftar file dan direktori tetap mutakhir saat menambahkan atau memodifikasi objek di bucket Amazon S3 tertaut. `AutoImportPolicy` dapat memiliki nilai-nilai berikut:

- `NEW`- Impor otomatis aktif. FSx for Lustre secara otomatis mengimpor daftar direktori objek baru yang ditambahkan ke bucket Amazon S3 tertaut yang saat ini tidak ada di sistem file FSx for Lustre.
- `NEW_CHANGED`- Impor otomatis aktif. FSx for Lustre secara otomatis mengimpor file dan daftar direktori objek baru yang ditambahkan ke bucket Amazon S3 dan objek yang ada yang diubah di bucket Amazon S3 setelah Anda memilih opsi ini.
- `NEW_CHANGED_DELETED`- Impor otomatis aktif. FSx for Lustre secara otomatis mengimpor file dan daftar direktori objek baru yang ditambahkan ke bucket Amazon S3, objek apa pun yang ada yang diubah di bucket Amazon S3, dan objek apa pun yang dihapus di bucket Amazon S3 setelah Anda memilih opsi ini.

### Note

Support untuk `NEW_CHANGED_DELETED` ditambahkan di AWS ParallelCluster versi 3.1.1.

Jika `AutoImportPolicy` tidak ditentukan, impor otomatis dinonaktifkan. FSx for Lustre hanya memperbarui daftar file dan direktori dari bucket Amazon S3 yang ditautkan saat sistem file dibuat. FSx for Lustre tidak memperbarui daftar file dan direktori untuk objek baru atau yang diubah setelah memilih opsi ini.

Untuk informasi selengkapnya, lihat [Mengimpor pembaruan secara otomatis dari bucket S3 Anda](#) di Panduan Pengguna Amazon FSx for Lustre.

### Note

Parameter ini tidak didukung untuk sistem file yang menggunakan tipe `PERSISTENT_2` penerapan. Untuk petunjuk tentang cara mengonfigurasi asosiasi repositori data, lihat [Menautkan sistem file ke bucket S3](#) di Panduan Pengguna Amazon FSx for Lustre.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### DriveCacheType(Opsional,String)

Menentukan bahwa sistem file memiliki cache drive SSD. Ini hanya dapat diatur jika StorageType pengaturan diatur keHDD, dan DeploymentType pengaturan diatur kePERSISTENT\_1. Ini sesuai dengan [DriveCacheType](#) properti. Untuk informasi selengkapnya, lihat [opsi penerapan FSx for Lustre](#) di Panduan Pengguna Amazon FSx for Lustre.

Satu-satunya nilai yang valid adalah READ. Untuk menonaktifkan cache drive SSD, jangan tentukan DriveCacheType pengaturannya.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### StorageType(Opsional,String)

Menetapkan jenis penyimpanan untuk sistem file FSx for Lustre yang Anda buat. Nilai yang valid adalah SSD dan HDD.

- Mengatur ke SSD untuk menggunakan penyimpanan solid state drive.
- Atur HDD untuk menggunakan penyimpanan hard disk drive. HDD didukung pada jenis PERSISTENT penerapan.

Nilai default-nya adalah SSD. Untuk informasi selengkapnya, lihat [Pilihan Jenis Penyimpanan](#) di Panduan Pengguna Amazon FSx untuk Windows dan [Beberapa Opsi Penyimpanan](#) di Panduan Pengguna Amazon FSx for Lustre.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### DeletionPolicy(Opsional,String)

Menentukan apakah sistem file harus dipertahankan atau dihapus ketika sistem file dihapus dari cluster atau cluster dihapus. Nilai yang di-support adalah Delete dan Retain. Nilai default-nya adalah Delete.

Ketika [DeletionPolicy](#) diatur keDelete, sistem file dikelola, dengan datanya, dihapus jika cluster dihapus, atau jika sistem file dihapus dengan pembaruan cluster.

Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#).

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**Note**

DeletionPolicy didukung dimulai dengan AWS ParallelCluster versi 3.3.0.

**DataRepositoryAssociations(Opsional,String)**

Daftar DRA (hingga 8 per sistem file)

Setiap asosiasi repositori data harus memiliki direktori sistem file Amazon FSx yang unik dan bucket atau awalan S3 unik yang terkait dengannya.

Anda tidak dapat menggunakan [ExportPath](#) dan [ImportPath](#) pada FsxLustreSettings saat yang sama menggunakan DRA.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**Name(Diperlukan,String)**

Nama DRA. Anda menggunakan nama ini saat memperbarui pengaturan.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**BatchImportMetaDataOnCreate(Opsional,Boolean)**

Bendera boolean yang menunjukkan apakah tugas repositori data impor untuk mengimpor metadata harus dijalankan setelah asosiasi repositori data dibuat. Tugas berjalan jika flag ini disetel ke true.

Nilai default: false

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

**DataRepositoryPath(Diperlukan,String)**

Jalur ke repositori data Amazon S3 yang akan ditautkan ke sistem file. Path dapat berupa bucket S3 atau awalan dalam format. s3://myBucket/myPrefix/ Path ini menentukan di mana dalam file repositori data S3 akan diimpor dari atau diekspor ke.

Tidak dapat tumpang tindih dengan DRA lainnya

Pola: `^[^\u0000\u0085\u2028\u2029\r\n]{3,4357}$`

Minimal: 3

Maksimum: 4357


[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

FileSystemPath(Diperlukan,String)

Jalur pada sistem file Amazon FSx for Lustre yang menunjuk ke direktori tingkat tinggi (seperti) atau subdirektori (/ns1//ns1/subdir/seperti) yang akan dipetakan 1-1 dengan. DataRepositoryPath Garis miring ke depan dalam nama diperlukan. Dua asosiasi repositori data tidak dapat memiliki jalur sistem file yang tumpang tindih. Misalnya, jika repositori data dikaitkan dengan jalur sistem file/ns1/, maka Anda tidak dapat menautkan repositori data lain dengan jalur sistem file. /ns1/ns2

Path ini menentukan di mana file sistem file Anda akan diekspor dari atau diimpor ke. Direktori sistem file ini hanya dapat ditautkan ke satu bucket Amazon S3, dan tidak ada bucket S3 lain yang dapat ditautkan ke direktori.

Tidak dapat tumpang tindih dengan DRA lainnya

 Note

Jika Anda hanya menentukan garis miring (/) sebagai jalur sistem file, Anda hanya dapat menautkan satu repositori data ke sistem file. Anda hanya dapat menentukan "/" sebagai jalur sistem file untuk repositori data pertama yang terkait dengan sistem file.

Pola: `^[^\u0000\u0085\u2028\u2029\r\n]{1,4096}$`

Minimal: 1

Maksimum: 4096

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

ImportedFileChunkSize(Opsional,Integer)

Untuk file yang diimpor dari repositori data, nilai ini menentukan jumlah garis dan jumlah maksimum data per file (dalam MiB) yang disimpan pada disk fisik tunggal. Jumlah maksimum disk yang dapat digaris-garis satu file dibatasi oleh jumlah total disk yang membentuk sistem file atau cache.

Ukuran potongan default adalah 1.024 MiB (1 GiB) dan dapat mencapai 512.000 MiB (500 GiB). Objek Amazon S3 memiliki ukuran maksimal 5 TB.

Minimal: 1

Maksimum: 4096

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

AutoExportPolicy(Opsional,Array of strings)

Daftar ini dapat berisi satu atau lebih dari nilai-nilai berikut:

- NEW- File dan direktori baru secara otomatis diekspor ke repositori data saat ditambahkan ke sistem file.
- CHANGED- Perubahan pada file dan direktori pada sistem file secara otomatis diekspor ke repositori data.
- DELETED- File dan direktori secara otomatis dihapus pada repositori data ketika mereka dihapus pada sistem file.

Anda dapat menentukan kombinasi jenis acara apa pun untuk AndaAutoExportPolicy.

Maksimum: 3

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

AutoImportPolicy(Opsional,Array of strings)

Daftar ini dapat berisi satu atau lebih dari nilai-nilai berikut:

- NEW- Amazon FSx secara otomatis mengimpor metadata file yang ditambahkan ke bucket S3 tertaut yang saat ini tidak ada di sistem file FSx.
- CHANGED- Amazon FSx secara otomatis memperbarui metadata file dan membatalkan konten file yang ada pada sistem file saat file berubah dalam repositori data.
- DELETED- Amazon FSx secara otomatis menghapus file pada sistem file karena file yang sesuai dihapus dalam repositori data.

Anda dapat menentukan kombinasi jenis acara apa pun untuk AndaAutoImportPolicy.

Maksimum: 3

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)



## FsxOntapSettings

### Note

Anda harus menentukan FsxOntapSettings FsxOntap apakah ditentukan untuk [StorageType](#).

(Opsional) Pengaturan untuk FSx untuk sistem file ONTAP.

```
FsxOntapSettings:
 VolumeId: string
```

### FsxOntapSettingsproperti

VolumeId(Diperlukan,String)

Menentukan ID volume FSx yang ada untuk sistem ONTAP.

### Note

- Jika AWS Batch penjadwal digunakan, fsX untuk ONTAP hanya tersedia di node kepala.
- Jika fsX untuk jenis penyebaran ONTAP adalah `Multi-AZ`, pastikan bahwa tabel rute subnet node head dikonfigurasi dengan benar.
- Support untuk FSx untuk ONTAP telah ditambahkan dalam AWS ParallelCluster versi 3.2.0.
- Sistem file harus dikaitkan dengan grup keamanan yang memungkinkan lalu lintas TCP dan UDP masuk dan keluar melalui port 111, 635, 2049, dan 4046.

Pastikan lalu lintas diperbolehkan antara cluster dan sistem file dengan melakukan salah satu tindakan berikut:

- Konfigurasi grup keamanan sistem file untuk memungkinkan lalu lintas ke dan dari CIDR atau daftar awalan subnet cluster.

**Note**

AWS ParallelCluster memvalidasi bahwa port terbuka dan bahwa CIDR atau daftar awalan dikonfigurasi. AWS ParallelCluster tidak memvalidasi konten blok CIDR atau daftar awalan.

- Tetapkan grup keamanan khusus untuk node cluster dengan menggunakan [SlurmQueues/Networking/SecurityGroups](#) dan [HeadNode/Networking/SecurityGroups](#). Grup keamanan khusus harus dikonfigurasi untuk memungkinkan lalu lintas antara cluster dan sistem file.

**Note**

Jika semua node cluster menggunakan grup keamanan khusus, AWS ParallelCluster hanya memvalidasi bahwa port terbuka. AWS ParallelCluster tidak memvalidasi bahwa sumber dan tujuan dikonfigurasi dengan benar.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## FsxOpenZfsSettings

**Note**

Anda harus menentukan FsxOpenZfsSettings FsxOpenZfs apakah ditentukan untuk [StorageType](#).

(Opsional) Pengaturan untuk FSx untuk sistem file OpenZFS.

```
FsxOpenZfsSettings:
VolumeId: string
```

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## FsxOpenZfsSettings

properti

VolumeId(Diperlukan,String)

Menentukan ID volume FSx yang ada untuk sistem OpenZFS.

**Note**

- Jika AWS Batch penjadwal digunakan, FSx untuk OpenZFS hanya tersedia di node kepala.
- Support untuk FSx untuk OpenZFS telah ditambahkan dalam versi 3.2.0. AWS ParallelCluster
- Sistem file harus dikaitkan dengan grup keamanan yang memungkinkan lalu lintas TCP dan UDP masuk dan keluar melalui port 111, 2049, 20001, 20002, dan 20003.

Pastikan lalu lintas diperbolehkan antara cluster dan sistem file dengan melakukan salah satu hal berikut:

- Konfigurasi grup keamanan sistem file untuk memungkinkan lalu lintas ke dan dari CIDR atau daftar awalan subnet cluster.

**Note**

AWS ParallelCluster memvalidasi bahwa port terbuka dan bahwa CIDR atau daftar awalan dikonfigurasi. AWS ParallelCluster tidak memvalidasi konten blok CIDR atau daftar awalan.

- Tetapkan grup keamanan khusus untuk node cluster dengan menggunakan [SlurmQueues/Networking/SecurityGroups](#) dan [HeadNode/Networking/SecurityGroups](#). Grup keamanan khusus harus dikonfigurasi untuk memungkinkan lalu lintas antara cluster dan sistem file.

**Note**

Jika semua node cluster menggunakan grup keamanan khusus, AWS ParallelCluster hanya memvalidasi bahwa port terbuka. AWS ParallelCluster tidak memvalidasi bahwa sumber dan tujuan dikonfigurasi dengan benar.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## FileCacheSettings

### Note

Anda harus menentukan FileCacheSettings FileCache apakah ditentukan untuk [StorageType](#).

(Opsional) Pengaturan untuk File Cache.

```
FileCacheSettings:
FileCacheId: string
```

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

### FileCacheSettingsproperti

FileCacheId(Diperlukan,String)

Menentukan ID Cache File dari File Cache yang ada.

### Note

- File Cache tidak mendukung AWS Batch penjadwal.
- Support untuk File Cache ditambahkan dalam AWS ParallelCluster versi 3.7.0.
- Sistem file harus dikaitkan dengan grup keamanan yang memungkinkan lalu lintas TCP masuk dan keluar melalui port 988.

Pastikan lalu lintas diperbolehkan antara cluster dan sistem file dengan melakukan salah satu hal berikut:

- Konfigurasi grup keamanan File Cache untuk memungkinkan lalu lintas ke dan dari CIDR atau daftar awalan subnet cluster.

**Note**

AWS ParallelCluster memvalidasi bahwa port terbuka dan bahwa CIDR atau daftar awalan dikonfigurasi. AWS ParallelCluster tidak memvalidasi konten blok CIDR atau daftar awalan.

- Tetapkan grup keamanan khusus untuk node cluster dengan menggunakan [SlurmQueues/Networking/SecurityGroups](#) dan [HeadNode/Networking/SecurityGroups](#). Grup keamanan khusus harus dikonfigurasi untuk memungkinkan lalu lintas antara cluster dan sistem file.

**Note**

Jika semua node cluster menggunakan grup keamanan khusus, AWS ParallelCluster hanya memvalidasi bahwa port terbuka. AWS ParallelCluster tidak memvalidasi bahwa sumber dan tujuan dikonfigurasi dengan benar.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## Bagian **Iam**

(Opsional) Menentukan properti IAM untuk cluster.

Iam:

Roles:

LambdaFunctionsRole: *string*

PermissionsBoundary: *string*

ResourcePrefix: *string*

Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.

### **Iam**properti

PermissionsBoundary(Opsional,String)

ARN dari kebijakan IAM untuk digunakan sebagai batas izin untuk semua peran yang dibuat oleh AWS ParallelCluster Untuk informasi selengkapnya, lihat [Batas izin untuk entitas IAM](#) di Panduan Pengguna IAM. Formatnya adalah `arn:${Partition}:iam:${Account}:policy/${PolicyName}`.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## Roles(Optional)

Menentukan pengaturan untuk peran IAM yang digunakan oleh cluster.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

### LambdaFunctionsRole(Optional,String)

ARN dari peran IAM yang akan digunakan untuk. AWS Lambda Ini mengesampingkan peran default yang dilampirkan ke semua fungsi Lambda yang mendukung sumber daya khusus. AWS CloudFormation Lambda perlu dikonfigurasi sebagai prinsipal yang diizinkan untuk mengambil peran. Ini tidak akan mengesampingkan peran fungsi Lambda yang digunakan untuk. AWS Batch Formatnya adalah `arn:${Partition}:iam::${Account}:role/${RoleName}`.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## ResourcePrefix(Optional)

Menentukan jalur atau nama awalan untuk sumber daya IAM yang dibuat oleh. AWS ParallelCluster

Awalan sumber daya harus mengikuti [aturan penamaan yang ditentukan oleh IAM](#):

- Sebuah nama dapat berisi hingga 30 karakter.
- Sebuah nama hanya bisa berupa string tanpa karakter slash (/).
- Path bisa sampai 512 karakter.
- Jalur harus dimulai dan diakhiri dengan garis miring (/). Ini dapat berisi beberapa garis miring (/) antara garis miring awal dan akhir (/).
- Anda dapat menggabungkan jalur dan nama/path/name.

Tentukan nama.

```
Iam:
 ResourcePrefix: my-prefix
```

Tentukan jalur.

```
Iam:
 ResourcePrefix: /org/dept/team/project/user/
```

Tentukan jalur dan nama.

```
Iam:
 ResourcePrefix: /org/dept/team/project/user/my-prefix
```

Jika Anda menentukan `/my-prefix`, kesalahan dikembalikan.

```
Iam:
 ResourcePrefix: /my-prefix
```

Kesalahan konfigurasi dikembalikan. Sebuah jalan harus memiliki dua `/` s. Awalan dengan sendirinya tidak dapat memiliki `/` s.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Bagian **LoginNodes**

### Note

Support untuk LoginNodes ditambahkan dalam AWS ParallelCluster versi 3.7.0.

(Opsional) Menentukan konfigurasi untuk kumpulan node login.

```
LoginNodes:
 Pools:
 - Name: string
 Count: integer
 InstanceType: string
 GracetimePeriod: integer
 Image:
 CustomAmi: string
 Ssh:
 KeyName: string
 Networking:
 SubnetIds:
 - string
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
```

```

 - string
 Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string

```

Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.

## **LoginNodes**properti

## **Pools**properti

Mendefinisikan grup node login yang memiliki konfigurasi sumber daya yang sama. Hanya satu kolom yang dapat ditentukan.

```

Pools:
 - Name: string
 Count: integer
 InstanceType: string
 GracetimePeriod: integer
 Image:
 CustomAmi: string
 Ssh:
 KeyName: string
 Networking:
 SubnetIds:
 - string
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string

```

## **Name**(DiperlukanString)

Menentukan nama LoginNodes kolom renang. Ini digunakan untuk menandai LoginNodes sumber daya.



[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

Count(DiperlukanInteger)

Menentukan jumlah node login untuk tetap aktif.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)


InstanceType(DiperlukanString)

Menentukan jenis instans Amazon EC2 yang digunakan untuk node login. Arsitektur tipe instance harus sama dengan arsitektur yang digunakan untuk Slurm InstanceType pengaturan.

[Kebijakan pembaruan:](#) Pengaturan ini dapat diubah jika kumpulan node login dihentikan.

GracetimePeriod(OpsionalInteger)

Menentukan jumlah minimum waktu dalam menit yang berlalu antara pemberitahuan ke pengguna yang masuk bahwa node login akan dinonaktifkan dan peristiwa berhenti yang sebenarnya. Nilai yang valid untuk GracetimePeriod adalah dari 3 hingga 120 menit. Defaultnya adalah 60 menit.

 Note

Peristiwa pemicu melibatkan interaksi antara beberapa AWS layanan. Terkadang, latensi jaringan dan penyebaran informasi mungkin memakan waktu sehingga masa tenggang mungkin memakan waktu lebih lama dari yang diharapkan karena keterlambatan internal dalam layanan. AWS

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

Image(Opsional)

Mendefinisikan konfigurasi gambar untuk node login.

Image:

CustomAmi: *String*

CustomAmi(OpsionalString)

Menentukan AMI kustom yang digunakan untuk menyediakan node login. Jika tidak ditentukan, nilai default ke yang ditentukan dalam. [Bagian HeadNode](#)

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Ssh(Opsional)

Mendefinisikan ssh konfigurasi untuk node login.

```
Ssh:
 KeyName: string
```

### KeyName(OpsionalString)

Menentukan ssh kunci yang digunakan untuk login ke node login. Jika tidak ditentukan, nilai default ke yang ditentukan dalam [Bagian HeadNode](#)

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Networking(Diperlukan)

```
Networking:
 SubnetIds:
 - string
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
```

### SubnetIds(Diperlukan[String])

ID subnet yang ada tempat Anda menyediakan kumpulan node login. Anda hanya dapat menentukan satu subnet.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### SecurityGroups(Opsional[String])

Daftar grup keamanan yang akan digunakan untuk kumpulan node login. Jika tidak ada grup keamanan yang ditentukan, AWS ParallelCluster buat grup keamanan untuk Anda.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### AdditionalSecurityGroups(Opsional[String])

Daftar grup keamanan tambahan yang akan digunakan untuk kumpulan node login.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Iam(Opsional)

Menentukan baik peran instance atau profil instance untuk digunakan pada node login untuk mengganti peran instance default atau profil instance untuk cluster.

```
Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string
```

## InstanceProfile(OpsionalString)

Menentukan profil contoh untuk mengganti profil contoh node login default. Anda tidak dapat menentukan InstanceProfile dan InstanceRole sekaligus. Formatnya adalah `arn:Partition:iam::Account:instance-profile/InstanceProfileName`. Jika ini ditentukan, AdditionalIamPolicies pengaturan InstanceRole dan tidak dapat ditentukan.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## InstanceRole(OpsionalString)

Menentukan peran contoh untuk mengganti peran instance node login default. Anda tidak dapat menentukan InstanceProfile dan InstanceRole sekaligus. Formatnya adalah `arn:Partition:iam::Account:role/RoleName`. Jika ini ditentukan, AdditionalIamPolicies pengaturan S3Access dan tidak dapat ditentukan. Jika ini ditentukan, AdditionalIamPolicies pengaturan InstanceProfile dan tidak dapat ditentukan.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## AdditionalIamPolicies(Opsional)

```
AdditionalIamPolicies:
 - Policy: string
```

Kebijakan IAM Nama Sumber Daya Amazon (ARN).

Menentukan daftar Nama Sumber Daya Amazon (ARN) kebijakan IAM untuk Amazon EC2. Daftar ini dilampirkan ke peran root yang digunakan untuk node login selain izin yang diperlukan oleh AWS ParallelCluster.

Nama kebijakan IAM dan ARN-nya berbeda. Nama tidak bisa digunakan.

Jika ini ditentukan, InstanceRole pengaturan InstanceProfile dan tidak dapat ditentukan. Kami menyarankan Anda menggunakan AdditionalIamPolicies karena AdditionalIamPolicies ditambahkan ke izin yang AWS ParallelCluster diperlukan, dan InstanceRole harus menyertakan semua izin yang diperlukan. Izin yang diperlukan sering berubah dari rilis ke rilis saat fitur ditambahkan.

Tidak ada nilai default.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

Policy(Diperlukan[String])

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Bagian **Monitoring**

(Opsional) Menentukan pengaturan pemantauan untuk cluster.

Monitoring:

Logs:

CloudWatch:

Enabled: *boolean*

RetentionInDays: *integer*

DeletionPolicy: *string*

Rotation:

Enabled: *boolean*

Dashboards:

CloudWatch:

Enabled: *boolean*

DetailedMonitoring: *boolean*

Alarms:

Enabled: *boolean*

[Kebijakan pembaruan: Pengaturan ini tidak dianalisis selama pembaruan.](#)

### **Monitoring**properti

Logs(Opsional)

Pengaturan log untuk cluster.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### CloudWatch(Opsional)

Pengaturan CloudWatch Log untuk cluster.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

Enabled(Diperlukan,Boolean)

Jika `true`, log cluster dialirkan ke CloudWatch Log. Nilai default-nya adalah `true`.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

RetentionInDays(Opsional,Integer)

Jumlah hari untuk menyimpan peristiwa log di CloudWatch Log. Nilai defaultnya adalah 180. Nilai yang didukung adalah 0, 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, dan 3653. Nilai 0 akan menggunakan pengaturan penyimpanan CloudWatch log default, yaitu tidak pernah kedaluwarsa.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

DeletionPolicy(Opsional,String)

Menunjukkan apakah akan menghapus peristiwa CloudWatch log pada Log saat cluster dihapus. Nilai yang mungkin adalah `Delete` dan `Retain`. Nilai default-nya adalah `Retain`.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

Rotation(Opsional)

Pengaturan rotasi log untuk cluster.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

Enabled(Diperlukan,Boolean)

Jika `true`, rotasi log diaktifkan. Nilai default-nya `true`. Ketika file log yang AWS ParallelCluster dikonfigurasi mencapai ukuran tertentu, itu diputar dan satu cadangan dipertahankan. Untuk informasi selengkapnya, lihat [AWS ParallelCluster rotasi log yang dikonfigurasi](#).

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Dashboards(Opsional)

Pengaturan dasbor untuk cluster.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## CloudWatch(Opsional)

Pengaturan CloudWatch dasbor untuk cluster.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

Enabled(Diperlukan,Boolean)

Jika `true`, CloudWatch dasbor diaktifkan. Nilai default-nya adalah `true`.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

## DetailedMonitoring(Opsional,Boolean)

Jika disetel ke `true`, pemantauan terperinci diaktifkan untuk instans EC2 armada komputasi. Saat diaktifkan, konsol Amazon EC2 menampilkan grafik untuk memantau instans pada interval 1 menit. Ada biaya tambahan saat fitur ini diaktifkan. Nilai default-nya `false`.

Untuk informasi lebih lanjut, lihat [Aktifkan atau matikan pemantauan terperinci untuk instans Anda](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

### Note

DetailedMonitoring ditambahkan dimulai dengan AWS ParallelCluster versi 3.6.0.

## Alarms(Opsional)

CloudWatch Alarm untuk cluster.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

Enabled(Opsional)

Jika `true`, CloudWatch Alarm untuk cluster akan dibuat. Nilai default-nya adalah `true`.

[Kebijakan pembaruan: Pengaturan ini dapat diubah selama pembaruan.](#)

**Note**

Dimulai dengan AWS ParallelCluster versi 3.8.0, alarm berikut dibuat untuk Head Node: EC2 Health Check, CPU/Memori/pemanfaatan disk dan alarm komposit termasuk yang lainnya.

## Bagian **Tags**

(Opsional), Array Mendefinisikan tag yang digunakan oleh AWS CloudFormation dan disebarkan ke semua sumber daya cluster. Untuk informasi selengkapnya, lihat [tag AWS CloudFormation sumber daya](#) di Panduan AWS CloudFormation Pengguna.

**Tags:**

- **Key:** *string*
- Value:** *string*

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

### **Tags**properti

Key(Diperlukan,String)

Mendefinisikan nama tag.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

Value(Diperlukan,String)

Mendefinisikan nilai tag.

Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.

## Bagian **AdditionalPackages**

(Opsional) Digunakan untuk mengidentifikasi paket tambahan untuk diinstal.

**AdditionalPackages:**

**IntelSoftware:**

**IntelHpcPlatform:** *boolean*

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## IntelSoftware

(Opsional) Mendefinisikan konfigurasi untuk solusi pilihan Intel.

```
IntelSoftware:
 IntelHpcPlatform: boolean
```

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## IntelSoftwareproperti

IntelHpcPlatform(Opsional,Boolean)

Jika `true`, menunjukkan bahwa [perjanjian lisensi pengguna akhir](#) untuk Intel Parallel Studio diterima. Hal ini menyebabkan Intel Parallel Studio diinstal pada node kepala dan dibagikan dengan node komputasi. Ini menambahkan beberapa menit ke waktu yang dibutuhkan node kepala untuk bootstrap. `IntelHpcPlatform` Pengaturan hanya didukung pada CentOS 7.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

## Bagian DirectoryService

### Note

Support untuk `DirectoryService` ditambahkan di AWS ParallelCluster versi 3.1.1.

(Opsional) Pengaturan layanan direktori untuk cluster yang mendukung beberapa akses pengguna.

AWS ParallelCluster mengelola izin yang mendukung akses beberapa pengguna ke cluster dengan Active Directory (AD) melalui Lightweight Directory Access Protocol (LDAP) yang didukung oleh [Daemon Layanan Keamanan Sistem](#) (SSSD). Untuk informasi lebih lanjut, lihat [Apa itu AWS Directory Service?](#) di Panduan Administrasi AWS Directory Service .

Kami menyarankan Anda menggunakan LDAP melalui TLS/SSL (disingkat LDAPS) untuk memastikan bahwa setiap informasi yang berpotensi sensitif ditransmisikan melalui saluran terenkripsi.

```
DirectoryService:
```



```
DomainName: string
DomainAddr: string
PasswordSecretArn: string
DomainReadOnlyUser: string
LdapTlsCaCert: string
LdapTlsReqCert: string
LdapAccessFilter: string
GenerateSshKeysForUsers: boolean
AdditionalSssdConfigs: dict
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## DirectoryServiceproperty

### Note

Jika Anda berencana untuk menggunakan AWS ParallelCluster dalam satu subnet tanpa akses internet, lihat [AWS ParallelCluster dalam subnet tunggal tanpa akses internet](#) persyaratan tambahan.

## DomainName(Diperlukan,String)

Domain Active Directory (AD) yang Anda gunakan untuk informasi identitas.

DomainName menerima format Fully Qualified Domain Name (FQDN) dan LDAP Distinguished Name (DN).

- Contoh FQDN: corp.*example*.com
- Contoh LDAP DN: DC=*corp*,DC=*example*,DC=*com*

Properti ini sesuai dengan parameter sssd-ldap yang dipanggil. `ldap_search_base`

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## DomainAddr(Diperlukan,String)

URI atau URI yang mengarah ke pengontrol domain AD yang digunakan sebagai server LDAP. URI sesuai dengan parameter SSSD-LDAP yang dipanggil. `ldap_uri` Nilainya bisa berupa string URI yang dipisahkan koma. Untuk menggunakan LDAP, Anda harus menambahkan `ldap://` ke awal setiap URI.

## Nilai contoh:

```
ldap://192.0.2.0,ldap://203.0.113.0 # LDAP
ldaps://192.0.2.0,ldaps://203.0.113.0 # LDAPS without support for certificate
 verification
ldaps://abcdef01234567890.corp.example.com # LDAPS with support for certificate
 verification
192.0.2.0,203.0.113.0 # AWS ParallelCluster uses LDAPS by
 default
```

Jika Anda menggunakan LDAPS dengan verifikasi sertifikat, URI harus menjadi nama host.

Jika Anda menggunakan LDAPS tanpa verifikasi sertifikat atau LDAP, URI dapat berupa nama host atau alamat IP.

Gunakan LDAP melalui TLS/SSL (LDAPS) untuk menghindari transmisi kata sandi dan informasi sensitif lainnya melalui saluran yang tidak terenkripsi. Jika AWS ParallelCluster tidak menemukan protokol, itu `ldaps://` menambah awal setiap URI atau nama host.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## PasswordSecretArn(Diperlukan,String)

Nama Sumber Daya Amazon (ARN) dari AWS Secrets Manager rahasia yang berisi kata sandi teks biasa `DomainReadOnlyUser`. Isi rahasia sesuai dengan parameter `SSSD-LDAP` yang disebut `ldap_default_authtok`

### Note

Saat membuat rahasia menggunakan AWS Secrets Manager konsol pastikan untuk memilih “Jenis rahasia lainnya”, pilih teks biasa, dan hanya sertakan teks kata sandi dalam rahasia.

Untuk informasi lebih lanjut tentang cara menggunakan AWS Secrets Manager untuk membuat rahasia, lihat [Buat AWS Secrets Manager Rahasia](#)

Klien LDAP menggunakan kata sandi untuk mengautentikasi ke domain AD sebagai `DomainReadOnlyUser` saat meminta informasi identitas.

Jika pengguna memiliki izin untuk [DescribeSecret](#), PasswordSecretArn divalidasi. PasswordSecretArn valid jika rahasia yang ditentukan ada. Jika kebijakan IAM pengguna tidak disertakan DescribeSecret, PasswordSecretArn tidak divalidasi dan pesan peringatan ditampilkan. Untuk informasi selengkapnya, lihat [Kebijakan AWS ParallelCluster pcluster pengguna dasar](#).

Ketika nilai rahasia berubah, kluster tidak diperbarui secara otomatis. Untuk memperbarui cluster untuk nilai rahasia baru, Anda harus menghentikan armada komputasi dengan [the section called "pcluster update-compute-fleet"](#) perintah dan kemudian menjalankan perintah berikut dari dalam node kepala.

```
$ sudo /opt/parallelcluster/scripts/directory_service/
update_directory_service_password.sh
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

DomainReadOnlyUser(Diperlukan,String)

Identitas yang digunakan untuk menanyakan domain AD untuk informasi identitas saat mengautentikasi login pengguna cluster. Ini sesuai dengan parameter SSSD-LDAP yang disebut. `ldap_default_bind_dn` Gunakan informasi identitas iklan Anda untuk nilai ini.

Tentukan identitas dalam formulir yang diperlukan oleh klien LDAP tertentu yang ada di node:

- MicrosoftAD:

```
cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

- SimpleAD:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

LdapTlsCaCert(Opsional,String)

Jalur absolut ke bundel sertifikat yang berisi sertifikat untuk setiap otoritas sertifikasi dalam rantai sertifikasi yang mengeluarkan sertifikat untuk pengontrol domain. Ini sesuai dengan parameter SSSD-LDAP yang disebut. `ldap_tls_cacert`

Bundel sertifikat adalah file yang terdiri dari rangkaian sertifikat berbeda dalam format PEM, juga dikenal sebagai format DER Base64 di Windows. Ini digunakan untuk memverifikasi identitas pengontrol domain AD yang bertindak sebagai server LDAP.

AWS ParallelCluster tidak bertanggung jawab atas penempatan awal sertifikat ke node. Sebagai administrator cluster, Anda dapat mengonfigurasi sertifikat di node kepala secara manual setelah cluster dibuat atau Anda dapat menggunakan [skrip bootstrap](#). Atau, Anda dapat menggunakan Amazon Machine Image (AMI) yang menyertakan sertifikat yang dikonfigurasi pada node kepala.

[Simple AD](#) tidak menyediakan dukungan LDAPS. Untuk mempelajari cara mengintegrasikan direktori Simple AD dengan AWS ParallelCluster, lihat [Cara mengonfigurasi titik akhir LDAPS untuk Simple AD di Blog Keamanan.AWS](#)

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

#### LdapTlsReqCert(Opsional,String)

Menentukan pemeriksaan apa yang akan dilakukan pada sertifikat server dalam sesi TLS. Ini sesuai dengan parameter SSSD-LDAP yang disebut. `ldap_tls_reqcert`

Nilai yang valid: `never,allow,try,demand,danhard`.

`never,allow`, dan `try` aktifkan koneksi untuk melanjutkan bahkan jika masalah dengan sertifikat ditemukan.

`demand` dan `hard` memungkinkan komunikasi untuk melanjutkan jika tidak ada masalah dengan sertifikat yang ditemukan.

Jika administrator klaster menggunakan nilai yang tidak memerlukan validasi sertifikat untuk berhasil, pesan peringatan akan dikembalikan ke administrator. Untuk alasan keamanan, kami menyarankan Anda untuk tidak menonaktifkan verifikasi sertifikat.

Nilai default-nya adalah `hard`.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

#### LdapAccessFilter(Opsional,String)

Menentukan filter untuk membatasi akses direktori ke subset pengguna. Properti ini sesuai dengan parameter SSSD-LDAP yang dipanggil. `ldap_access_filter` Anda dapat menggunakannya untuk membatasi kueri pada iklan yang mendukung sejumlah besar pengguna.

Filter ini dapat memblokir akses pengguna ke cluster. Namun, itu tidak memengaruhi kemampuan ditemukan pengguna yang diblokir.

Jika properti ini disetel, parameter SSSD `access_provider` diatur ke `ldap` internal oleh AWS ParallelCluster dan tidak boleh dimodifikasi oleh [DirectoryService/AdditionalSssdConfigs](#) pengaturan.

Jika properti ini dihilangkan dan akses pengguna yang disesuaikan tidak ditentukan di [DirectoryService/AdditionalSssdConfigs](#), semua pengguna di direktori dapat mengakses klaster.

Contoh:

```
"! (cn=SomeUser*)" # denies access to every user with alias starting with "SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias starting with "SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

`GenerateSshKeysForUsers`(Opsional, Boolean)

Mendefinisikan apakah AWS ParallelCluster menghasilkan kunci SSH untuk pengguna cluster segera setelah otentikasi awal mereka pada node kepala.

Jika disetel ke `true`, kunci SSH dihasilkan dan disimpan ke `USER_HOME_DIRECTORY/.ssh/id_rsa`, jika tidak ada, untuk setiap pengguna setelah otentikasi pertama mereka di node kepala.

Untuk pengguna yang belum diautentikasi pada node kepala, otentikasi pertama dapat terjadi dalam kasus berikut:

- Pengguna masuk ke node kepala untuk pertama kalinya dengan kata sandinya sendiri.
- Di node kepala, sudoer beralih ke pengguna untuk pertama kalinya: `su USERNAME`
- Di node kepala, sudoer menjalankan perintah sebagai pengguna untuk pertama kalinya: `su -u USERNAME COMMAND`

Pengguna dapat menggunakan kunci SSH untuk login berikutnya ke node kepala cluster dan node komputasi. Dengan AWS ParallelCluster, login kata sandi ke node komputasi cluster dinonaktifkan berdasarkan desain. Jika pengguna belum masuk ke node kepala, kunci SSH tidak dihasilkan dan pengguna tidak akan dapat masuk untuk menghitung node.

Nilai default-nya true.

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

#### AdditionalSssdConfigs(Opsional,Dict)

Dikte pasangan nilai kunci yang berisi parameter dan nilai SSSD untuk ditulis ke file konfigurasi SSSD pada instance cluster. Untuk deskripsi lengkap tentang file konfigurasi SSSD, lihat halaman manual on-instance untuk SSSD dan file konfigurasi terkait.

Parameter dan nilai SSSD harus kompatibel dengan AWS ParallelCluster konfigurasi SSSD seperti yang dijelaskan dalam daftar berikut.

- `id_provider` diatur ke `ldap` internal oleh AWS ParallelCluster dan tidak boleh dimodifikasi.
- `access_provider` diatur ke `ldap` internal oleh AWS ParallelCluster ketika [DirectoryService/LdapAccessFilter](#) ditentukan, dan pengaturan ini tidak boleh diubah.

Jika [DirectoryService/LdapAccessFilter](#) dihilangkan, `access_provider` spesifikasinya juga dihilangkan. Misalnya, jika Anda menyetel `access_provider` ke `simple` in [AdditionalSssdConfigs](#), maka [DirectoryService](#)/tidak [LdapAccessFilter](#) boleh ditentukan.

Cuplikan konfigurasi berikut adalah contoh konfigurasi yang valid untuk.

#### AdditionalSssdConfigs

Contoh ini memungkinkan tingkat debug untuk log SSSD, membatasi basis pencarian ke unit organisasi tertentu, dan menonaktifkan caching kredensial.

```
DirectoryService:
 ...
AdditionalSssdConfigs:
 debug_level: "0xFFF0"
 ldap_search_base: OU=Users,OU=CORP,DC=corp,DC=example,DC=com
 cache_credentials: False
```

Contoh ini menentukan konfigurasi [simple](#) `access_provider` SSSD. Pengguna dari `EngineeringTeam` disediakan akses ke direktori. [DirectoryService](#)/tidak [LdapAccessFilter](#) boleh diatur dalam kasus ini.

```
DirectoryService:
```

```
...
AdditionalSssdConfigs:
 access_provider: simple
 simple_allow_groups: EngineeringTeam
```

[Kebijakan pembaruan: Armada komputasi harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)

## Bagian **DeploymentSettings**

### Note

DeploymentSettings ditambahkan dimulai dengan AWS ParallelCluster versi 3.4.0.

(Opsional) Menentukan konfigurasi pengaturan penyebaran.

```
DeploymentSettings:
 LambdaFunctionsVpcConfig:
 SecurityGroupIds
 - string
 SubnetIds
 - string
 DisableSudoAccessForDefaultUser: Boolean
 DefaultUserHome: string # 'Shared' or 'Local'
```

## **DeploymentSettings** properti

### **LambdaFunctionsVpcConfig**

(Opsional) Menentukan AWS Lambda fungsi konfigurasi VPC. Untuk informasi selengkapnya, lihat [AWS Lambda Konfigurasi VPC di AWS ParallelCluster.](#)

```
LambdaFunctionsVpcConfig:
 SecurityGroupIds
 - string
 SubnetIds
 - string
```

## LambdaFunctionsVpcConfig properties

SecurityGroupIds(Diperlukan,[String])

Daftar ID grup keamanan Amazon VPC yang dilampirkan ke fungsi Lambda.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

SubnetIds(Diperlukan,[String])

Daftar ID subnet yang dilampirkan ke fungsi Lambda.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

### Note

Subnet dan grup keamanan harus berada dalam VPC yang sama.

DisableSudoAccessForDefaultUser properti

### Note

Opsi konfigurasi ini hanya didukung dengan Slurm Clusters.

(Opsional) Jika True, hak istimewa sudo dari Pengguna default akan dinonaktifkan. Ini berlaku untuk semua node di cluster.

```
Main DeploymentSettings section in config yaml(applyes to HN, CF and LN)
DeploymentSettings:
 DisableSudoAccessForDefaultUser: True
```

Untuk memperbarui nilai `DisableSudoAccessForDefaultUser`, Anda harus menghentikan armada komputasi dan semua node login.

[Kebijakan pembaruan: Armada komputasi dan node login harus dihentikan agar pengaturan ini diubah untuk pembaruan.](#)



## DefaultUserHomeproperty

Ketika diatur ke`Shared`, cluster akan menggunakan pengaturan default dan berbagi direktori pengguna default di seluruh cluster oleh `/home/<default user>`.

Ketika diatur ke`Local`, node kepala, node login, dan node komputasi masing-masing akan memiliki direktori pengguna default lokal yang terpisah yang disimpan di `local/home/<default user>`.

## Membangun file konfigurasi gambar

AWS ParallelCluster versi 3 menggunakan file YAMAL 1.1 untuk membangun parameter konfigurasi gambar. Harap konfirmasi bahwa lekukan sudah benar untuk mengurangi kesalahan konfigurasi. Untuk informasi selengkapnya, lihat spesifikasi YAMAL 1.1 di <https://yaml.org/spec/1.1/>

File konfigurasi ini digunakan untuk menentukan bagaimana AWS ParallelCluster AMI kustom Anda dibuat menggunakan EC2 Image Builder. Proses pembuatan AMI khusus dipicu menggunakan `pcluster build-image` perintah. Untuk beberapa contoh file konfigurasi, lihat [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test\\_imagebuilder\\_schema/test\\_imagebuilder\\_schema](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema).

### Topik

- [Membangun properti file konfigurasi gambar](#)
- [Bagian Build](#)
- [Bagian Image](#)
- [Bagian DeploymentSettings](#)

## Membangun properti file konfigurasi gambar

### Region(Opsional,String)

Menentukan Wilayah AWS untuk `build-image` operasi. Misalnya, `us-east-2`.

## Bagian **Build**

(Wajib) Menentukan konfigurasi di mana gambar akan dibangun.

```
Build:
 Imds:
 ImdsSupport: string
```

```
InstanceType: string
SubnetId: string
ParentImage: string
Iam:
 InstanceRole: string
 InstanceProfile: string
 CleanupLambdaRole: string
 AdditionalIamPolicies:
 - Policy: string
 PermissionsBoundary: string
Components:
 - Type: string
 Value: string
Tags:
 - Key: string
 Value: string
SecurityGroupIds:
 - string
UpdateOsPackages:
 Enabled: boolean
```

## Buildproperti

### InstanceType(Diperlukan,String)

Menentukan jenis contoh untuk contoh yang digunakan untuk membangun gambar.

### SubnetId(Opsional,String)

Menentukan ID dari subnet yang ada di mana untuk menyediakan contoh untuk membangun gambar. Subnet yang disediakan membutuhkan akses internet.

#### Warning

pcluster build-imagemenggunakan VPC default. Jika VPC default telah dihapus, mungkin dengan menggunakan AWS Control Tower atau AWS Landing Zone, maka ID subnet harus ditentukan.

### ParentImage(Diperlukan,String)

Menentukan gambar dasar. Gambar induk dapat berupa AWS ParallelCluster AMI non atau AWS ParallelCluster AMI resmi untuk versi yang sama. Anda

tidak dapat menggunakan AMI AWS ParallelCluster resmi atau kustom dari versi yang berbeda AWS ParallelCluster. Formatnya harus berupa ARN gambar `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion` atau ID AMI. `ami-12345678`

`SecurityGroupIds`(Opsional,[String])

Menentukan daftar ID grup keamanan untuk gambar.

## Imds

### Imdsproperti

(Opsional) Menentukan pengaturan EC2 ImageBuilder build dan test instance metadata service (IMDS).

`Imds`:

`ImdsSupport`: *string*

`ImdsSupport`(Opsional,String)

Menentukan versi IMDS mana yang didukung dalam instance ImageBuilder build dan pengujian EC2. Nilai yang didukung adalah `v2.0` dan `v1.0`. Nilai default-nya adalah `v2.0`.

Jika `ImdsSupport` disetel ke `v1.0`, IMDSv1 dan IMDSv2 didukung.

Jika `ImdsSupport` disetel ke `v2.0`, hanya IMDSv2 yang didukung.

Untuk informasi selengkapnya, lihat [Menggunakan IMDSv2](#) di Panduan Pengguna EC2 untuk instance Linux.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

#### Note

Dimulai dengan AWS ParallelCluster versi 3.7.0, nilai `ImdsSupport` defaultnya adalah `v2.0`. Kami menyarankan Anda mengatur `v2.0` dan `ImdsSupport` mengganti IMDSv1 dengan IMDSv2 dalam panggilan tindakan kustom Anda.

Support for [Imds/ImdsSupport](#) ditambahkan dengan AWS ParallelCluster versi 3.3.0.

## Iam

### Iamproperti

(Opsional) Menentukan resource IAM untuk build image.

```
Iam:
 InstanceRole: string
 InstanceProfile: string
 CleanupLambdaRole: string
 AdditionalIamPolicies:
 - Policy: string
 PermissionsBoundary: string
```

### InstanceProfile(Opsional,String)

Menentukan profil instance untuk mengganti profil instance default untuk instance EC2 Image Builder. InstanceProfileInstanceRole dan AdditionalIamPolicies tidak dapat ditentukan bersama. Formatnya adalah `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

### InstanceRole(Opsional,String)

Menentukan peran instance untuk mengganti peran instance default untuk instance EC2 Image Builder. InstanceProfileInstanceRole dan AdditionalIamPolicies tidak dapat ditentukan bersama. Formatnya adalah `arn:Partition:iam::Account:role/RoleName`.

### CleanupLambdaRole(Opsional,String)

ARN dari peran IAM yang akan digunakan untuk AWS Lambda fungsi yang mendukung sumber daya AWS CloudFormation kustom yang menghapus artefak build saat penyelesaian build. Lambda perlu dikonfigurasi sebagai prinsipal yang diizinkan untuk mengambil peran. Formatnya adalah `arn:Partition:iam::Account:role/RoleName`.

### AdditionalIamPolicies(Opsional)

Menentukan kebijakan IAM tambahan untuk dilampirkan ke instans EC2 Image Builder yang digunakan untuk menghasilkan AMI kustom.

```
AdditionalIamPolicies:
 - Policy: string
```

Policy(Opsional,[String])

Daftar kebijakan IAM. Formatnya adalah

arn:*Partition*:iam::*Account*:policy/*PolicyName*.

PermissionsBoundary(Opsional,String)

ARN dari kebijakan IAM untuk digunakan sebagai batas izin untuk semua peran yang dibuat oleh. AWS ParallelCluster Untuk informasi selengkapnya tentang batas izin IAM, silakan lihat batas [Izin untuk entitas IAM](#) di Panduan Pengguna IAM. Formatnya adalah arn:*Partition*:iam::*Account*:policy/*PolicyName*.

## Components

**Components**properti

(Opsional) Menentukan ImageBuilder komponen EC2 yang akan digunakan selama proses pembuatan AMI selain komponen yang disediakan secara default oleh. AWS ParallelCluster Komponen tersebut dapat digunakan untuk menyesuaikan proses pembuatan AMI. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Kustomisasi AMI](#).

Components:

- Type: *string*
- Value: *string*

Type(Opsional,String)

Menentukan jenis pasangan tipe-nilai untuk komponen. Jenis bisa arn atau script.

Value(Opsional,String)

Menentukan nilai pasangan tipe-nilai untuk komponen. Ketika jenisnya arn, ini adalah ARN dari komponen EC2 Image Builder. Saat ketiknyascript, ini adalah tautan https atau s3 yang menunjuk ke skrip yang akan digunakan saat membuat komponen EC2 Image Builder.

## Tags

**Tags**properti

(Opsional) Menentukan daftar tag yang akan diatur dalam sumber daya yang digunakan untuk membangun AMI.

**Tags:**

- **Key:** *string*
- Value:** *string*

Key(Optional,String)

Mendefinisikan nama tag.

Value(Optional,String)

Mendefinisikan nilai tag.

## UpdateOsPackages

### UpdateOsPackagesproperti

(Optional) Menentukan apakah sistem operasi diperbarui sebelum menginstal tumpukan AWS ParallelCluster perangkat lunak.

**UpdateOsPackages:**

- Enabled:** *boolean*

Enabled(Optional,Boolean)

Jika `true`, OS diperbarui dan di-boot ulang sebelum menginstal perangkat lunak. AWS ParallelCluster Nilai default-nya `false`.

**Note**

Ketika `UpdateOsPackages` diaktifkan, semua paket OS yang tersedia diperbarui, termasuk kernel. Sebagai pelanggan, Anda bertanggung jawab untuk memverifikasi bahwa pembaruan kompatibel dengan dependensi AMI yang tidak disertakan dalam pembaruan.

Misalnya, Anda sedang membangun AMI untuk AWS ParallelCluster versi X.0 yang dikirimkan dengan kernel versi Y.0 dan beberapa versi komponen Z.0. Misalkan pembaruan yang tersedia mencakup versi kernel yang diperbarui Y.1 tanpa pembaruan ke komponen Z.0. Sebelum Anda mengaktifkan `UpdateOsPackages`, Anda bertanggung jawab untuk memverifikasi bahwa komponen Z.0 mendukung kernel Y.1.

## Bagian Image

(Opsional) Mendefinisikan properti image untuk build image.

```
Image:
 Name: string
 RootVolume:
 Size: integer
 Encrypted: boolean
 KmsKeyId: string
 Tags:
 - Key: string
 Value: string
```

### Imageproperti

Name(Opsional,String)

Menentukan nama AMI. Jika tidak ditentukan, nama yang digunakan saat memanggil [pcluster build-image](#) perintah digunakan.

### Tags

#### Tagsproperti

(Opsional) Menentukan pasangan kunci-nilai untuk gambar.

```
Tags:
- Key: string
 Value: string
```

Key(Opsional,String)

Mendefinisikan nama tag.

Value(Opsional,String)

Mendefinisikan nilai tag.

## RootVolume

### RootVolumeproperti

(Opsional) Menentukan properti volume root untuk gambar.

```
RootVolume:
 Size: integer
 Encrypted: boolean
 KmsKeyId: string
```

### Size(Opsional,Integer)

Menentukan ukuran volume root untuk gambar, di GiB. Ukuran default adalah ukuran [ParentImage](#) plus 27 GiB.

### Encrypted(Opsional,Boolean)

Menentukan apakah volume dienkripsi. Nilai default-nya adalah `false`.

### KmsKeyId(Opsional,String)

Menentukan ARN dari kunci yang AWS KMS digunakan untuk mengenkripsi volume. Formatnya adalah `"arn:Partition:kms:Region:Account:key/KeyId"`.

## Bagian DeploymentSettings

(Opsional) Menentukan konfigurasi pengaturan penyebaran.

```
DeploymentSettings:
 LambdaFunctionsVpcConfig:
 SecurityGroupIds
 - string
 SubnetIds
 - string
```

### DeploymentSettingsproperti

### LambdaFunctionsVpcConfig

(Opsional) Menentukan AWS Lambda fungsi konfigurasi VPC. Untuk informasi selengkapnya, lihat [AWS LambdaKonfigurasi VPC diAWS ParallelCluster](#).



**LambdaFunctionsVpcConfig:****SecurityGroupIds**

- *string*

**SubnetIds**

- *string*

**LambdaFunctionsVpcConfig properties****SecurityGroupIds(Diperlukan,[String])**


Daftar ID grup keamanan Amazon VPC yang dilampirkan ke fungsi Lambda.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)


**SubnetIds(Diperlukan,[String])**

Daftar ID subnet yang dilampirkan ke fungsi Lambda.

[Kebijakan pembaruan: Jika pengaturan ini diubah, pembaruan tidak diizinkan.](#)

** Note**

Subnet dan grup keamanan harus berada dalam VPC yang sama.

** Note**

DeploymentSettingsditambahkan dimulai dengan AWS ParallelCluster versi 3.4.0.

## AWS ParallelClusterReferensi API

Bagian ini memberikan deskripsi, sintaks, dan contoh penggunaan untuk masing-masingAWS ParallelClusterTindakan API.

**Topik**

- [BuildImage](#)
- [createCluster](#)
- [HapusCluster](#)

- [deleteClusterInstances](#)
- [DeleteImage](#)
- [DescribeCluster](#)
- [describeClusterInstances](#)
- [describeComputeFleet](#)
- [DescribeImage](#)
- [getClusterLogAcara](#)
- [getClusterStackAcara](#)
- [getImageLogAcara](#)
- [getImageStackAcara](#)
- [ListClusters](#)
- [listClusterLogAliran](#)
- [listImageLogAliran](#)
- [ListImages](#)
- [listOfficialImages](#)
- [UpdateCluster](#)
- [updateComputeFleet](#)

## BuildImage

Buat kustomAWS ParallelClustergambar dalam sebuahWilayah AWS.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
POST /v3/images/custom
```

```
{
 "imageConfiguration": "string",
 "imageId": "string",
 "dryrun": boolean,
 "region": "string",
 "rollbackOnFailure": boolean,
 "supressValidators": ["string"],
 "validationFailureLevel": "string"
}
```

## Isi permintaan

### ImageConfiguration

Konfigurasi gambar sebagai dokumen YAMAL.

Tipe string

Wajib: Ya

### imageId

ID gambar yang akan dibangun.

Tipe string

Wajib: Ya

### dryrun

Jika disetel ke `true`, hanya melakukan validasi permintaan tanpa membuat sumber daya apa pun. Gunakan parameter ini untuk memvalidasi konfigurasi gambar. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

### wilayah

TheWilayah AWS bahwa Anda menjalankan perintah untuk membangun gambar.

Jenis: string

Wajib: Tidak

## rollbackOnFailure

Jika disetel ke `true`, rollback tumpukan gambar terjadi jika gambar gagal dibuat. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

## SuppressValidator

Identifikasi satu atau beberapa validator konfigurasi untuk ditekan.

Jenis: daftar string

Format: (ALL | type:[A-Za-z0-9]+)

Wajib: Tidak

## validationFailureLevel

Level validasi minimum yang menyebabkan build image gagal. Default-nya adalah ERROR.

Tipe: string

Nilai valid: INFO | WARNING | ERROR

Wajib: Tidak

## Sintaksis Respons

```
{
 "image": {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS"
 },
}
```

```
"validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
]
}
```

## Isi respons

### gambar

#### imageId

ID gambar.

Tipe: string

#### cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

#### cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

#### ec2AmiInfo

##### ami\_id

ID AMI EC2.

Tipe: string

imageBuildStatus

Status pembuatan gambar.

Tipe: string

Nilai yang valid: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

wilayah

TheWilayah AWSb bahwa gambar dibangun.

Tipe: string

versi

TheAWS ParallelClusterversi yang digunakan untuk membangun gambar.

Tipe: string

ValidationMessages

Daftar pesan dengan tingkat validasi lebih rendah darivalidationFailureLevel. Daftar pesan dikumpulkan selama validasi konfigurasi.

id

ID validator.

Tipe: string

level

Tingkat validasi.

Tipe: string

Nilai yang valid: INFO | WARNING | ERROR

pesan

Pesan validasi.

Tipe: string

jenis

Jenis validator.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ build_image(custom-image-id, custom-image-config.yaml)
```

#### 200 Respon

```
{
 'image': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/custom-image-id/711b76b0-af81-11ec-a29f-0ee549109f1f',
 'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
 'image_build_status': 'BUILD_IN_PROGRESS',
 'image_id': 'custom-image-id',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
}
```

## createCluster

Buat cluster terkelola diWilayah AWS.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)

- [Contoh](#)

## Sintaksis Permintaan

```
POST /v3/clusters
{
 "clusterName": "string",
 "clusterConfiguration": "string",
 "dryrun": boolean,
 "region": "string",
 "rollbackOnFailure", boolean,
 "suppressValidators": ["string"],
 "validationFailureLevel": "string"
}
```

## Isi permintaan

### Konfigurasi Cluster

Konfigurasi cluster sebagai dokumen YAMAL.

Tipe string

Wajib: Ya

### ClusterName

Nama cluster yang akan dibuat.

Nama harus dimulai dengan karakter alfabet. Nama dapat memiliki hingga 60 karakter. Jika Slurm kuantansi diaktifkan, nama dapat memiliki hingga 40 karakter.

Tipe string

Wajib: Ya

### dryrun

Jika disetel ke `true`, hanya melakukan validasi permintaan tanpa membuat sumber daya apa pun. Gunakan parameter ini untuk memvalidasi konfigurasi cluster. Defaultnya adalah `false`.

Tipe: boolean



Wajib: Tidak

wilayah

TheWilayah AWSbaha cluster berada di.

Jenis: string

Wajib: Tidak

rollbackOnFailure

Jika disetel ke `true`, rollback tumpukan cluster terjadi jika cluster gagal membuat. Defaultnya adalah `true`.

Tipe: boolean

Wajib: Tidak

SuppressValidator

Identifikasi satu atau beberapa validator konfigurasi untuk ditekan.

Jenis: daftar string

Format: (ALL | type : [A-Za-z0-9]+)

Wajib: Tidak

validationFailureLevel

Tingkat validasi minimum yang menyebabkan cluster create gagal. Default-nya adalah ERROR.

Tipe: string

Nilai valid: INFO | WARNING | ERROR

Wajib: Tidak

## Sintaksis Respons

```
{
 "cluster": {
 "clusterName": "string",
```

```
"region": "string",
"version": "string",
"cloudformationStackArn": "string",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"clusterStatus": "CREATE_IN_PROGRESS",
"scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
},
"validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
]
}
```

## Isi respons

### ClusterName

Nama cluster.

Tipe: string

### cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

### cloudformationStackStatus

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |

UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
 UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
 UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## Status Cluster

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
 | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
 UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## wilayah

TheWilayah AWSbaha cluster dibuat di.

Tipe: string

## penjadwal

### metadata

#### Metadata penjadwal

##### nama

Nama penjadwal.

Tipe: string

##### versi

Versi penjadwal.

Tipe: string

## jenis

Jenis penjadwal.

Tipe: string

## versi

TheAWS ParallelClusterversi yang digunakan untuk membuat cluster.

Tipe: string

## validation\_messages

Daftar pesan dengan tingkat validasi lebih rendah dari `validationFailureLevel`. Daftar pesan dikumpulkan selama validasi konfigurasi.

id

ID validator.

Tipe: string

level

Tipe: string

Nilai yang valid: INFO | WARNING | ERROR

pesan

Pesan validasi.

Tipe: string

jenis

Jenis validator.

Tipe: string

## Contoh

### Python

Permintaan

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

200 Respon

```
{
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster-3x/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
```

```
'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
'cluster_name': 'cluster-3x',
'cluster_status': 'CREATE_IN_PROGRESS',
'region': 'us-east-1',
'scheduler': {
 'type': 'slurm'
},
'version': '3.2.1'
}
}
```

## HapusCluster

Memulai menghapus cluster.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
DELETE /v3/clusters/{clusterName}
{
 "region": "string"
}
```

### Isi permintaan

ClusterName

Nama klaster.

Tipe string

Wajib: Ya

## wilayah

TheWilayah AWSbahwa cluster dihapus di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "clusterStatus": "DELETE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
 }
}
```

## Isi respons

### klaster

Daftar instance cluster

ClusterName

Nama sebuah cluster.

Tipe: string

cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

cloudformationStackStatus

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

Status Cluster

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

wilayah

TheWilayah AWSbaha cluster dibuat di.

Tipe: string

penjadwal

metadata

Metadata penjadwal.

nama

Nama penjadwal.

Tipe: string

versi

Versi penjadwal

Tipe: string

## jenis

Jenis penjadwal.

Tipe: string

## versi

TheAWS ParallelClusterversi yang digunakan untuk membuat cluster.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ delete_cluster(cluster_name_3x)
```

#### 200 Respon

```
{
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cloudformation_stack_status': 'DELETE_IN_PROGRESS',
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'DELETE_IN_PROGRESS',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
}
```

## deleteClusterInstances

Memulai penghentian paksa dari semua node komputasi cluster. Tindakan ini tidak mendukungAWS Batchcluster.

### Topik

- [Sintaksis Permintaan](#)



- [Isi permintaan](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
DELETE /v3/clusters/{clusterName}/instances
{
 "force": boolean,
 "region": "string"
}
```

### Isi permintaan

#### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

#### kekuatan

Jika disetel ke `true`, paksa penghapusan ketika cluster dengan nama yang diberikan tidak ditemukan. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

#### wilayah

TheWilayah AWS bahwa cluster berada di.

Jenis: string

Wajib: Tidak

### Isi respons

Tidak ada

## Contoh

### Python

#### Permintaan

```
$ delete_cluster_instances(cluster_name_3x)
```

200 Respon

Tidak ada

## DeletelImage

Memulai menghapus kustomAWS ParallelClustergambar.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
DELETE /v3/images/custom/{imageId}
{
 "force": boolean,
 "region": "string"
}
```

### Isi permintaan

#### imageId

ID gambar.

Tipe string

Wajib: Ya

kekuatan

Jika disetel ke `true`, paksa penghapusan AMI. Gunakan parameter ini jika ada instance yang menggunakan AMI atau jika AMI dibagikan. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

wilayah

TheWilayah AWS bahwa gambar itu dibuat di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "image": {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS"
 }
}
```

## Isi respons

gambar

cloudformationStackArn

Nama sumber daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

## cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2AmiInfo

AmiID

ID AMI EC2.

Tipe: string

## imageBuildStatus

Status pembuatan gambar.

Tipe: string

Nilai yang valid: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

## imageId

ID gambar.

Tipe: string

## wilayah

TheWilayah AWSbahwa gambar dibuat di.

Tipe: string

## versi

TheAWS ParallelClusterversi yang digunakan untuk membangun gambar.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ delete_image(custom-image-id)
```

#### 200 Respon

```
{
 'image': {
 'image_build_status': 'DELETE_IN_PROGRESS',
 'image_id': 'custom-image-id',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
}
```

## DescribCluster

Dapatkan informasi rinci tentang cluster yang ada.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/clusters/{clusterName}
{
```

```
"region": "string"
}
```

## Isi permintaan

### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

### wilayah

TheWilayah AWSbaha cluster berada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

### Note

`failureReason` telah berubah menjadi `failures` Dimulai dengan AWS ParallelCluster versi 3.5.0.

```
{
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudFormationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
}
```

```
},
"cloudformationStackArn": "string",
"creationTime": "2019-08-24T14:15:22Z",
"lastUpdatedTime": "2019-08-24T14:15:22Z",
"clusterConfiguration": {
 "url": "string"
},
"computeFleetStatus": "START_REQUESTED",
"tags": [
 {
 "key": "string",
 "value": "string"
 }
],
"headNode": {
 "instanceId": "string",
 "instanceType": "string",
 "launchTime": "2019-08-24T14:15:22Z",
 "privateIpAddress": "string",
 "publicIpAddress": "string",
 "state": "pending"
},
"failures": [
 {
 "failureCode": "string",
 "failureReason": "string"
 }
]
"loginNodes": {
 "status": "string",
 "address": "string",
 "scheme": "string",
 "healthyNodes": integer,
 "unhealthyNodes": integer
}
}
```

## Isi respons

### ClusterName

Nama klaster.

Tipe: string

## cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

## cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## Konfigurasi Cluster

url

URL dari file konfigurasi cluster.

Tipe: string

## Status Cluster

Status cluster.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## computeFleetStatus

Status armada komputasi.

Tipe: string

Nilai yang valid: START\_REQUESTED | STARTING | RUNNING | PROTECTED |  
STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED



## CreationTime

Stempel waktu untuk saat cluster dibuat.

Jenis: datetime

## lastUpdatedTime

Stempel waktu kapan cluster terakhir diperbarui.

Jenis: datetime

## wilayah

TheWilayah AWSbaha cluster dibuat di.

Tipe: string

## tag

Daftar tag yang terkait dengan cluster.

kunci

Nama tag.

Tipe: string

tanda

Nilai tag.

Tipe: string

## versi

TheAWS ParallelClusterversi yang digunakan untuk membuat cluster.

Tipe: string

## kegagalan

Daftar kegagalan saat tumpukan cluster masukCREATE\_FAILEDstatus.

FailureCode

Kode kegagalan saat tumpukan cluster masukCREATE\_FAILEDstatus.

Tipe: string

FailureReason

Alasan kegagalan saat tumpukan cluster masuk `CREATE_FAILED` status.

Tipe: string

head\_node

Node kepala cluster.

instanceId

ID instans EC2.

Tipe: string

instanceType

Jenis instans EC2.

Tipe: string

LaunchTime

Waktu ketika instans EC2 diluncurkan.

Jenis: datetime

privateIpAddress

Alamat IP pribadi cluster.

Tipe: string

publicIpAddress

Alamat IP publik cluster.

Tipe: string

status

Status instance node kepala.

Tipe: string

Nilai yang valid: pending | running | shutting-down | terminated | stopping  
| stopped

## penjadwal

### metadata

Metadata penjadwal.

#### nama

Nama penjadwal.

Tipe: string

#### versi

Versi penjadwal.

Tipe: string

## LoginNodes

### status

Status node login.

Tipe: string

Nilai yang valid: PENDING | FAILED | ACTIVE

### menegur

Alamat node login.

Tipe: string

### skema

Skema node login.

Tipe: string

### skema

Jumlah node yang sehat.

Tipe: integer

## skema

Jumlah node yang tidak sehat.

Tipe: integer

## jenis

Jenis penjadwal.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ describe_cluster(cluster_name_3x)
```

#### 200 Respon

```
{
 'cloud_formation_stack_status': 'CREATE_COMPLETE',
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cluster_configuration': {
 'url': 'https://parallelcluster-....'
 },
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'CREATE_COMPLETE',
 'compute_fleet_status': 'RUNNING',
 'creation_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
 'head_node': {
 'instance_id': 'i-abcdef01234567890',
 'instance_type': 't2.micro',
 'launch_time': datetime.datetime(2022, 3, 28, 22, 21, 56, tzinfo=tzlocal()),
 'private_ip_address': '172.31.56.3',
 'public_ip_address': '107.23.100.164',
 'state': 'running'
 },
}
```

```
'last_updated_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [
 {
 'key': 'parallelcluster:version', 'value': '3.2.1'
 }
],
'version': '3.2.1'
}
```

## describeClusterInstances

Jelaskan contoh yang termasuk dalam cluster.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/clusters/{clusterName}/instances
{
 "nextToken": "string",
 "nodeType": "string",
 "queueName": "string",
 "region": "string"
}
```

### Isi permintaan

ClusterName

Nama klaster.

Tipe string

Wajib: Ya

### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

### NodeType

Filter instance berdasarkan tipe node.

Tipe: string

Nilai valid: HeadNode, ComputeNode, LoginNode

Wajib: Tidak

### Antrian

Filter instance dengan nama antrian.

Jenis: string

Wajib: Tidak

### wilayah

TheWilayah AWSbakwa cluster berada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "instances": [
 {
 "instanceId": "string",
 "instanceType": "string",
```

```
 "launchTime": "2019-08-24T14:15:22Z",
 "privateIpAddress": "string",
 "publicIpAddress": "string",
 "state": "pending",
 "nodeType": "HeadNode",
 "queueName": "string"
 }
]
}
```

## Isi respons

### Instans

Daftar instance cluster.

instanceId

ID instans EC2.

Tipe: string

instanceType

Jenis instans EC2.

Tipe: string

launchTime

Waktu ketika instans EC2 diluncurkan.

Jenis: datetime

nodeType

Tipe simpul.

Tipe: string

Nilai valid: HeadNode, ComputeNode, LoginNode

publicIpAddress

Alamat IP publik cluster.

Tipe: string

## Antrian

Nama antrian tempat instans EC2 mendukung node.

Tipe: string

## status

Status instance EC2 node.

Tipe: string

Nilai yang valid: pending | running | shutting-down | terminated | stopping  
| stopped

## NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ describe_cluster_instances(cluster_name_3x)
```

#### 200 Respon

```
{
 'instances': [
 {
 'instance_id': 'i-abcdef01234567890',
 'instance_type': 't2.micro',
 'launch_time': datetime.datetime(2022, 3, 30, 14, 2, 7, tzinfo=tzlocal()),
 'node_type': 'HeadNode',
 'private_ip_address': '192.0.2.5',
 'public_ip_address': '198.51.100.180',
 'state': 'running'
 }
]
}
```



```
}
]
}
```

## describeComputeFleet

Jelaskan status armada komputasi.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/clusters/{clusterName}/computeFleet
{
 "region": "string"
}
```

### Isi permintaan

#### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

#### wilayah

TheWilayah AWSbaha cluster berada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "status": "START_REQUESTED",
 "lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

### Isi respons

#### status

Tipe: string

Nilai yang valid: START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

#### lastStatusUpdatedWaktu

Stempel waktu yang mewakili waktu pembaruan status terakhir.

Jenis: datetime

### Contoh

#### Python

##### Permintaan

```
$ describe_compute_fleet(cluster_name_3x)
```

##### 200 Respon

```
{
 'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
 tzinfo=tzlocal()),
 'status': 'RUNNING'
}
```

## DescribeImage

Dapatkan informasi rinci tentang gambar yang ada.

## Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
GET /v3/images/custom/{imageId}
{
 "region": "string"
}
```

## Isi permintaan

### imageId

ID gambar.

Tipe string

Wajib: Ya

### wilayah

TheWilayah AWSbaha gambar itu dibuat di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "imageId": "string",
 "region": "string",
 "version": "string",
```

```
"imageBuildStatus": "BUILD_IN_PROGRESS",
"imageBuildLogsArn": "string",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackStatusReason": "string",
"cloudformationStackArn": "string",
"creationTime": "2019-08-24T14:15:22Z",
"cloudformationStackCreationTime": "2019-08-24T14:15:22Z",
"cloudformationStackTags": [
 {
 "key": "string",
 "value": "string"
 }
],
"imageConfiguration": {
 "url": "string"
},
"imagebuilderImageStatus": "PENDING",
"imagebuilderImageStatusReason": "string",
"ec2AmiInfo": {
 "amiId": "string",
 "tags": [
 {
 "key": "string",
 "value": "string"
 }
],
 "amiName": "string",
 "architecture": "string",
 "state": "PENDING",
 "description": "string"
}
}
```

## Isi respons

### imageId

ID gambar untuk mengambil informasi rinci untuk.

Tipe: string

### imageBuildStatus

Status pembuatan gambar.

Tipe: string

Nilai yang valid: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

ImageConfiguration

url

URL dari file konfigurasi gambar.

Tipe: string

wilayah

TheWilayah AWSbahwa gambar dibuat di.

Tipe: string

versi

TheAWS ParallelClusterversi yang digunakan untuk membangun gambar.

Tipe: string

cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

cloudformationStackCreationWaktu

Stempel waktu untuk saat CloudFormation tumpukan telah dibuat.

Jenis: datetime

cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |

UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

cloudformationStackStatusAlasan

Alasan untuk CloudFormation status tumpukan.

Tipe: string

cloudformationStackTags

Daftar tag untuk CloudFormation tumpukan.

kunci

Nama tag.

Tipe: string

nilai

Nilai tag.

Tipe: string

CreationTime

Stempel waktu untuk saat gambar dibuat.

Jenis: datetime

ec2AmiInfo

AmiID

ID AMI EC2.

Tipe: string

Aminame

Nama EC2 AMI.

Tipe: string

arsitektur

Arsitektur EC2 AMI.

Tipe: string

status

Keadaan EC2 AMI.

Tipe: string

Nilai yang valid: PENDING | AVAILABLE | INVALID | DEREGISTERED | TRANSIENT  
| FAILED | ERROR

tag

Daftar Tag AMI EC2.

kunci

Nama tag.

Tipe: string

nilai

Nilai tag.

Tipe: string

imagebuilderImageStatus

The ImageBuilder status.

Tipe: string

Nilai yang valid: PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING |  
INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

imagebuilderImageStatusAlasan

Alasan untuk ImageBuilder Status gambar.

Tipe: string

imageBuildLogsArn

Amazon Resource Name (ARN) dari log untuk proses pembuatan gambar.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ describe_image(custom-image-id)
```

#### 200 Respon

```
{
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-image-id/6accc570-b080-11ec-845e-0e2dc6386985',
 'cloudformation_stack_creation_time': datetime.datetime(2022, 3, 30, 23, 23, 33,
731000, tzinfo=tzlocal()),
 'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
 'cloudformation_stack_tags': [
 {
 'key': 'parallelcluster:version', 'value': '3.2.1'
 },
 {
 'key': 'parallelcluster:image_name',
 'value': 'custom-image-id'
 },
 {
 'key': 'parallelcluster:custom-image-id',
 'value': 'custom-image-id'
 },
 {
 'key': 'parallelcluster:s3_bucket',
 'value': 'parallelcluster-abcdef01234567890-v1-do-not-delete'
 },
 {
 'key': 'parallelcluster:s3_image_dir',
 'value': 'parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0'
 },
 {
 'key': 'parallelcluster:build_log',
 'value': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-image-id'
 },
 {
 'key': 'parallelcluster:build_config',
```



```
 'value': 's3://parallelcluster-abcdef01234567890-v1-do-not-delete/parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-config.yaml'
 }
],
'image_build_logs_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/ParallelClusterImage-alinux2-image',
'image_build_status': 'BUILD_IN_PROGRESS',
'image_configuration': {
 'url': 'https://parallelcluster-abcdef01234567890-v1-do-not-delete.s3.amazonaws.com/parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-config.yaml?...'
},
'image_id': 'custom-image-id',
'imagebuilder_image_status': 'PENDING',
'region': 'us-east-1',
'version': '3.2.1'
}
```

## getClusterLogAcara

Ambil peristiwa yang terkait dengan aliran log.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/clusters/{clusterName}/logstreams/{logStreamName}
{
 "endTime": datetime,
 "limit": float,
 "nextToken": "string",
 "region": "string",
 "startFromHead": boolean,
```

```
"startTime": datetime
}
```

## Isi permintaan

### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

### logStreamName

Nama pengaliran log.

Tipe string

Wajib: Ya

### endTime

Akhir rentang waktu, dinyatakan dalam format ISO 8601. Acara dengan stempel waktu sama dengan atau lebih lambat dari waktu ini tidak termasuk.

Jenis: datetime

Format: 2021-01-01T20:00:00Z

Wajib: Tidak

### batasan

Jumlah maksimum peristiwa log yang dikembalikan. Jika Anda tidak menentukan nilai, maksimum adalah sebanyak peristiwa log yang dapat muat dalam ukuran respons 1 MB, hingga 10.000 peristiwa log.

Tipe: float

Diperlukan: Tidak

### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

wilayah

TheWilayah AWSbaha cluster berada di.

Jenis: string

Wajib: Tidak

startFromHead

Jika disetel ke `true`, peristiwa log paling awal dikembalikan terlebih dahulu. Jika nilainya `false`, peristiwa log terbaru dikembalikan terlebih dahulu. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

startTime

Awal rentang waktu, dinyatakan dalam format ISO 8601. Acara dengan stempel waktu yang sama dengan waktu ini atau lebih lambat dari waktu ini disertakan.

Jenis: datetime

Format: `2021-01-01T20:00:00Z`

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "prevToken": "string",
 "events": [
 {
 "timestamp": "2019-08-24T14:15:22Z",
 "message": "string"
 }
]
}
```

## Isi respons

### peristiwa

Daftar acara yang disaring.

pesan

Pesan acara.

Tipe: string

timestamp

Stempel waktu acara.

Jenis: datetime

### NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

### PrevToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

### Python

Permintaan

```
$ get_cluster_log_events(cluster_name_3x, log_stream_name=ip-192-0-2-26.i-
abcdef01234567890.cfn-init)
```

200 Respon

```
"events": [
 {
 "message": "2022-09-22 16:40:15,127 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.us-east-1.amazonaws.com",
```

```
 "timestamp": "2022-09-22T16:40:15.127Z"
 },
 {
 "message": "2022-09-22 16:40:15,127 [DEBUG] Describing resource
HeadNodeLaunchTemplate in stack cluster_name_3x",
 "timestamp": "2022-09-22T16:40:15.127Z"
 },
 ...
]
```

## getClusterStackAcara

Ambil peristiwa yang terkait dengan tumpukan untuk sebuah cluster.

### Note

Mulai dari versi 3.6.0, AWS ParallelCluster menggunakan tumpukan bersarang untuk membuat sumber daya yang terkait dengan antrian dan sumber daya komputasi. The `GetClusterStackEventsAPI` dan `pccluster get-cluster-stack-events` perintah hanya mengembalikan peristiwa tumpukan utama cluster. Anda dapat melihat peristiwa tumpukan cluster, termasuk yang terkait dengan antrian dan sumber daya komputasi, di CloudFormation konsol.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/clusters/{clusterName}/stackevents
{
 "nextToken": "string",
 "region": "string"
```

```
}
```

## Isi permintaan

### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

### wilayah

TheWilayah AWSbaha cluster berada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "events": [
 {
 "stackId": "string",
 "eventId": "string",
 "stackName": "string",
 "logicalResourceId": "string",
 "physicalResourceId": "string",
 "resourceType": "string",
 "timestamp": "2019-08-24T14:15:22Z",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "string",
```

```
 "resourceProperties": "string",
 "clientRequestToken": "string"
 }
]
}
```

## Isi respons

### peristiwa

Daftar acara yang disaring.

#### clientRequestToken

Token diteruskan ke tindakan yang menghasilkan acara ini.

Tipe: string

#### EventID

ID unik dari acara ini.

Tipe: string

#### logicalResourceId

Nama logis dari sumber daya yang ditentukan dalam template.

Tipe: string

#### physicalResourceId

Nama atau pengidentifikasi unik yang terkait dengan contoh fisik sumber daya.

Tipe: string

#### ResourceProperties

Sebuah BLOB dari properti yang digunakan untuk membuat sumber daya.

Tipe: string

#### Sumber DayaStatus

Status sumber daya.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED | UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED | IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS | IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

resourceStatusReason

Pesan sukses atau gagal yang terkait dengan sumber daya.

Tipe: string

resourceType

Tipe sumber daya.

Tipe: string

StackID

Nama ID unik dari instance tumpukan.

Tipe: string

Nama tumpukan

Nama yang terkait dengan tumpukan.

Tipe: string

timestamp

Waktu ketika status diperbarui.

Jenis: datetime

NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

Python

Permintaan



```
$ get_cluster_stack_events(cluster_name_3x)
```

200 Respon

```
{
 'events': [
 {
 'event_id': '590b3820-b081-11ec-985e-0a7af5751497',
 'logical_resource_id': 'cluster_name_3x',
 'physical_resource_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
 'resource_status': 'CREATE_COMPLETE',
 'resource_type': 'AWS::CloudFormation::Stack',
 'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
 'stack_name': 'cluster_name_3x',
 'timestamp': datetime.datetime(2022, 3, 30, 23, 30, 13, 268000,
tzinfo=tzlocal())
 },
 ...
]
}
```

## getImageLogAcara

Ambil peristiwa yang terkait dengan pembuatan gambar.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
GET /v3/images/custom/{imageId}/logstreams/{logStreamName}
{
```

```
"endTime": datetime,
"limit": float,
"nextToken": "string",
"region": "string",
"startFromHead": boolean,
"startTime": datetime
}
```

## Isi permintaan

### imageId

ID gambar.

Tipe string

Wajib: Ya

### logStreamName

Nama logstream.

Tipe string

Wajib: Ya

### endTime

Akhir rentang waktu, dinyatakan dalam format ISO 8601. Peristiwa dengan stempel waktu sama dengan atau lebih lambat dari waktu ini tidak disertakan.

Jenis: datetime

Format: 2021-01-01T20:00:00Z

Wajib: Tidak

### batasan

Jumlah maksimum peristiwa log yang dikembalikan. Jika Anda tidak menentukan nilai, maksimum adalah sebanyak peristiwa log yang dapat muat dalam ukuran respons 1 MB, hingga 10.000 peristiwa log.

Tipe: float

Diperlukan: Tidak

## NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

## wilayah

TheWilayah AWSbahwa gambar di.

Jenis: string

Wajib: Tidak

## startFromHead

Jika disetel ke `true`, kembalikan peristiwa log paling awal terlebih dahulu. Jika disetel ke `false`, kembalikan peristiwa log terbaru terlebih dahulu. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

## startTime

Awal rentang waktu, dinyatakan dalam format ISO 8601. Acara dengan stempel waktu yang sama dengan waktu ini atau lebih lambat dari waktu ini disertakan.

Jenis: datetime

Format: `2021-01-01T20:00:00Z`

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "prevToken": "string",
 "events": [
 {
```

```
 "timestamp": "2019-08-24T14:15:22Z",
 "message": "string"
 }
]
}
```

## Isi respons

### peristiwa

Daftar acara yang disaring.

pesan

Pesan acara.

Tipe: string

timestamp

Stempel waktu acara.

Jenis: datetime

### NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

### PrevToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

### Python

Permintaan

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

200 Respon

```
"events": [
 {
 "message": "ExecuteBash: STARTED EXECUTION",
 "timestamp": "2022-04-05T15:51:20.228Z"
 },
 {
 "message": "ExecuteBash: Created temporary directory: /tmp/1234567890abcdef0",
 "timestamp": "2022-04-05T15:51:20.228Z"
 },
 ...
]
```

## getImageStackAcara

Ambil peristiwa yang terkait dengan tumpukan untuk pembuatan gambar.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/images/custom/{imageId}/stackevents
{
 "nextToken": "string",
 "region": "string"
}
```

### Isi permintaan

imageId

ID gambar.

Tipe string

Wajib: Ya

NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

wilayah

TheWilayah AWSbaha gambar itu ada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "events": [
 {
 "stackId": "string",
 "eventId": "string",
 "stackName": "string",
 "logicalResourceId": "string",
 "physicalResourceId": "string",
 "resourceType": "string",
 "timestamp": "2019-08-24T14:15:22Z",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "string",
 "resourceProperties": "string",
 "clientRequestToken": "string"
 }
]
}
```

## Isi respons

peristiwa

Daftar acara yang disaring.

### clientRequestToken

Token diteruskan ke tindakan yang menghasilkan acara ini.

Tipe: string

### EventID

ID unik dari acara ini.

Tipe: string

### logicalResourceId

Nama logis dari sumber daya yang ditentukan dalam template.

Tipe: string

### physicalResourceId

Nama atau pengidentifikasi unik yang terkait dengan contoh fisik sumber daya.

Tipe: string

### ResourceProperties

Sebuah BLOB dari properti yang digunakan untuk membuat sumber daya.

Tipe: string

### Sumber DayaStatus

Status sumber daya.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED | UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED | IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS | IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

### resourceStatusReason

Pesan sukses atau gagal yang terkait dengan sumber daya.

Tipe: string

## resourceType

Tipe sumber daya.

Tipe: string

## StackID

Nama ID unik dari instance tumpukan.

Tipe: string

## Nama tumpukan

Nama yang terkait dengan tumpukan.

Tipe: string

## timestamp

Waktu ketika status diperbarui.

Jenis: datetime

## NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ get_image_stack_events(image_id)
```

#### 200 Respon

```
{
 'events': [
 {
 'event_id': 'ParallelClusterImage-
CREATE_IN_PROGRESS-2022-03-30T23:26:33.499Z',
 'logical_resource_id': 'ParallelClusterImage',
```



```

 'physical_resource_id': 'arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-alinux2-image/3.2.1/1',
 'resource_properties': {
 "InfrastructureConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
 "ImageRecipeArn": "arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-alinux2-image/3.2.1",
 "DistributionConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
 "EnhancedImageMetadataEnabled": "false",
 "Tags": {
 "parallelcluster:image_name": "alinux2-
image", "parallelcluster:image_id": "alinux2-image"
 }
 },
 'resource_status': 'CREATE_IN_PROGRESS',
 'resource_status_reason': 'Resource creation Initiated',
 'resource_type': 'AWS::ImageBuilder::Image',
 'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/alinux2-
image/6accc570-b080-11ec-845e-0e2dc6386985',
 'stack_name': 'alinux2-image',
 'timestamp': datetime.datetime(2022, 3, 30, 23, 26, 33, 499000,
tzinfo=tzlocal())
},
...
]
}

```

## ListClusters

Ambil daftar cluster yang ada.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
GET /v3/clusters
{
 "clusterStatus": "string",
 "nextToken": "string",
 "region": "string"
}
```

## Isi permintaan

### Status Cluster

Filter berdasarkan status cluster. Defaultnya adalah semua cluster.

Tipe: string

Nilai valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_FAILED

Wajib: tidak

### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

### wilayah

TheWilayah AWSdari cluster.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
```

```
"nextToken": "string",
"clusters": [
 {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
 }
]
}
```

## Isi respons

### kluster

#### cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

#### cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ClusterName

Nama klaster.

Tipe: string

## Status Cluster

Status cluster.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## penjadwal

### metadata

Metadata penjadwal.

#### nama

Nama penjadwal.

Tipe: string

#### versi

Versi penjadwal.

Tipe: string

## jenis

Jenis penjadwal.

Tipe: string

## wilayah

TheWilayah AWSbaha cluster dibuat di.

Tipe: string

## versi

TheAWS ParallelClusterversi yang digunakan untuk membuat cluster.

Tipe: string

NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

Contoh

Python

Permintaan

```
$ list_clusters()
```

200 Respon

```
{
 'clusters':
 [
 {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cloudformation_stack_status': 'CREATE_COMPLETE',
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'CREATE_COMPLETE',
 'region': 'us-east-1',
 'version': '3.2.1'
 },
 ...
]
}
```

## listClusterLogAliran

Ambil daftar aliran log yang terkait dengan cluster.

Topik

- [Sintaksis Permintaan](#)

- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
GET /v3/clusters/{clusterName}/logstreams
{
 "filters": ["string"],
 "nextToken": "string",
 "region": "string"
}
```

### Isi permintaan

#### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

#### saringan

Saring aliran log.

Filter yang diterima adalah:

- private-dns-name: Bentuk singkat dari nama DNS pribadi instance (misalnya ip-10-0-0-101).
- tipe-simpul: Nilai valid:HeadNode.

Jenis: Array string unik

Format: Name=a,Values=1 Name=b,Values=2,3

Diperlukan: Tidak

#### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

wilayah

TheWilayah AWSbaha cluster berada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "logStreams": [
 {
 "logStreamName": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "firstEventTimestamp": "2019-08-24T14:15:22Z",
 "lastEventTimestamp": "2019-08-24T14:15:22Z",
 "lastIngestionTime": "2019-08-24T14:15:22Z",
 "uploadSequenceToken": "string",
 "logStreamArn": "string"
 }
]
}
```

## Isi respons

### LogStreams

Daftar aliran log.

CreationTime

Waktu ketika aliran itu dibuat.

Jenis: datetime

firstEventTimestamp

Waktu acara pertama aliran.

Jenis: datetime

lastEventTimestamp

Waktu peristiwa terakhir aliran. The lastEventTime pembaruan nilai berdasarkan konsistensi akhirnya. Ini biasanya diperbarui dalam waktu kurang dari satu jam dari konsumsi, tetapi dalam situasi yang jarang mungkin memakan waktu lebih lama.

Jenis: datetime

lastIngestionTime

Waktu konsumsi terakhir.

Jenis: datetime

logStreamArn

Nama Sumber Daya Amazon (ARN) dari aliran log.

Tipe: string

logStreamName

Nama aliran log.

Tipe: string

uploadSequenceToken

Token urutan.

Tipe: string

NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

Python

Permintaan

```
$ list_cluster_log_streams(cluster_name_3x)
```



## 200 Respon

```
{
 'log_streams': [
 {
 'creation_time': datetime.datetime(2022, 3, 30, 14, 7, 34, 354000,
tzinfo=tzlocal()),
 'first_event_timestamp': datetime.datetime(2022, 3, 30, 14, 6, 41, 444000,
tzinfo=tzlocal()),
 'last_event_timestamp': datetime.datetime(2022, 3, 30, 14, 25, 55, 462000,
tzinfo=tzlocal()),
 'last_ingestion_time': datetime.datetime(2022, 3, 30, 14, 49, 50, 62000,
tzinfo=tzlocal()),
 'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
parallelcluster/cluster_name_3x:log-stream:ip-192-0-2-26.i-abcdef01234567890.cfn-
init',
 'log_stream_name': 'ip-192-0-2-26.i-abcdef01234567890.cfn-init',
 ...
 'upload_sequence_token': '####'
 },
 ...
]
}
```

## listImageLogAliran

Ambil daftar aliran log yang terkait dengan gambar.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
GET /v3/images/custom/{imageId}/logstreams
{
```

```
"nextToken": "string",
"region": "string"
}
```

## Isi permintaan

### imageId

ID gambar.

Tipe string

Wajib: Ya

### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

### wilayah

TheWilayah AWS bahwa gambar itu ada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "logStreams": [
 {
 "logStreamName": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "firstEventTimestamp": "2019-08-24T14:15:22Z",
 "lastEventTimestamp": "2019-08-24T14:15:22Z",
 "lastIngestionTime": "2019-08-24T14:15:22Z",
 "uploadSequenceToken": "string",
 "logStreamArn": "string"
 }
]
}
```

```
}
]
}
```

## Isi respons

### LogStreams

Daftar aliran log.

CreationTime

Waktu ketika aliran itu dibuat.

Jenis: datetime

firstEventTimestamp

Waktu acara pertama di sungai.

Jenis: datetime

lastEventTimestamp

Waktu peristiwa terakhir aliran. The lastEventTime pembaruan nilai berdasarkan konsistensi akhirnya. Ini biasanya diperbarui dalam waktu kurang dari satu jam dari konsumsi, tetapi dalam situasi yang jarang mungkin memakan waktu lebih lama.

Jenis: datetime

lastIngestionTime

Waktu konsumsi terakhir.

Jenis: datetime

logStreamArn

Nama Sumber Daya Amazon (ARN) dari aliran log.

Tipe: string

logStreamName

Nama pengaliran log.

Tipe: string

uploadSequenceToken

Token urutan.

Tipe: string

next\_token

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

### Python

Permintaan

```
$ list_image_log_streams(custom-image-id)
```

200 Respon

```
{
 'log_streams': [
 {
 'creation_time': datetime.datetime(2022, 3, 29, 20, 29, 24, 875000,
tzinfo=tzlocal()),
 'first_event_timestamp': datetime.datetime(2022, 3, 29, 20, 29, 24, 775000,
tzinfo=tzlocal()),
 'last_event_timestamp': datetime.datetime(2022, 3, 29, 20, 38, 23, 944000,
tzinfo=tzlocal()),
 'last_ingestion_time': datetime.datetime(2022, 3, 29, 20, 51, 56, 26000,
tzinfo=tzlocal()),
 'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image:log-stream:3.2.1/1',
 'log_stream_name': '3.2.1/1',
 'upload_sequence_token': '####'
 },
 ...
]
}
```

# ListImages

Ambil daftar gambar kustom yang ada.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
GET /images/custom
{
 "imageStatus": "string",
 "nextToken": "string",
 "region": "string"
}
```

## Isi permintaan

### ImageStatus

Filter gambar berdasarkan status yang disediakan.

Tipe: string

Nilai yang valid: AVAILABLE | PENDING | FAILED

Wajib: Ya

### NextToken

Token yang digunakan untuk permintaan paginasi.

Jenis: string

Wajib: Tidak

## wilayah

TheWilayah AWSgambar-gambar itu ada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "nextToken": "string",
 "images": [
 {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS"
 }
]
}
```

## Isi respons

### citra

Daftar gambar.

cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

#### ec2AmiInfo

ami\_id

ID AMI EC2.

Tipe: string

#### imageBuildStatus

Status pembuatan gambar.

Nilai yang valid: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

Tipe: string

#### imageId

ID gambar.

Tipe: string

#### wilayah

TheWilayah AWSbahwa gambar dibuat di.

Tipe: string

#### versi

TheAWS ParallelClusterversi yang digunakan untuk membangun gambar.

Tipe: string

#### NextToken

Token yang digunakan untuk permintaan paginasi.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ list_images("AVAILABLE")
```

#### 200 Respon

```
{
 'images': [
 {
 'ec2_ami_info': {
 'ami_id': 'ami-abcdef01234567890'
 },
 'image_build_status': 'BUILD_COMPLETE',
 'image_id': 'custom-image',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
]
}
```

## listOfficialImages

Ambil daftar AWS ParallelCluster gambar resmi.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

### Sintaksis Permintaan

```
GET /v3/images/official
```



```
{
 "architecture": "string",
 "os": "string",
 "region": "string"
}
```

## Isi permintaan

### arsitektur

Filter berdasarkan arsitektur. Defaultnya adalah tidak ada penyaringan.

Tipe: string

Nilai valid: x86\_64 | arm64

Wajib: Tidak

### os

Filter berdasarkan distribusi OS. Defaultnya adalah tidak ada penyaringan.

Tipe: string

Nilai valid: alinux2 | centos7 | ubuntu2204 | ubuntu2004 | rhel8

Wajib: Tidak

### wilayah

TheWilayah AWSbaha gambar resmi tercantum di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "images": [
 {
 "architecture": "string",
 "amiId": "string",
 "name": "string",
 }
]
}
```

```
 "os": "string",
 "version": "string"
 }
]
```

## Isi respons

### citra

#### AmiID

ID AMI.

Tipe: string

#### arsitektur

Arsitektur AMI.

Tipe: string

#### nama

Nama AMI.

Tipe: string

#### os

Sistem operasi AMI.

Tipe: string

#### versi

TheAWS ParallelClusterversi.

Tipe: string

## Contoh

### Python

#### Permintaan

```
$ list_official_images()
```

200 Respon

```
{
 'images': [
 {
 'ami_id': 'ami-015cfef4e0d6306b2',
 'architecture': 'x86_64',
 'name': 'aws-parallelcluster-3.2.1-ubuntu-2004-lts-hvm-x86_64-202202261505 '
 '2022-02-26T15-08-34.759Z',
 'os': 'ubuntu2004',
 'version': '3.2.1'
 },
 ...
]
}
```

## UpdateCluster

Perbarui cluster.

Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
PUT /v3/clusters/{clusterName}
{
 "clusterConfiguration": "string",
 "dryrun": boolean,
 "forceUpdate": boolean,
 "region": "string",
 "suppressValidators": "string",
```

```
"validationFailureLevel": "string"
}
```

## Isi permintaan

### Konfigurasi Cluster

Konfigurasi cluster sebagai dokumen YAMAL.

Wajib: Ya

#### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

#### dryrun

Jika disetel ke `true`, hanya melakukan validasi permintaan tanpa membuat sumber daya apa pun. Gunakan parameter ini untuk memvalidasi konfigurasi cluster dan memperbarui persyaratan. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

#### ForceUpdate

Jika disetel ke `true`, abaikan kesalahan validasi pembaruan dan paksa pembaruan. Defaultnya adalah `false`.

Tipe: boolean

Wajib: Tidak

#### wilayah

TheWilayah AWS bahwa cluster berada di.

Jenis: string

Wajib: Tidak

## SuppressValidator

Mengidentifikasi satu atau lebih validator konfigurasi untuk menekan.

Tipe: string

Format: (ALL | type:[A-Za-z0-9]+)

Wajib: Tidak

Contoh nilai yang valid:currentValue,requestedValue,message

## validationFailureLevel

Tingkat validasi minimum menyebabkan pembaruan gagal.

Tipe: string

Nilai valid: INFO | WARNING | ERROR

Wajib: Tidak

## Sintaksis Respons

```
{
 "cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "clusterStatus": "UPDATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
 },
 "validationMessages": [
 {
 "id": "string",
```

```
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
],
"changeSet": [
 {
 "parameter": "string",
 "currentValue": "string",
 "requestedValue": "string"
 }
]
}
```

## Isi respons

### ChangeSet

Perubahan ditetapkan untuk pembaruan cluster.

#### CurrentValue

Nilai parameter saat ini yang akan diperbarui.

Tipe: string

#### parameter

Parameter yang akan diperbarui.

Tipe: string

#### RequestedValue

Nilai yang diminta untuk parameter yang akan diperbarui.

Tipe: string

### klaster

#### cloudformationStackArn

Nama Sumber Daya Amazon (ARN) utama CloudFormation tumpukan.

Tipe: string

## cloudformationStackStatus

The CloudFormation status tumpukan.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ClusterName

Nama cluster.

Tipe: string

## Status Cluster

Status cluster.

Tipe: string

Nilai yang valid: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## wilayah

TheWilayah AWSbaha cluster dibuat di.

Tipe: string

## penjadwal

### metadata

Metadata penjadwal.

### nama

Nama penjadwal.

Tipe: string

versi

Versi penjadwal.

Tipe: string

jenis

Jenis penjadwal.

Tipe: string

versi

AWS ParallelClusterversi yang digunakan untuk membuat cluster.

Tipe: string

ValidationMessages

Daftar pesan dengan tingkat validasi lebih rendah darivalidationFailureLevel. Daftar pesan dikumpulkan selama validasi konfigurasi.

id

ID validator.

Tipe: string

level

Tingkat validasi.

Tipe: string

Nilai yang valid: INFO | WARNING | ERROR

pesan

Pesan validasi.

Tipe: string

jenis

Jenis validator.



Tipe: string

## Contoh

### Python

#### Permintaan

```
$ update_cluster(cluster_name_3x, path/config-file.yaml)
```

#### 200 Respon

```
{
 'change_set': [
 {
 'current_value': '10',
 'parameter':
'Scheduling.SlurmQueues[queue1].ComputeResources[t2micro].MaxCount',
 'requested_value': '15'
 }
],
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/test-api-cluster/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
 'cloudformation_stack_status': 'UPDATE_IN_PROGRESS',
 'cluster_name': 'cluster-3x',
 'cluster_status': 'UPDATE_IN_PROGRESS',
 'region': 'us-east-1',
 'scheduler': {
 'type': 'slurm'
 },
 'version': '3.2.1'
 }
}
```

## updateComputeFleet

Perbarui status armada komputasi cluster.

### Topik

- [Sintaksis Permintaan](#)
- [Isi permintaan](#)
- [Sintaksis Respons](#)
- [Isi respons](#)
- [Contoh](#)

## Sintaksis Permintaan

```
PATCH /v3/clusters/{clusterName}/computeFleet
{
 "status": "string",
 "region": "string"
}
```

### Isi permintaan

#### ClusterName

Nama klaster.

Tipe string

Wajib: Ya

#### status

Status armada komputasi.

Tipe: string

Nilai yang valid: START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

Wajib: Ya

#### wilayah

TheWilayah AWS bahwa cluster berada di.

Jenis: string

Wajib: Tidak

## Sintaksis Respons

```
{
 "status": "START_REQUESTED",
 "lastStatusUpdatedTime": "2019-08-24T14:15:22Z"
}
```

## Isi respons

### status

Status armada komputasi.

Tipe: string

Nilai yang valid: START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdatedWaktu

Stempel waktu yang mewakili waktu pembaruan status terakhir.

Jenis: datetime

## Contoh

### Python

#### Permintaan

```
$ update_compute_fleet(cluster_name_3x, "START_REQUESTED")
```

#### 200 Respon

```
{
 'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
 tzinfo=tzlocal()),
 'status': 'START_REQUESTED'
```

}

## AWS ParallelClusterAPI pustaka Python

Dimulai denganAWS ParallelClusterversi 3.5.0, Anda dapat mengaksesAWS ParallelClusterdenganAWS ParallelClusterPustaka Python. Anda dapat mengaksesAWS ParallelClusterperpustakaan di AndapcClusterlingkungan atau dari dalamAWS Lambdaruntime. Pelajari cara mengaksesAWS ParallelClusterAPI dengan menggunakanAWS ParallelClusterPustaka Python. TheAWS ParallelClusterPustaka Python menawarkan fungsionalitas yang sama denganAWS ParallelClusterAPI memberikan.

TheAWS ParallelClusterOperasi dan parameter pustaka Python mencerminkan parameter API saat dikonversi kesnake\_casetanpa huruf kapital.

### Topik

- [AWS ParallelClusterOtorisasi perpustakaan Python](#)
- [InstalAWS ParallelClusterPustaka Python](#)
- [Operasi API cluster](#)
- [Hitung operasi API armada](#)
- [Operasi cluster dan stack log](#)
- [Operasi API gambar](#)
- [Operasi log gambar dan tumpukan](#)
- [Contoh](#)
- [AWS LambdauntukAWS ParallelClusterPustaka Python](#)

## AWS ParallelClusterOtorisasi perpustakaan Python

Tentukan kredensyal dengan menggunakan salah satu cara standar yang valid untuk boto3. Untuk informasi lebih lanjut, lihat[dokumentasi boto3](#).

## InstalAWS ParallelClusterPustaka Python

1. InstalpcClusterCLI versi 3.5.0 atau yang lebih baru dengan mengikuti instruksi yang diberikan di[Menyiapkan AWS ParallelCluster](#).

2. Impor `pcluster` modul dan mulai menggunakan perpustakaan, seperti yang ditunjukkan pada contoh berikut:

```
import pcluster.lib as pc
pc.create_cluster(cluster_name="mycluster", cluster_configuration="config.yaml")
```

## Operasi API cluster

Topik

- [list\\_clusters](#)
- [create\\_cluster](#)
- [delete\\_cluster](#)
- [describe\\_cluster](#)
- [update\\_cluster](#)

### **list\_clusters**

```
list_clusters(region, next_token, cluster_status)
```

Ambil daftar cluster yang ada.

Parameter:

#### **region**

Daftar cluster yang digunakan untuk diberikan Wilayah AWS.

#### **next\_token**

Token yang digunakan untuk permintaan paginasi.

#### **cluster\_status**

Filter berdasarkan status cluster. Defaultnya adalah daftar semua cluster.

Nilai yang

valid: `CREATE_IN_PROGRESS|CREATE_FAILED|CREATE_COMPLETE|DELETE_IN_PROGRESS|DELETE_F`

## create\_cluster

```
create_cluster(cluster_name, cluster_configuration, region, suppress_validators,
validation_failure_level, dry_run, rollback_on_failure, wait)
```

Buat cluster di Wilayah tertentu.

Parameter:

### **cluster\_name**(diperlukan)

Nama cluster.

### **cluster\_configuration**(diperlukan)

Konfigurasi cluster sebagai tipe data Python.

### **region**

ClusterWilayah AWS.

### **suppress\_validators**

Mengidentifikasi satu atau beberapa validator konfigurasi cluster untuk menekan.

Format: (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

Tingkat validasi minimum yang menyebabkan pembuatan cluster gagal. Defaultnya adalah ERROR.

Nilai yang valid:INFO|WARNING|ERROR.

### **dry\_run**

Melakukan validasi permintaan tanpa membuat sumber daya apa pun. Anda dapat menggunakan ini untuk memvalidasi konfigurasi cluster. Defaultnya adalah `False`.

### **rollback\_on\_failure**

Jika disetel ke `True`, AWS ParallelCluster secara otomatis memulai rollback tumpukan cluster pada kegagalan. Defaultnya adalah `True`.

### **wait**

Jika disetel ke `True`, AWS ParallelCluster menunggu operasi selesai. Defaultnya adalah `False`.

## delete\_cluster

```
delete_cluster(cluster_name, region, wait)
```

Hapus cluster di Wilayah tertentu.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**region**

ClusterWilayah AWS.

**wait**

Jika disetel keTrue, menunggu operasi selesai. Defaultnya adalah False.

## describe\_cluster

```
describe_cluster(cluster_name, region)
```

Dapatkan informasi rinci tentang cluster yang ada.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**region**

ClusterWilayah AWS.

## update\_cluster

```
update_cluster(cluster_name, cluster_configuration, suppress_validators,
validation_failure_level, region, force_update, dry_run, wait)
```

Perbarui cluster di Wilayah tertentu.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**cluster\_configuration**(diperlukan)

Konfigurasi cluster sebagai tipe data Python.

**suppress\_validators**

Mengidentifikasi satu atau beberapa validator konfigurasi cluster untuk menekan.

Format: (ALL | type:[A-Za-z0-9]+)

**validation\_failure\_level**

Tingkat validasi minimum yang menyebabkan pembaruan cluster gagal. Defaultnya adalah ERROR.

Nilai yang valid: INFO | WARNING | ERROR

**region**

ClusterWilayah AWS.

**dry\_run**

Melakukan validasi permintaan tanpa membuat atau memperbarui sumber daya apa pun. Anda dapat menggunakan ini untuk memvalidasi konfigurasi cluster. Defaultnya adalah False.

**force\_update**

Jika disetel ke True, memaksa pembaruan dengan mengabaikan kesalahan validasi pembaruan. Defaultnya adalah False.

**wait**

Jika disetel ke True, menunggu operasi selesai. Defaultnya adalah False.

## Hitung operasi API armada

Topik

- [describe\\_compute\\_fleet](#)



- [update\\_compute\\_fleet](#)
- [delete\\_cluster\\_instances](#)
- [describe\\_cluster\\_instances](#)

## describe\_compute\_fleet

```
describe_compute_fleet(cluster_name, region)
```

Jelaskan status armada komputasi cluster untuk cluster tertentu.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**region**

Menjelaskan status armada komputasi untuk kluster yang dikerahkan ke tertentuWilayah AWS.

## update\_compute\_fleet

```
update_compute_fleet(cluster_name, status, region)
```

Perbarui status armada komputasi cluster.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**status**(diperlukan)

Status untuk memperbarui ke.

Nilai yang valid: START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

**region**

ClusterWilayah AWS.

## delete\_cluster\_instances

```
delete_cluster_instances(cluster_name, region, force)
```

Hapus cluster di Wilayah tertentu.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**region**

ClusterWilayah AWS.

**force**

Jika disetel keTrue, memaksa penghapusan ketika cluster dengan yang diberikanc`cluster_name`tidak ditemukan. Defaultnya adalah False.

## describe\_cluster\_instances

```
describe_cluster_instances(cluster_name, region, next_token, node_type, queue_name)
```

Jelaskan contoh cluster.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**region**

ClusterWilayah AWS.

**next\_token**

Token yang digunakan untuk permintaan paginasi.

**node\_type**

Memfilter instance dengannode\_type.

Nilai yang valid: HeadNode | ComputeNode

## queue\_name

Memfilter instance dengan nama antrian.

## Operasi cluster dan stack log

Topik

- [list\\_cluster\\_log\\_streams](#)
- [get\\_cluster\\_log\\_events](#)
- [get\\_cluster\\_stack\\_events](#)

## list\_cluster\_log\_streams

```
list_cluster_log_streams(cluster_name, region, filters, next_token)
```

Daftar aliran log untuk cluster tertentu.

Parameter:

**cluster\_name**(diperlukan)

Nama cluster.

**region**

ClusterWilayah AWS.

**filters**

Memfilter aliran log cluster.

Format: 'Name=a,Values=1 Name=b,Values=2,3'

Filter yang diterima:

code-dns-name

Bentuk singkat dari nama DNS pribadi dari instance; misalnya,ip-10-0-0-101.

## tipe simpul

Tipe simpul.

Nilai yang valid: HeadNode

## next\_token

Token yang digunakan untuk permintaan paginasi.

## get\_cluster\_log\_events

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,
start_from_head, limit, start_time, end_time)
```

Dapatkan peristiwa log untuk kluster dan aliran log tertentu.

Parameter:

### **cluster\_name**(diperlukan)

Nama cluster.

### **log\_stream\_name**(diperlukan)

Nama log stream.

### **region**

ClusterWilayah AWS.

### **next\_token**

Token yang digunakan untuk permintaan paginasi.

### **start\_from\_head**

Jika disetel ke `True`, AWS ParallelCluster mengembalikan peristiwa log paling awal terlebih dahulu. Jika disetel ke `False`, ia mengembalikan peristiwa log terbaru terlebih dahulu. Defaultnya adalah `False`.

### **limit**

Jumlah maksimum peristiwa log yang dikembalikan. Jika Anda tidak menentukan nilai, maksimum adalah jumlah log yang dapat muat dalam ukuran respons 1 MB, hingga 10.000 peristiwa log.

## start\_time

Awal rentang waktu untuk peristiwa log, dinyatakan dalam format ISO 8601; misalnya, '2021-01-01T20:00:00Z'. Acara dengan stempel waktu yang sama dengan, atau lebih lambat dari, kali ini disertakan.

## end\_time

Akhir rentang waktu untuk peristiwa log, dinyatakan dalam format ISO 8601; misalnya, '2021-01-01T20:00:00Z'. Acara dengan stempel waktu yang sama dengan, atau lebih lambat dari, kali ini tidak termasuk.

## get\_cluster\_stack\_events

```
get_cluster_stack_events(cluster_name, region, next_token)
```

Dapatkan acara tumpukan untuk cluster tertentu.

Parameter:

### **cluster\_name**(diperlukan)

Nama cluster.

### **region**

ClusterWilayah AWS.

### **next\_token**

Token yang digunakan untuk permintaan paginasi.

## Operasi API gambar

Topik

- [list\\_images](#)
- [build\\_image](#)
- [delete\\_image](#)
- [describe\\_image](#)

## list\_images

```
list_images(image_status, region, next_token)
```

Ambil daftar gambar yang ada.

Parameter:

**image\_status**(diperlukan)

Filter berdasarkan status gambar.

Nilai yang valid: AVAILABLE | PENDING | FAILED

**region**

Daftar gambar dibangun di tertentuWilayah AWS.

**next\_token**

Token untuk digunakan untuk permintaan paginasi.

## build\_image

```
build_image(image_configuration, image_id, suppress_validators,
validation_failure_level, dry_run, rollback_on_failure, region)
```

Buat kustomAWS ParallelClustergambar di Wilayah tertentu.

Parameter:

**image\_configuration**(diperlukan)

Konfigurasi gambar sebagai data Python.

**image\_id**(diperlukan)

ID gambar.

**suppress\_validators**

Mengidentifikasi satu atau beberapa validator konfigurasi gambar untuk ditekan.

Format: (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

Tingkat validasi minimum yang menyebabkan pembuatan gambar gagal. Defaultnya adalah ERROR.

Nilai yang valid: INFO | WARNING | ERROR

### **dry\_run**

Jika disetel ke `True`, AWS ParallelCluster melakukan validasi permintaan tanpa membuat sumber daya apa pun. Anda dapat menggunakan ini untuk memvalidasi konfigurasi gambar. Defaultnya adalah `False`.

### **rollback\_on\_failure**

Jika disetel ke `True`, AWS ParallelCluster secara otomatis memulai rollback tumpukan gambar pada kegagalan. Defaultnya adalah `False`.

### **region**

GambarWilayah AWS.

## **delete\_image**

```
delete_image(image_id, region, force)
```

Hapus gambar di Wilayah tertentu.

Parameter:

#### **image\_id**(diperlukan)

ID gambar.

#### **region**

GambarWilayah AWS.

#### **force**

Jika disetel ke `True`, AWS ParallelCluster memaksa penghapusan jika instance menggunakan AMI atau jika AMI dibagikan. Defaultnya adalah `False`.

## describe\_image

```
describe_image(image_id, region)
```

Dapatkan informasi rinci tentang gambar yang ada.

Parameter:

### **image\_id**(diperlukan)

ID gambar.

### **region**

GambarWilayah AWS.

## Operasi log gambar dan tumpukan

Topik

- [list\\_image\\_log\\_streams](#)
- [get\\_image\\_log\\_events](#)
- [get\\_image\\_stack\\_events](#)
- [list\\_official\\_images](#)

## list\_image\_log\_streams

```
list_image_log_streams(image_id, region, next_token)
```

Daftar aliran log untuk gambar.

Parameter:

### **image\_id**(diperlukan)

ID gambar.

### **region**

GambarWilayah AWS.



## **next\_token**

Token yang digunakan untuk permintaan paginasi.

## **get\_image\_log\_events**

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head,
 limit, start_time, end_time)
```

Dapatkan peristiwa log untuk gambar tertentu dan aliran log.

Parameter:

### **image\_id**(diperlukan)

ID gambar.

### **log\_stream\_name**(diperlukan)

Nama log stream.

### **region**

GambarWilayah AWS.

### **next\_token**

Token yang digunakan untuk permintaan paginasi.

### **start\_from\_head**

Jika disetel ke `True`, AWS ParallelCluster mengembalikan peristiwa log paling awal terlebih dahulu. Jika disetel ke `False`, ia mengembalikan peristiwa log terbaru terlebih dahulu. Defaultnya adalah `False`.

### **limit**

Jumlah maksimum peristiwa log yang dikembalikan. Jika Anda tidak menentukan nilai, maksimum adalah jumlah log yang dapat muat dalam ukuran respons 1 MB, hingga 10.000 peristiwa log.

### **start\_time**

Awal rentang waktu untuk peristiwa log, dinyatakan dalam format ISO 8601; misalnya, '2021-01-01T20:00:00Z'. Acara dengan stempel waktu yang sama dengan, atau lebih lambat dari, kali ini disertakan.

## **end\_time**

Akhir rentang waktu untuk peristiwa log, dinyatakan dalam format ISO 8601; misalnya, '2021-01-01T20:00:00Z'. Acara dengan stempel waktu yang sama dengan, atau lebih lambat dari, kali ini tidak termasuk.

## **get\_image\_stack\_events**

```
get_image_stack_events(image_id, region, next_token)
```

Dapatkan acara tumpukan untuk gambar tertentu.

Parameter:

### **image\_id**(diperlukan)

ID gambar.

### **region**

GambarWilayah AWS.

### **next\_token**

Token yang digunakan untuk permintaan paginasi.

## **list\_official\_images**

```
list_official_images(region,os, architecture)
```

Ambil daftar resmiAWS ParallelClustergambar.

Parameter:

### **region**

GambarWilayah AWS.

### **os**

Filter berdasarkan distribusi sistem operasi. Defaultnya adalah tidak ada penyaringan.

## architecture

Filter berdasarkan arsitektur. Defaultnya adalah tidak ada penyaringan.

## Contoh

Topik

- [Membuat klaster](#)

## Membuat klaster

Saat Anda menjalankan contoh skrip berikut, dengan input yang diberikan disimpan di lingkungan Anda, Anda membuat cluster. Konfigurasi cluster dibuat sebagai tipe data Python berdasarkan [dokumentasi konfigurasi cluster](#).

```
import os
import pprint
import pcluster.lib as pc
pp = pprint.PrettyPrinter()

HEAD_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
COMPUTE_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
KEY_NAME = os.environ["KEY_NAME"]
CONFIG = {'Image': {'Os': 'alinux2'},
 'HeadNode': {'InstanceType': 't2.large',
 'Networking': {'SubnetId': HEAD_NODE_SUBNET},
 'Ssh': {'KeyName': KEY_NAME}},
 'Scheduling': {'Scheduler': 'slurm',
 'SlurmQueues':
 [{'Name': 'queue0',
 'ComputeResources':
 [{'Name': 'queue0-i0', 'InstanceType': 't2.micro',
 'MinCount': 0, 'MaxCount': 10}],
 'Networking': {'SubnetIds': [COMPUTE_NODE_SUBNET]}]
]}

pp.pprint(pc.create_cluster(cluster_name="mycluster", cluster_configuration=CONFIG))
```

Output:

```
{'cluster': {'cloudformationStackArn': 'arn:aws:cloudformation:us-east-2:123456789012:stack/mycluster/00000000-aaaa-1111-999-000000000000',
 'cloudformationStackStatus': 'CREATE_IN_PROGRESS',
 'clusterName': 'mycluster',
 'clusterStatus': 'CREATE_IN_PROGRESS',
 'region': 'us-east-2',
 'scheduler': {'type': 'slurm'},
 'version': '3.7.0'}}
```

## AWS Lambda untuk AWS ParallelCluster Pustaka Python

Anda dapat menerapkan layer Lambda dan runtime untuk mengakses AWS ParallelCluster Pustaka Python. Kami menjadi tuan rumah AWS ParallelCluster zip yang dapat Anda gunakan dengan memasukkan tautan ke file zip seperti yang dijelaskan dalam langkah-langkah berikut. Lambda menggunakan file zip untuk mempersiapkan lingkungan runtime untuk mendukung akses ke pustaka Python. The AWS ParallelCluster Pustaka Python ditambahkan dengan AWS ParallelCluster versi 3.5.0. Anda hanya dapat menggunakan pustaka untuk versi 3.5.0 dan yang lebih baru.

URL file zip yang dihosting dalam format: `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`

### Mulai mengakses AWS ParallelCluster Pustaka Python dengan AWS Lambda

#### Buat layer Lambda

1. Masuk ke AWS Management Console dan navigasikan ke AWS Lambda konsol.
2. Di panel navigasi, pilih Lapisan, maka Buat layer.
3. Masukkan nama untuk layer Anda dan pilih Unggah file dari Amazon S3.
4. Masukkan URL ke file zip: `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/lapisan/aws-parallelcluster/lambda-layer.zip`.
5. Untuk Arsitektur yang kompatibel, pilih x86\_64 arsitektur.
6. Untuk Runtime yang kompatibel, pilih Python 3.9 runtime.
7. Pilih Create (Buat).

#### Gunakan layer Lambda Anda

1. Di panel navigasi konsol Lambda, pilih Fungsi, maka Buat fungsi.

2. Masukkan nama untuk fungsi Anda.
3. Untuk Runtime, pilih Python 3.9 runtime.
4. Untuk Arsitektur, pilih x86\_64 arsitektur.
5. Pilih Buat fungsi.
6. Setelah fungsi dibuat, pilih Lapisan dan pilih Tambahkan lapisan.
7. Pilih Lapisan kustom dan pilih layer yang Anda buat di langkah sebelumnya.
8. Pilih versi layer.
9. Pilih Tambahkan.
10. Lambda Anda memerlukan izin untuk mengelola kluster yang dibuat dengan AWS ParallelCluster. Buat peran Lambda dengan izin yang tercantum di [Kebijakan AWS ParallelCluster pcluster pengguna dasar](#).

Anda sekarang dapat mengakses AWS ParallelCluster dari perpustakaan Python seperti yang dijelaskan dalam [AWS ParallelCluster API pustaka Python](#).

# Bagaimana AWS ParallelCluster berhasil

AWS ParallelCluster dibangun tidak hanya sebagai cara untuk mengelola cluster, tetapi sebagai referensi tentang cara menggunakan AWS layanan untuk membangun lingkungan HPC Anda.

Topik

- [AWS ParallelCluster proses](#)
- [AWS Layanan yang digunakan oleh AWS ParallelCluster](#)
- [AWS ParallelCluster Direktori internal](#)

## AWS ParallelCluster proses

Bagian ini berlaku untuk cluster yang digunakan dengan Slurm. Saat digunakan dengan penjadwal ini, AWS ParallelCluster mengelola penyediaan dan penghapusan node komputasi dengan berinteraksi dengan penjadwal pekerjaan yang mendasarinya.

Untuk cluster HPC yang didasarkan pada AWS Batch, AWS ParallelCluster bergantung pada kemampuan yang disediakan oleh AWS Batch untuk manajemen node komputasi.

### **clustermgtd**

Tugas-tugas berikut dilakukan oleh daemon manajemen cluster.

- Pembersihan partisi tidak aktif
- Manajemen reservasi Slurm dan node yang terkait dengan Blok Kapasitas (lihat bagian berikut)
- Manajemen kapasitas statis: pastikan kapasitas statis selalu naik dan sehat
- Sinkronkan penjadwal dengan Amazon EC2.
- Pembersihan contoh yatim piatu
- Mengembalikan status node penjadwal pada penghentian Amazon EC2 yang terjadi di luar alur kerja penanggulangan
- Manajemen instans Amazon EC2 yang tidak sehat (gagal pemeriksaan kesehatan Amazon EC2)
- Manajemen acara pemeliharaan terjadwal
- Manajemen node Scheduler yang tidak sehat (gagal pemeriksaan kesehatan Scheduler)

## Manajemen reservasi Slurm dan node yang terkait dengan Blok Kapasitas

ParallelCluster mendukung Pemesanan Kapasitas Sesuai Permintaan (ODCR) dan Blok Kapasitas untuk Machine Learning (CB). Tidak seperti ODCR, CB dapat memiliki waktu mulai masa depan dan terikat waktu.

Clustermgtd mencari node yang tidak sehat dalam satu loop dan mengakhiri instans EC2 yang sedang down, menggantinya dengan instance baru jika mereka adalah node statis.

ParallelCluster mengelola node statis yang terkait dengan Blok Kapasitas secara berbeda. AWS ParallelCluster membuat cluster bahkan jika CB belum aktif, dan instance diluncurkan secara otomatis setelah CB aktif.

Node Slurm yang sesuai dengan sumber daya komputasi yang terkait dengan CB yang belum aktif disimpan dalam pemeliharaan sampai waktu mulai CB tercapai. Node slurm akan tetap dalam status reservasi/pemeliharaan yang terkait dengan pengguna admin slurm, yang berarti mereka dapat menerima pekerjaan, tetapi pekerjaan akan tetap tertunda hingga reservasi Slurm dihapus.

Clustermgtd akan secara otomatis membuat/menghapus reservasi Slurm, menempatkan node CB terkait dalam pemeliharaan berdasarkan status CB. Ketika CB akan aktif, reservasi Slurm akan dihapus, node akan mulai dan akan tersedia untuk pekerjaan yang tertunda atau untuk pengiriman pekerjaan baru.

Ketika waktu akhir CB tercapai, node akan dipindahkan kembali ke status reservasi/pemeliharaan. Terserah pengguna untuk mengirimkan ulang/meminta ulang pekerjaan ke antrian/sumber daya komputasi baru saat CB tidak lagi aktif dan instance dihentikan.

## **clusterstatusmgtd**

Daemon manajemen status cluster mengelola pembaruan status armada komputasi. Setiap menit ia mengambil status armada yang disimpan dalam tabel DynamoDB dan mengelola permintaan STOP/START apa pun.

## **computemgtd**

Proses daemon (computemgtd) manajemen komputasi berjalan pada setiap node komputasi cluster. Setiap lima (5) menit, daemon manajemen komputasi menegaskan bahwa node kepala dapat dijangkau dan sehat. Jika lima (5) menit berlalu di mana simpul kepala tidak dapat dicapai atau tidak sehat, simpul komputasi dimatikan.

# AWSlayanan yang digunakan oleh AWS ParallelCluster

Layanan Amazon Web Services (AWS) berikut digunakan olehAWS ParallelCluster.

## Topik

- [Amazon API Gateway](#)
- [AWS Batch](#)
- [AWS CloudFormation](#)
- [Amazon CloudWatch](#)
- [CloudWatch Acara Amazon](#)
- [CloudWatch Log Amazon](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx for Lustre](#)
- [Amazon FSx untuk ONTAP NetApp](#)
- [Amazon FSx for OpenZFS](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon RDS](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service](#)
- [Amazon Simple Storage Service](#)
- [Amazon VPC](#)
- [Elastic Fabric Adapter](#)
- [EC2 Image Builder](#)
- [NICE DCV](#)



## Amazon API Gateway

Amazon API Gateway adalah AWS layanan untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan REST, HTTP, dan WebSocket API pada skala apa pun

AWS ParallelCluster menggunakan API Gateway untuk meng-host AWS ParallelCluster API.

Untuk informasi lebih lanjut tentang AWS Batch, lihat <https://aws.amazon.com/api-gateway/> dan <https://docs.aws.amazon.com/apigateway/>.

## AWS Batch

AWS Batch adalah layanan penjadwal pekerjaan AWS terkelola. Ini secara dinamis menyediakan kuantitas dan jenis sumber daya komputasi yang optimal (misalnya, CPU atau instance yang dioptimalkan memori) dalam cluster. AWS Batch Sumber daya ini disediakan berdasarkan persyaratan spesifik pekerjaan batch Anda, termasuk persyaratan volume. Dengan AWS Batch, Anda tidak perlu menginstal atau mengelola perangkat lunak komputasi batch tambahan atau cluster server untuk menjalankan pekerjaan Anda secara efektif.

AWS Batch hanya digunakan dengan AWS Batch cluster.

Untuk informasi lebih lanjut tentang AWS Batch, lihat <https://aws.amazon.com/batch/> dan <https://docs.aws.amazon.com/batch/>.

## AWS CloudFormation

AWS CloudFormation adalah infrastructure-as-code layanan yang menyediakan bahasa umum untuk model dan penyediaan AWS dan sumber daya aplikasi pihak ketiga di lingkungan cloud Anda. Ini adalah layanan utama yang digunakan oleh AWS ParallelCluster. Setiap cluster di AWS ParallelCluster direpresentasikan sebagai tumpukan, dan semua sumber daya yang dibutuhkan oleh setiap cluster didefinisikan dalam AWS ParallelCluster AWS CloudFormation template. Dalam kebanyakan kasus, perintah AWS ParallelCluster CLI secara langsung sesuai dengan perintah AWS CloudFormation stack, seperti membuat, memperbarui, dan menghapus perintah. Instans yang diluncurkan dalam kluster membuat panggilan HTTPS ke AWS CloudFormation titik akhir di Wilayah AWS mana cluster diluncurkan.

Untuk informasi lebih lanjut tentang AWS CloudFormation, lihat <https://aws.amazon.com/cloudformation/> dan <https://docs.aws.amazon.com/cloudformation/>.

## Amazon CloudWatch

Amazon CloudWatch (CloudWatch) adalah layanan pemantauan dan observabilitas yang memberi Anda data dan wawasan yang dapat ditindaklanjuti. Wawasan ini dapat digunakan untuk memantau aplikasi Anda, menanggapi perubahan kinerja dan pengecualian layanan, dan mengoptimalkan pemanfaatan sumber daya. InAWS ParallelCluster, CloudWatch digunakan untuk dasbor, untuk memantau dan mencatat langkah-langkah pembuatan gambar Docker dan output AWS Batch pekerjaan.

Sebelum AWS ParallelCluster versi 2.10.0, hanya CloudWatch digunakan dengan AWS Batch cluster.

Untuk informasi lebih lanjut tentang CloudWatch, lihat <https://aws.amazon.com/cloudwatch/> dan <https://docs.aws.amazon.com/cloudwatch/>.

## CloudWatch Acara Amazon

Amazon CloudWatch CloudWatch Events (Events) memberikan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan dalam sumber daya Amazon Web Services (AWS). Dengan menggunakan aturan sederhana yang dapat Anda siapkan dengan cepat, Anda dapat mencocokkan acara dan merutuskannya ke satu atau beberapa fungsi atau pengaliran target. DiAWS ParallelCluster, CloudWatch Acara digunakan untuk AWS Batch pekerjaan.

Untuk informasi lebih lanjut tentang CloudWatch Acara, lihat <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cwe-now-eb>.

## CloudWatch Log Amazon

Amazon CloudWatch CloudWatch Logs (Log) adalah salah satu fitur inti Amazon CloudWatch. Anda dapat menggunakannya untuk memantau, menyimpan, melihat, dan mencari file log untuk banyak komponen yang digunakan olehAWS ParallelCluster.

Sebelum AWS ParallelCluster versi 2.6.0, CloudWatch Log hanya digunakan dengan AWS Batch cluster.

Untuk informasi selengkapnya, lihat [Integrasi dengan Amazon CloudWatch Logs](#).

## AWS CodeBuild

AWS CodeBuild(CodeBuild) adalah layanan integrasi berkelanjutan AWS terkelola yang mematuhi kode sumber, menjalankan pengujian, dan menghasilkan paket perangkat lunak yang siap

digunakan. InAWS ParallelCluster, CodeBuild digunakan untuk secara otomatis dan transparan membangun gambar Docker saat cluster dibuat.

CodeBuild hanya digunakan dengan AWS Batch cluster.

Untuk informasi lebih lanjut tentang CodeBuild, lihat <https://aws.amazon.com/codebuild/> dan <https://docs.aws.amazon.com/codebuild/>.

## Amazon DynamoDB

Amazon DynamoDB (DynamoDB) adalah layanan database NoSQL yang cepat dan fleksibel. Ini digunakan untuk menyimpan informasi status minimal cluster. Node kepala melacak instance yang disediakan dalam tabel DynamoDB.

DynamoDB tidak digunakan dengan cluster. AWS Batch

Untuk informasi lebih lanjut tentang DynamoDB, lihat <https://aws.amazon.com/dynamodb/> dan <https://docs.aws.amazon.com/dynamodb/>.

## Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) adalah layanan penyimpanan blok berkinerja tinggi yang menyediakan penyimpanan persisten untuk volume bersama. Semua pengaturan Amazon EBS dapat dilewatkan melalui konfigurasi. Volume Amazon EBS dapat diinisialisasi kosong atau dari snapshot Amazon EBS yang ada.

Untuk informasi selengkapnya tentang Amazon EBS, lihat <https://aws.amazon.com/ebs/> dan <https://docs.aws.amazon.com/ebs/>.

## Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi untuk. AWS ParallelCluster Node head dan compute adalah instans Amazon EC2. Tipe instans apa pun yang mendukung HVM dapat dipilih. Node head dan compute dapat berupa tipe instance yang berbeda. Selain itu, jika beberapa antrian digunakan, beberapa atau semua node komputasi juga dapat diluncurkan sebagai Instans Spot. Volume penyimpanan instans yang ditemukan pada instance dipasang sebagai volume LVM bergaris.

Untuk informasi selengkapnya tentang Amazon EC2, lihat <https://aws.amazon.com/ec2/> dan <https://docs.aws.amazon.com/ec2/>.

## Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) adalah registri kontainer Docker yang dikelola sepenuhnya yang memudahkan untuk menyimpan, mengelola, dan menyebarkan gambar kontainer Docker. Di AWS ParallelCluster, Amazon ECR menyimpan gambar Docker yang dibuat saat cluster dibuat. Gambar Docker kemudian digunakan oleh AWS Batch untuk menjalankan kontainer untuk pekerjaan yang dikirimkan.

Amazon ECR hanya digunakan dengan AWS Batch cluster.

Untuk informasi lebih lanjut, lihat <https://aws.amazon.com/ecr/> dan <https://docs.aws.amazon.com/ecr/>.

## Amazon EFS

Amazon Elastic File System (Amazon EFS) menyediakan sistem file NFS elastis yang sederhana, dapat diskalakan, dan dikelola sepenuhnya untuk digunakan dengan AWS Cloud layanan dan sumber daya lokal. Amazon EFS digunakan saat [EfsSettings](#) ditentukan. Support untuk Amazon EFS telah ditambahkan dalam AWS ParallelCluster versi 2.1.0.

Untuk informasi selengkapnya tentang Amazon EFS, lihat <https://aws.amazon.com/efs/> dan <https://docs.aws.amazon.com/efs/>.

## Amazon FSx for Lustre

FSx for Lustre menyediakan sistem file berkinerja tinggi yang menggunakan sistem file Lustre open-source. FSx for Lustre digunakan ketika ditentukan. [FsxLustreSettingsproperty](#) Support untuk FSx for Lustre ditambahkan di versi 2.2.1. AWS ParallelCluster

[Untuk informasi lebih lanjut tentang FSx for Lustre, lihat https://aws.amazon.com/fsx/lustre/ dan https://docs.aws.amazon.com/fsx/.](https://aws.amazon.com/fsx/lustre/)

## Amazon FSx untuk ONTAP NetApp

FSx untuk ONTAP menyediakan sistem penyimpanan bersama yang dikelola sepenuhnya yang dibangun di atas sistem file ONTAP NetApp yang populer. FSx untuk ONTAP digunakan ketika [FsxOntapSettingsproperty](#) ditentukan. Support untuk FSx untuk ONTAP telah ditambahkan dalam AWS ParallelCluster versi 3.2.0.

[Untuk informasi lebih lanjut tentang FSx untuk ONTAP, lihat https://aws.amazon.com/fsx/netapp-ontap/ dan https://docs.aws.amazon.com/fsx/.](https://aws.amazon.com/fsx/netapp-ontap/)

## Amazon FSx for OpenZFS

FSx untuk OpenZFS menyediakan sistem penyimpanan bersama yang dikelola sepenuhnya yang dibangun di atas sistem file OpenZFS yang populer. FSx untuk OpenZFS digunakan ketika ditentukan. [FsxOpenZfsSettingsproperti](#) Support untuk FSx untuk OpenZFS telah ditambahkan dalam versi 3.2.0. AWS ParallelCluster

[Untuk informasi lebih lanjut tentang FSx untuk OpenZFS, lihat https://aws.amazon.com/fsx/openzfs/ dan https://docs.aws.amazon.com/fsx/.](https://aws.amazon.com/fsx/openzfs/)

## AWS Identity and Access Management

AWS Identity and Access Management(IAM) digunakan di dalam AWS ParallelCluster untuk menyediakan peran IAM yang paling tidak memiliki hak istimewa untuk Amazon EC2 untuk instance yang spesifik untuk setiap cluster individu. AWS ParallelClusterinstance diberikan akses hanya ke panggilan API tertentu yang diperlukan untuk menerapkan dan mengelola klaster.

Dengan AWS Batch cluster, peran IAM juga dibuat untuk komponen yang terlibat dengan proses pembuatan image Docker saat cluster dibuat. Komponen-komponen ini termasuk fungsi Lambda yang diizinkan untuk menambah dan menghapus gambar Docker ke dan dari repositori Amazon ECR. Mereka juga menyertakan fungsi yang diizinkan untuk menghapus bucket Amazon S3 yang dibuat untuk cluster dan CodeBuild proyek. Ada juga peran untuk AWS Batch sumber daya, contoh, dan pekerjaan.

Untuk informasi lebih lanjut tentang IAM, lihat <https://aws.amazon.com/iam/> dan <https://docs.aws.amazon.com/iam/>.

## AWS Lambda

AWS Lambda(Lambda) menjalankan fungsi yang mengatur pembuatan gambar Docker. Lambda juga mengelola pembersihan sumber daya cluster khusus, seperti gambar Docker yang disimpan di repositori Amazon ECR dan di Amazon S3.

[Untuk informasi lebih lanjut tentang Lambda, lihat https://aws.amazon.com/lambda/ dan https://docs.aws.amazon.com/lambda/.](https://aws.amazon.com/lambda/)

## Amazon RDS

Amazon Relational Database Service(Amazon RDS) adalah layanan web yang membuatnya lebih mudah untuk mengatur, mengoperasikan, dan menskalakan database relasional di AWS Cloud.

AWS ParallelCluster menggunakan Amazon RDS untuk AWS Batch dan Slurm.

Untuk informasi selengkapnya tentang Amazon RDS, lihat <https://aws.amazon.com/rds/> dan <https://docs.aws.amazon.com/rds/>.

## Amazon Route 53

Amazon Route 53 (Route 53) digunakan untuk membuat zona yang dihosting dengan nama host dan nama domain yang memenuhi syarat untuk setiap node komputasi.

Untuk informasi lebih lanjut tentang Route 53, lihat <https://aws.amazon.com/route53/> dan <https://docs.aws.amazon.com/route53/>.

## Amazon Simple Notification Service

(Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan dari penerbit ke pelanggan (juga dikenal sebagai produsen dan konsumen).

AWS ParallelCluster menggunakan Amazon SNS untuk hosting API.

Untuk informasi selengkapnya tentang Amazon SNS, lihat <https://aws.amazon.com/sns/> dan <https://docs.aws.amazon.com/sns/>.

## Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) AWS ParallelCluster menyimpan template yang terletak di masing-masing Wilayah AWS. AWS ParallelCluster dapat dikonfigurasi untuk memungkinkan alat CLI/SDK menggunakan Amazon S3.

AWS ParallelCluster juga membuat bucket Amazon S3 Akun AWS untuk menyimpan sumber daya yang digunakan oleh cluster Anda, seperti file konfigurasi kluster. AWS ParallelCluster memelihara satu bucket Amazon S3 di setiap Wilayah AWS tempat Anda membuat cluster.

Saat Anda menggunakan AWS Batch cluster, bucket Amazon S3 di akun Anda digunakan untuk menyimpan data terkait. Misalnya, bucket menyimpan artefak yang dibuat saat image dan skrip Docker dibuat dari pekerjaan yang dikirimkan.

Untuk informasi lebih lanjut, lihat <https://aws.amazon.com/s3/> dan <https://docs.aws.amazon.com/s3/>.

## Amazon VPC

VPC Amazon mendefinisikan jaringan yang digunakan oleh node di cluster Anda.

Untuk informasi selengkapnya tentang Amazon VPC, lihat <https://aws.amazon.com/vpc/> dan <https://docs.aws.amazon.com/vpc/>.

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) adalah antarmuka jaringan untuk instans Amazon EC2 yang dapat digunakan pelanggan untuk menjalankan aplikasi yang membutuhkan komunikasi antar-simpul tingkat tinggi dalam skala besar. AWS

Untuk informasi lebih lanjut tentang EC2 Image Builder, [lihat](https://aws.amazon.com/hpc/efa/) <https://aws.amazon.com/hpc/efa/>.

## EC2 Image Builder

EC2 Image Builder adalah layanan AWS terkelola penuh yang membantu Anda mengotomatiskan pembuatan, pengelolaan, dan penyebaran gambar yang disesuaikan, aman, up-to-date dan server.

AWS ParallelCluster menggunakan Image Builder untuk membuat dan mengelola AWS ParallelCluster gambar.

Untuk informasi selengkapnya tentang EC2 Image Builder, [lihat](https://aws.amazon.com/image-builder/) <https://aws.amazon.com/image-builder/> [dan](https://docs.aws.amazon.com/imagebuilder/) <https://docs.aws.amazon.com/imagebuilder/>.

## NICE DCV

NICE DCV adalah protokol tampilan jarak jauh berkinerja tinggi yang menyediakan cara aman untuk mengirimkan desktop jarak jauh dan streaming aplikasi ke perangkat apa pun dalam berbagai kondisi jaringan. NICE DCV digunakan ketika [Dcv](#) pengaturan [Bagian HeadNode](#) ditentukan. Support untuk NICE DCV telah ditambahkan dalam AWS ParallelCluster versi 2.5.0.

Untuk informasi lebih lanjut tentang NICE DCV, lihat <https://aws.amazon.com/hpc/dcv/> dan <https://docs.aws.amazon.com/dcv/>.

## AWS ParallelCluster Direktori internal

Ada beberapa direktori internal yang AWS ParallelCluster digunakan untuk berbagi data dalam cluster. Direktori berikut dibagi antara node kepala, node komputasi, dan node login:


`/opt/slurm`

`/opt/intel`

`/opt/parallelcluster/shared` (only with compute nodes)

`/opt/parallelcluster/shared_login_nodes` (only with login nodes)

`/home` (unless specified in `SharedStorage`)

 Note

Secara default direktori ini dibuat pada volume EBS node kepala dan dibagikan sebagai ekspor NFS ke node komputasi dan login. Mulai dari AWS ParallelCluster 3.8 Anda dapat mengaktifkan AWS ParallelCluster untuk membuat dan mengelola sistem file Amazon EFS untuk meng-host dan membagikan direktoris ini dengan menyetel parameter ke `efs`.

[SharedStorageType](#)

Ketika skala cluster keluar, ekspor NFS melalui volume EBS dapat menimbulkan kemacetan kinerja. Dengan menggunakan EFS, Anda dapat menghindari ekspor NFS saat klaster Anda mengurangi skala dan menghindari kemacetan kinerja yang terkait dengannya.



# Tutorial

Tutorial berikut menunjukkan cara memulai dengan AWS ParallelCluster versi 3, dan memberikan panduan praktik terbaik untuk beberapa tugas umum.

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Layanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelClusterBiaya UI](#).

## Topik

- [Menjalankan pekerjaan pertama Anda diAWS ParallelCluster](#)
- [Membangun kebiasaanAWS ParallelClusterAMI](#)
- [Mengintegrasikan Active Directory](#)
- [Mengonfigurasi enkripsi penyimpanan bersama dengan kunci AWS KMS](#)
- [Menjalankan pekerjaan dalam kluster mode antrian ganda](#)
- [Menggunakan API AWS ParallelCluster ini](#)
- [Membuat cluster dengan akuntansi Slurm](#)
- [Mengembalikan ke versi dokumen Manajer AWS Sistem sebelumnya](#)
- [Membuat cluster dengan AWS CloudFormation](#)
- [AWS ParallelClusterIntegrasi UI dengan Pusat Identitas](#)

## Menjalankan pekerjaan pertama Anda diAWS ParallelCluster

Tutorial ini memandu Anda menjalankan pekerjaan Hello World pertama Anda diAWS ParallelCluster

Saat menggunakanAWS ParallelClusterantarmuka baris perintah (CLI) atau API, Anda hanya membayar untukAWSsumber daya yang dibuat saat Anda membuat atau memperbaruiAWS ParallelClustergambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Layanan yang digunakan oleh AWS ParallelCluster](#).

The AWS ParallelCluster UI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya di dalam AWS Kategori Tier Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Biaya UI](#).

## Prasyarat

- AWS ParallelCluster [diinstal](#).
- The AWS CLI [diinstal dan dikonfigurasi](#).
- Anda memiliki [Pasangan kunci EC2](#).
- Anda memiliki peran IAM dengan [izin](#) diperlukan untuk menjalankan [pcluster](#) CLI.

## Memverifikasi instalasi Anda

Pertama, kami memverifikasi bahwa AWS ParallelCluster benar, termasuk ketergantungan Node.js, diinstal dan dikonfigurasi.

```
$ node --version
v16.8.0
$ pcluster version
{
 "version": "3.7.0"
}
```

Ini mengembalikan versi berjalan AWS ParallelCluster.

## Membuat cluster pertama Anda

Sekarang saatnya membuat cluster pertama Anda. Karena beban kerja untuk tutorial ini tidak intensif kinerja, kita dapat menggunakan ukuran instance default dari `2.micro`. (Untuk beban kerja produksi, Anda memilih ukuran instans yang paling sesuai dengan kebutuhan Anda.) Mari kita panggil cluster Anda `hello-world`.

```
$ pcluster create-cluster \
 --cluster-name hello-world \
 --cluster-configuration hello-world.yaml
```

**Note**

TheWilayah AWSuntuk digunakan harus ditentukan untuk sebagian besarpclusterperintah. Jika tidak ditentukan dalamAWS\_DEFAULT\_REGIONvariabel lingkungan, atauregionpengaturan di[default]bagian dari~/.aws/configberkas, lalu--regionparameter harus disediakan padapclusterbaris perintah.

Jika output memberi Anda pesan tentang konfigurasi, Anda perlu menjalankan yang berikut ini untuk mengkonfigurasiAWS ParallelCluster:

```
$ pcluster configure --config hello-world.yaml
```

Jikapcluster [create-cluster](#)perintah berhasil, Anda melihat output mirip dengan berikut ini:

```
{
 "cluster": {
 "clusterName": "hello-world",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:xxx:stack/xxx",
 "region": "...",
 "version": "...",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Anda memantau pembuatan cluster menggunakan:

```
$ pcluster describe-cluster --cluster-name hello-world
```

TheclusterStatuslaporan"CREATE\_IN\_PROGRESS"sementara cluster sedang dibuat. TheclusterStatustransisi ke"CREATE\_COMPLETE"ketika cluster berhasil dibuat. Outputnya juga memberi kitapublicIpAddressdanprivateIpAddressdari simpul kepala kita.

## Masuk ke node kepala Anda

Gunakan file pem OpenSSH Anda untuk masuk ke node kepala Anda.

```
$ pcluster ssh --cluster-name hello-world -i /path/to/keyfile.pem
```

Setelah Anda masuk, jalankan perintah `sinfo` untuk memverifikasi bahwa node komputasi Anda telah diatur dan dikonfigurasi.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 10 idle~ queue1-dy-queue1t2micro-[1-10]
```

Output menunjukkan bahwa kita memiliki satu antrian di cluster kita, dengan hingga sepuluh node.

## Menjalankan pekerjaan pertama Anda menggunakan Slurm

Selanjutnya, kami membuat pekerjaan yang tidur sebentar dan kemudian mengeluarkan nama hostnya sendiri. Buat file bernama `hellojob.sh`, dengan konten berikut.

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

Selanjutnya, kirimkan pekerjaan menggunakan `sbatch`, dan verifikasi bahwa itu berjalan.

```
$ sbatch hellojob.sh
Submitted batch job 2
```

Sekarang, Anda dapat melihat antrian Anda dan memeriksa status pekerjaan. Penyediaan instans Amazon EC2 baru dimulai di latar belakang. Anda dapat memantau status instance cluster dengan `sinfo` perintah.

```
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 queue1 hellojob ec2-user CF 3:30 1 queue1-dy-
queue1t2micro-1
```

Output menunjukkan bahwa pekerjaan telah diserahkan ke `queue1`. Tunggu 30 detik hingga pekerjaan selesai, lalu jalankan `squeue` lagi.

```
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Sekarang tidak ada pekerjaan dalam antrian, kita dapat memeriksa output di direktori kita saat ini.

```
$ ls -l
total 8
-rw-rw-r-- 1 ec2-user ec2-user 57 Sep 1 14:25 hellojob.sh
-rw-rw-r-- 1 ec2-user ec2-user 43 Sep 1 14:30 slurm-2.out
```

Dalam output, kita melihat a"out"berkas. Kita bisa melihat output dari pekerjaan kita:

```
$ cat slurm-2.out
Hello World from queue1-dy-queue1t2micro-1
```

Outputnya juga menunjukkan bahwa pekerjaan kami berhasil berjalan pada instancequeue1-dy-queue1t2micro-1.

Di cluster yang baru saja Anda buat, hanya direktori home yang dibagikan di antara semua node cluster.

Untuk mempelajari lebih lanjut tentang membuat dan menggunakan cluster, lihat[Praktik terbaik](#).

Jika aplikasi Anda memerlukan perangkat lunak, pustaka, atau data bersama, pertimbangkan opsi berikut:

- MembangunAWS ParallelClustermengaktifkan AMI kustom yang menyertakan perangkat lunak Anda seperti yang dijelaskan dalam[Membangun kebiasaanAWS ParallelClusterAMI](#).
- Gunakan[StorageSettings](#)opsi diAWS ParallelClusterfile konfigurasi untuk menentukan sistem file bersama dan menyimpan perangkat lunak yang Anda instal di lokasi pemasangan yang ditentukan.
- Gunakan[Tindakan bootstrap kustom](#)untuk mengotomatiskan prosedur bootstrap dari setiap node cluster Anda.

## Membangun kebiasaanAWS ParallelClusterAMI

Saat menggunakanAWS ParallelClusterantarmuka baris perintah (CLI) atau API, Anda hanya membayarAWSsumber daya yang dibuat saat Anda membuat atau memperbaruiAWS ParallelClustergambar dan cluster. Untuk informasi selengkapnya, lihat [AWSlayanan yang digunakan oleh AWS ParallelCluster](#).

TheAWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya di dalamAWSKategori Tingkat Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelClusterBiaya UI](#).

### Important

Jika Anda membuat AMI kustom, Anda harus mengulangi langkah-langkah yang Anda gunakan untuk membuat AMI kustom Anda dengan setiap yang baru AWS ParallelCluster melepaskan.

Sebelum membaca lebih lanjut, kami sarankan Anda terlebih dahulu meninjau [Tindakan bootstrap kustom](#) bagian. Tentukan apakah modifikasi yang ingin Anda buat dapat ditulis dan didukung di masa mendatang AWS ParallelCluster rilis.

Meskipun membangun AMI khusus secara umum tidak ideal, ada skenario khusus di mana membangun AMI khusus untuk AWS ParallelCluster diperlukan. Tutorial ini mencakup cara membangun AMI kustom untuk skenario ini.

### Prasyarat

- AWS ParallelCluster [diinstal](#).
- The AWS CLI [diinstal dan dikonfigurasi](#).
- Anda memiliki [Pasangan kunci EC2](#).
- Anda memiliki peran IAM dengan [izin](#) diperlukan untuk menjalankan [pcluster](#) CLI dan buat gambar.

## Cara menyesuaikan AWS ParallelCluster AMI

Ada dua cara untuk membangun kustom AWS ParallelCluster AMI. Salah satu dari dua metode ini adalah membangun AMI baru menggunakan AWS ParallelCluster CLI. Metode lain mengharuskan Anda untuk membuat modifikasi manual untuk membangun AMI baru yang tersedia di bawah Anda Akun AWS.

## Bangun kustom AWS ParallelCluster AMI

Jika Anda memiliki AMI dan perangkat lunak yang disesuaikan, Anda dapat menerapkan perubahan yang diperlukan oleh AWS ParallelCluster di atasnya. AWS ParallelCluster bergantung pada layanan EC2 Image Builder untuk membangun AMI yang disesuaikan. Untuk informasi lebih lanjut, lihat [Panduan Pengguna Pembuat Gambar](#).

Poin kunci:

- Prosesnya memakan waktu sekitar 1 jam. Waktu ini dapat bervariasi jika ada tambahan [Build/Components](#) untuk diinstal pada waktu pembuatan.
- AMI ditandai dengan versi komponen utama. Ini termasuk kernel, scheduler, dan [EFA](#) sopir. Subset dari versi komponen juga dilaporkan dalam deskripsi AMI.
- Mulai dari [AWS ParallelCluster 3.0.0](#), satu set baru perintah CLI dapat digunakan untuk mengelola siklus hidup gambar. Ini termasuk [build-image](#), [list-images](#), [describe-image](#), dan [delete-image](#).
- Metode ini dapat diulang. Anda dapat menjalankannya kembali untuk menjaga AMI diperbarui (misalnya, pembaruan OS), dan kemudian menggunakannya saat Anda memperbarui cluster yang ada.

#### Note

Jika Anda menggunakan metode ini di [AWS China Partition](#), Anda mungkin mendapatkan kesalahan jaringan. Misalnya, Anda mungkin melihat kesalahan ini dari `cluster build-image` perintah saat mengunduh paket dari GitHub atau dari repositori OS. Jika ini terjadi, kami sarankan Anda menggunakan salah satu metode alternatif berikut:

1. Ikuti [Memodifikasi AWS ParallelCluster AMI](#) pendekatan yang melewati perintah ini.
2. Bangun gambar di Partisi dan Wilayah lain, seperti `us-east-1`, dan kemudian simpan-kembalikan untuk memindahkannya ke Wilayah China. Untuk informasi lebih lanjut, lihat [Simpan dan pulihkan AMI menggunakan S3](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

#### Langkah:

1. Konfigurasi akun AWS kredensialnya sehingga [AWS ParallelCluster](#) klien dapat melakukan panggilan ke [AWS Operasi API](#) atas nama Anda. Untuk daftar izin yang diperlukan, lihat [AWS Identity and Access Management izin di AWS ParallelCluster](#).
2. Buat dasar membangun gambar file konfigurasi. Untuk melakukan ini, tentukan [InstanceType](#) untuk digunakan untuk membangun gambar dan [ParentImage](#). Ini digunakan sebagai titik awal untuk membuat AMI. Untuk informasi selengkapnya tentang parameter build opsional, lihat [Konfigurasi Gambar](#).

```
Build:
InstanceType: <BUILD_INSTANCE_TYPE>
```

```
ParentImage: <BASE_AMI_ID>
```

- Gunakan perintah CLI `pcluster build-image` untuk membangun sebuah AWS ParallelCluster AMI mulai dari AMI yang Anda sediakan sebagai basis.

```
$ pcluster build-image --image-id IMAGE_ID --image-configuration IMAGE_CONFIG.yaml --
region REGION
 {
 "image": {
 "imageId": "IMAGE_ID",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
 "region": "us-east-1",
 "version": "3.7.0"
 }
}
```

#### Warning

`pcluster build-image` menggunakan VPC default. Jika Anda menghapus VPC default menggunakan AWS Control Tower atau AWS Zona Pendaratan, ID subnet harus ditentukan dalam file konfigurasi gambar. Untuk informasi selengkapnya, lihat [SubnetId](#).

Untuk daftar parameter lainnya, lihat [pcluster build-image](#) halaman referensi perintah. Hasil dari perintah sebelumnya adalah sebagai berikut:

- SEBUAH CloudFormation stack dibuat berdasarkan konfigurasi gambar. Tumpukan mencakup semua sumber daya Pembuat Gambar EC2 yang diperlukan untuk build.
- Sumber daya yang dibuat termasuk Pembuat Gambar resmi AWS ParallelCluster komponen yang komponen Image Builder kustom dapat ditambahkan ke. Untuk mempelajari cara membuat komponen kustom, lihat [Contoh AMI kustom](#) di Lokakarya HPC untuk Pelanggan Sektor Publik.
- EC2 Image Builder meluncurkan instance build, menerapkan AWS ParallelCluster buku masak, instal AWS ParallelCluster tumpukan perangkat lunak, dan melakukan tugas konfigurasi yang diperlukan. The AWS ParallelCluster buku masak digunakan untuk membangun dan bootstrap AWS ParallelCluster.
- Instance dihentikan dan AMI baru dibuat darinya.



- Contoh lain diluncurkan dari AMI yang baru dibuat. Selama fase pengujian, EC2 Image Builder menjalankan pengujian yang ditentukan dalam komponen Image Builder.
  - Jika build berhasil, tumpukan akan dihapus. Jika build gagal, tumpukan dipertahankan dan tersedia untuk diperiksa.
4. Anda dapat memantau status proses build dengan menjalankan perintah berikut. Setelah build selesai, Anda dapat menjalankannya untuk mengambil ID AMI yang diberikan dalam respons.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION

BEFORE COMPLETE
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-
delete.s3.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/
configs/image-config.yaml?... ",
 },
 "imageId": "IMAGE_ID",
 "imagebuilderImageStatus": "BUILDING",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
 "region": "us-east-1",
 "version": "3.7.0",
 "cloudformationStackTags": [
 {
 "value": "3.7.0",
 "key": "parallelcluster:version"
 },
 {
 "value": "IMAGE_ID",
 "key": "parallelcluster:image_name"
 },
 ...
],
 "imageBuildLogsArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-IMAGE_ID",
 "cloudformationStackCreationTime": "2022-04-05T21:36:26.176Z"
}

AFTER COMPLETE
{
```

```

"imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
},
"imageId": "IMAGE_ID",
"imageBuildStatus": "BUILD_COMPLETE",
"region": "us-east-1",
"ec2AmiInfo": {
 "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
 "amiId": "ami-1234stuv5678wxyz",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
 "state": "AVAILABLE",
 "tags": [
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 ...
],
 "architecture": "x86_64"
},
"version": "3.7.0"
}

```

5. Untuk membuat klaster Anda, masukkan ID AMI di [CustomAmi](#) bidang dalam konfigurasi cluster Anda.

## Pemecahan masalah dan pemantauan proses pembuatan AMI

Pembuatan gambar selesai dalam waktu sekitar satu jam. Anda dapat memantau proses dengan menjalankan [pcluster describe-image](#) perintah atau perintah pengambilan log.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

The [build-image](#) perintah menciptakan CloudFormation tumpukan dengan semua sumber daya EC2 yang diperlukan untuk membangun gambar, dan meluncurkan proses EC2 Image Builder.

Setelah menjalankan [build-image](#) perintah, itu mungkin untuk mengambil CloudFormation tumpukan acara dengan menggunakan [pcluster get-image-stack-events](#). Anda dapat memfilter

hasil dengan `--queryparameter` untuk melihat peristiwa terbaru. Untuk informasi lebih lanjut, lihat [Penyaringan AWS CLI keluaran](#) di [AWS Command Line Interface Panduan Pengguna](#).

```
$ pcluster get-image-stack-events --image-id IMAGE_ID --region REGION --query
"events[0]"
{
 "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
 "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-IMAGE_ID/3.7.0/1",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "Resource creation Initiated",
 "resourceProperties": "{ \"InfrastructureConfigurationArn\":
\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-abcd1234-ef56-gh78-ij90-1234abcd5678\", \"ImageRecipeArn\":
\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/parallelclusterimage-
IMAGE_ID/3.7.0\", \"DistributionConfigurationArn\": \"arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-abcd1234-ef56-
gh78-ij90-1234abcd5678\", \"Tags\": { \"parallelcluster:image_name\": \"IMAGE_ID\",
\"parallelcluster:image_id\": \"IMAGE_ID\" } }",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/IMAGE_ID/abcd1234-
ef56-gh78-ij90-1234abcd5678",
 "stackName": "IMAGE_ID",
 "logicalResourceId": "ParallelClusterImage",
 "resourceType": "AWS::ImageBuilder::Image",
 "timestamp": "2022-04-05T21:39:24.725Z"
}
```

Setelah sekitar 15 menit, peristiwa tumpukan muncul di entri peristiwa log yang terkait dengan pembuatan Pembuat Gambar. Anda sekarang dapat membuat daftar aliran log gambar dan memantau langkah-langkah Pembuat Gambar dengan menggunakan [pcluster list-image-log-streams](#) dan [pcluster get-image-log-events](#) perintah.

```
$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
--query 'logStreams[*].logStreamName'

"3.7.0/1"
]

$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
--log-stream-name 3.7.0/1 --limit 3
{
 "nextToken": "f/36295977202298886557255241372854078762600452615936671762",
```

```

"prevToken": "b/36295977196879805474012299949460899222346900769983430672",
"events": [
 {
 "message": "ExecuteBash: FINISHED EXECUTION",
 "timestamp": "2022-04-05T22:13:26.633Z"
 },
 {
 "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-abcd1234-ef56-gh78-ij90-1234abcd5678/3.7.0/1",
 "timestamp": "2022-04-05T22:13:26.741Z"
 },
 {
 "message": "TOE has completed execution successfully",
 "timestamp": "2022-04-05T22:13:26.819Z"
 }
]
}

```

Lanjutkan untuk memeriksa dengan [describe-image](#) perintah sampai Anda melihat `BUILD_COMPLETE` status.

```

$ pcluster describe-image --image-id IMAGE_ID --region REGION
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
 },
 "imageId": "IMAGE_ID",
 "imageBuildStatus": "BUILD_COMPLETE",
 "region": "us-east-1",
 "ec2AmiInfo": {
 "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
 "amiId": "ami-1234stuv5678wxyz",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
 "state": "AVAILABLE",
 "tags": [
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 }
],
 },
}

```

```

 ...
],
 "architecture": "x86_64"
},
"version": "3.7.0"
}

```

Jika Anda perlu memecahkan masalah pembuatan AMI kustom, buat arsip log gambar seperti yang dijelaskan dalam langkah-langkah berikut.

Dimungkinkan untuk mengarsipkan log di bucket Amazon S3 atau dalam file lokal, tergantung pada `--outputparameter`.

```

$ pcluster export-image-logs --image-id IMAGE_ID --region REGION \
--bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER \
{
 "url": "https://BUCKET_NAME.s3.us-east-1.amazonaws.com/BUCKET-FOLDER/IMAGE_ID-
logs-202209071136.tar.gz?AWSAccessKeyId=..."
}

$ pcluster export-image-logs --image-id IMAGE_ID \
--region REGION --bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER --output-file /tmp/
archive.tar.gz
{
 "path": "/tmp/archive.tar.gz"
}

```

Arsip berisi CloudWatch Aliran Log yang terkait dengan proses Pembuat Gambar dan AWS CloudFormation tumpukan acara. Perintah mungkin memakan waktu beberapa menit untuk dijalankan.

## Mengelola AMI Kustom

Mulai dari AWS ParallelCluster 3.0.0, serangkaian perintah baru telah ditambahkan di CLI untuk membangun, memantau, dan mengelola siklus hidup gambar. Untuk informasi selengkapnya tentang perintah, lihat [perintah pcluster](#).

## Memodifikasi AWS ParallelCluster AMI

Metode ini terdiri dari memodifikasi pejabat AWS ParallelCluster AMI dengan menambahkan kustomisasi di atasnya. Basis AWS ParallelCluster AMI diperbarui dengan rilis baru. AMI ini memiliki

semua komponen yang diperlukan untuk AWS ParallelCluster berfungsi saat diinstal dan dikonfigurasi. Anda bisa mulai dengan salah satu dari ini sebagai basis Anda.

Poin kunci:

- Metode ini lebih cepat dari [build-image](#) perintah. Namun, ini adalah proses manual dan tidak dapat diulang secara otomatis.
- Dengan metode ini, Anda tidak memiliki akses ke pengambilan log dan perintah manajemen siklus hidup gambar yang tersedia melalui CLI.

Langkah:

New EC2 console

1. Temukan AMI yang sesuai dengan yang spesifik Wilayah AWS yang Anda gunakan. Untuk menemukannya, gunakan [pcluster list-official-images](#) perintah dengan `--region` parameter untuk memilih spesifik Wilayah AWS dan `--os` dan `--architecture` parameter untuk memfilter AMI yang diinginkan dengan OS dan arsitektur yang ingin Anda gunakan. Dari output, ambil ID Gambar EC2.
2. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
3. Di panel navigasi, pilih Gambar, dan kemudian AMI. Cari ID Gambar EC2 yang diambil, pilih AMI, dan pilih Luncurkan instance dari AMI.
4. Gulir ke bawah dan pilih Jenis instans.
5. Pilih Anda Pasangan kunci dan Luncurkan Instans.
6. Masuk ke instans Anda menggunakan pengguna OS dan SSH kunci.
7. Sesuaikan instans Anda secara manual untuk memenuhi kebutuhan Anda.
8. Jalankan perintah berikut untuk mempersiapkan instance Anda untuk pembuatan AMI.

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Dari konsol, pilih Keadaan instans dan Hentikan contoh.

Arahkan ke Contoh, pilih contoh baru, pilih Keadaan instans, dan Hentikan contoh.

10. Buat AMI baru dari instans menggunakan konsol EC2 atau AWS CLI [buat-gambar](#).

Dari konsol EC2

- a. Pilih Instans di panel navigasi.
- b. Pilih contoh yang Anda buat dan modifikasi.
- c. DiTindakan, pilihGambardan kemudianBuat gambar.
- d. Pilih Buat Gambar.

11Masukkan ID AMI baru di [CustomAmi](#) bidang dalam konfigurasi cluster Anda dan buat cluster.

Old EC2 console

1. TemukanAWS ParallelClusterAMI yang sesuai dengan yang spesifikWilayah AWSyang Anda gunakan. Untuk menemukannya, Anda dapat menggunakan [pcluster list-official-images](#) perintah dengan `--region` parameter untuk memilih spesifikWilayah AWSdan `--os` dan `--architecture` parameter untuk memfilter AMI yang diinginkan dengan OS dan arsitektur yang ingin Anda gunakan. Dari output Anda dapat mengambil ID Gambar EC2.
2. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
3. Di panel navigasi, pilihGambar, dan kemudianAMI. Atur filter untukGambar publikdan cari ID Gambar EC2 yang diambil, pilih AMI, dan pilihPeluncuran.
4. Pilih jenis instans Anda dan pilihBerikutnya: Konfigurasikan Detail InstanceatauTinjau dan Luncurkanuntuk meluncurkan instance Anda.
5. PilihPeluncuran, pilihPasangan kunci, danPeluncuran Instans.
6. Masuk ke instans Anda menggunakan pengguna OS dan kunci SSH Anda. Untuk informasi lebih lanjut, navigasikan keContoh, pilih contoh baru danHubungkan.
7. Sesuaikan instans Anda secara manual untuk memenuhi kebutuhan Anda.
8. Jalankan perintah berikut untuk menyiapkan instance Anda untuk pembuatan AMI:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Dari konsol EC2, pilihContohdi panel navigasi, pilih instance baru Anda dan pilihTindakan,Negara InstancedanBerhenti.
- 10Buat AMI baru dari instans menggunakan konsol EC2 atauAWS CLI [buat-gambar](#).

Dari konsol EC2

- a. Pilih Instans di panel navigasi.
- b. Pilih instance yang Anda buat dan modifikasi.
- c. DiTindakan, pilih Gambardan kemudian Buat Gambar.
- d. Pilih Buat Gambar.

11 Masukkan ID AMI baru di [CustomAmi](#) bidang dalam konfigurasi cluster Anda dan buat cluster.

## Mengintegrasikan Active Directory

Dalam tutorial ini, Anda membuat lingkungan beberapa pengguna. Lingkungan ini mencakup AWS ParallelCluster yang terintegrasi dengan AWS Managed Microsoft AD (Active Directory) `dicorp.example.com`. Anda mengonfigurasi Admin pengguna untuk mengelola direktori, ReadOnly pengguna untuk membaca direktori, dan `user000` pengguna untuk masuk ke cluster. Anda dapat menggunakan jalur otomatis atau jalur manual untuk membuat sumber daya jaringan, Active Directory (AD), dan instans EC2 yang Anda gunakan untuk mengonfigurasi AD. Terlepas dari jalurnya, infrastruktur yang Anda buat sudah dikonfigurasi sebelumnya untuk diintegrasikan AWS ParallelCluster menggunakan salah satu metode berikut:

- LDAPS dengan verifikasi sertifikat (direkomendasikan sebagai opsi paling aman)
- LDAPS tanpa verifikasi sertifikat
- LDAP

LDAP dengan sendirinya tidak menyediakan enkripsi. Untuk memastikan transmisi informasi yang berpotensi sensitif dengan aman, kami sangat menyarankan Anda menggunakan LDAPS (LDAP over TLS/SSL) untuk klaster yang terintegrasi dengan AD. Untuk informasi selengkapnya, lihat [Mengaktifkan LDAPS sisi server menggunakan AWS Managed Microsoft AD](#) dalam Panduan Administrasi. AWS Directory Service

Setelah Anda membuat sumber daya ini, lanjutkan untuk mengkonfigurasi dan membuat klaster Anda terintegrasi dengan Active Directory (AD) Anda. Setelah cluster dibuat, masuk sebagai pengguna yang Anda buat. Untuk informasi lebih lanjut tentang konfigurasi yang Anda buat dalam tutorial ini, lihat [Beberapa akses pengguna ke cluster](#) dan bagian [DirectoryService](#) konfigurasi.


Tutorial ini mencakup cara membuat lingkungan yang mendukung beberapa akses pengguna ke cluster. Tutorial ini tidak mencakup cara Anda membuat dan menggunakan AWS Directory



Service AD. Langkah-langkah yang Anda ambil untuk mengatur AWS Managed Microsoft AD dalam tutorial ini disediakan untuk tujuan pengujian saja. Mereka tidak disediakan untuk menggantikan dokumentasi resmi dan praktik terbaik yang dapat Anda temukan di [AWS Managed Microsoft AD](#) dan [Simple AD](#) di Panduan AWS Directory Service Administrasi.

 Note

Kata sandi pengguna direktori kedaluwarsa sesuai dengan definisi properti kebijakan kata sandi direktori. Untuk informasi selengkapnya lihat [Setelan kebijakan yang didukung](#). Untuk mengatur ulang kata sandi direktori dengan AWS ParallelCluster, lihat [Cara mengatur ulang kata sandi pengguna dan kata sandi yang kedaluwarsa](#).

 Note

Alamat IP pengontrol domain direktori dapat berubah karena perubahan pengontrol domain dan pemeliharaan direktori. Jika Anda memilih metode pembuatan cepat otomatis untuk membuat infrastruktur direktori, Anda harus menyelaraskan penyeimbang beban secara manual di depan pengontrol direktori saat alamat IP direktori berubah. Jika Anda menggunakan metode quick create, alamat IP direktori tidak secara otomatis selaras dengan load balancer.

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Layanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Biaya UI](#).

### Prasyarat

- AWS ParallelCluster [diinstal](#).
- AWS CLI [itu diinstal dan dikonfigurasi](#).
- Anda memiliki [key pair EC2](#).

- Anda memiliki peran IAM dengan [izin](#) yang diperlukan untuk menjalankan CLI [pcluster](#).

Saat Anda membaca tutorial, ganti *inputs highlighted in red*, seperti *region-id* dan *abcdef01234567890*, dengan nama dan ID Anda sendiri. Ganti *0123456789012* dengan Akun AWS nomor Anda.

## Langkah 1: Buat infrastruktur AD

Pilih tab Otomatis untuk membuat infrastruktur Direktori Aktif (AD) dengan templat buat AWS CloudFormation cepat.

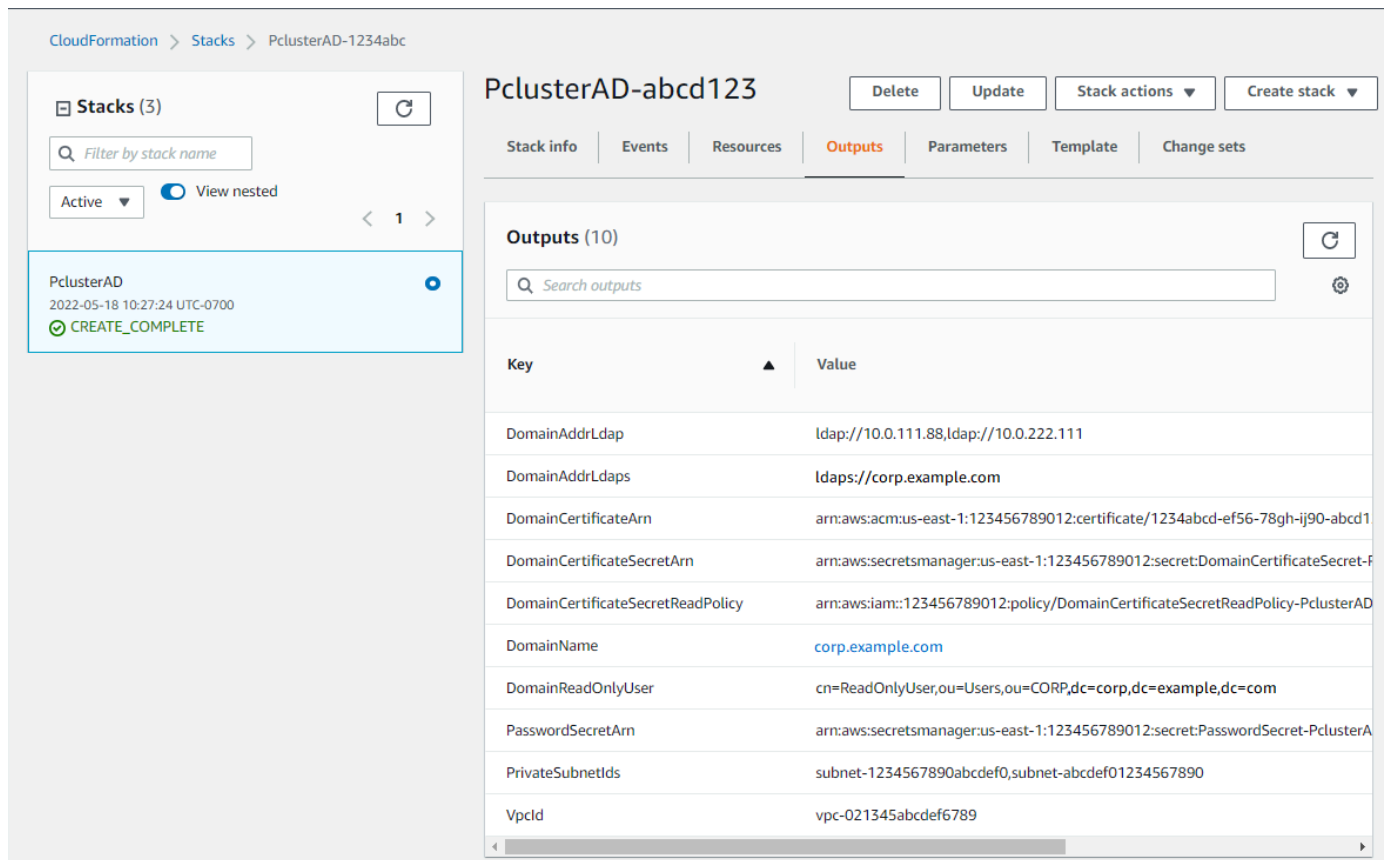
Pilih tab Manual untuk membuat infrastruktur AD secara manual.

### Otomatis

1. Masuk ke AWS Management Console.
2. Buka [CloudFormation Quick Create \(region us-east-1\)](#) untuk membuat sumber daya berikut di konsol: CloudFormation
  - VPC dengan dua subnet dan routing untuk akses publik, jika tidak ada VPC yang ditentukan.
  - Sesi AWS Managed Microsoft AD.
  - Instans EC2 yang bergabung dengan AD yang dapat Anda gunakan untuk mengelola direktori.
3. Di bagian Quick create stack page Parameters, masukkan kata sandi untuk parameter berikut:
  - AdminPassword
  - ReadOnlyPassword
  - UserPassword

Catat kata sandi. Anda menggunakannya nanti dalam tutorial ini.

4. Untuk DomainName, masukkan **corp.example.com**
5. Untuk Keypair, masukkan nama key pair EC2.
6. Centang kotak untuk mengetahui setiap kemampuan akses di bagian bawah halaman.
7. Pilih Create stack.
8. Setelah CloudFormation tumpukan mencapai CREATE\_COMPLETE status, pilih tab Output dari tumpukan. Catat nama sumber daya keluaran dan ID karena Anda perlu menggunakannya di langkah selanjutnya. Output memberikan informasi yang diperlukan untuk membuat cluster.



CloudFormation > Stacks > PclusterAD-1234abc

Stacks (3)     View nested < 1 >

PclusterAD  
2022-05-18 10:27:24 UTC-0700  
CREATE\_COMPLETE

### PclusterAD-abcd123

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

#### Outputs (10)

| Key                               | Value                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------|
| DomainAddrLdap                    | ldap://10.0.111.88,ldap://10.0.222.111                                         |
| DomainAddrLdaps                   | ldaps://corp.example.com                                                       |
| DomainCertificateArn              | arn:aws:acm:us-east-1:123456789012:certificate/1234abcd-ef56-78gh-ij90-abcd1   |
| DomainCertificateSecretArn        | arn:aws:secretsmanager:us-east-1:123456789012:secret:DomainCertificateSecret-F |
| DomainCertificateSecretReadPolicy | arn:aws:iam::123456789012:policy/DomainCertificateSecretReadPolicy-PclusterAD  |
| DomainName                        | corp.example.com                                                               |
| DomainReadOnlyUser                | cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com                     |
| PasswordSecretArn                 | arn:aws:secretsmanager:us-east-1:123456789012:secret>PasswordSecret-PclusterA  |
| PrivateSubnetIds                  | subnet-1234567890abcdef0,subnet-abcdef01234567890                              |
| VpcId                             | vpc-021345abcdef6789                                                           |

9. Untuk menyelesaikan latihan [\(Opsional\) Langkah 2: Kelola pengguna dan grup AD](#), Anda memerlukan ID direktori. Pilih Sumber Daya dan gulir ke bawah untuk mencatat ID direktori.
10. Lanjutkan di [\(Opsional\) Langkah 2: Kelola pengguna dan grup AD](#) atau [Langkah 3: Buat cluster](#).

## Manual

Buat VPC untuk layanan direktori dengan dua subnet di Availability Zone yang berbeda dan file. AWS Managed Microsoft AD

## Buat AD

### Note

- Direktori dan nama domain adalah `corp.example.com`. Nama pendeknya adalah `CORP`.
- Ubah Admin kata sandi dalam skrip.
- Active Directory (AD) membutuhkan waktu setidaknya 15 menit untuk membuatnya.

Gunakan skrip Python berikut untuk membuat VPC, subnet, dan sumber daya AD di lokal Anda. Wilayah AWS Simpan file ini sebagai `ad.py` dan jalankan.

```
import boto3
import time
from pprint import pprint

vpc_name = "PclusterVPC"
ad_domain = "corp.example.com"
admin_password = "asdfASDF1234"

ec2 = boto3.client("ec2")
ds = boto3.client("ds")
region = boto3.Session().region_name

Create the VPC, Subnets, IGW, Routes
vpc = ec2.create_vpc(CidrBlock="10.0.0.0/16")["Vpc"]
vpc_id = vpc["VpcId"]
time.sleep(30)
ec2.create_tags(Resources=[vpc_id], Tags=[{"Key": "Name", "Value": vpc_name}])
subnet1 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.0.0/17",
 AvailabilityZone=f"{region}a")["Subnet"]
subnet1_id = subnet1["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet1_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/subnet1"}])
ec2.modify_subnet_attribute(SubnetId=subnet1_id, MapPublicIpOnLaunch={"Value": True})
subnet2 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.128.0/17",
 AvailabilityZone=f"{region}b")["Subnet"]
subnet2_id = subnet2["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet2_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/subnet2"}])
ec2.modify_subnet_attribute(SubnetId=subnet2_id, MapPublicIpOnLaunch={"Value": True})
igw = ec2.create_internet_gateway()["InternetGateway"]
ec2.attach_internet_gateway(InternetGatewayId=igw["InternetGatewayId"], VpcId=vpc_id)
route_table = ec2.describe_route_tables(Filters=[{"Name": "vpc-id", "Values":
 [vpc_id]}])["RouteTables"][0]
ec2.create_route(RouteTableId=route_table["RouteTableId"],
 DestinationCidrBlock="0.0.0.0/0", GatewayId=igw["InternetGatewayId"])
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsSupport={"Value": True})
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsHostnames={"Value": True})
```

```
Create the Active Directory
ad = ds.create_microsoft_ad(
 Name=ad_domain,
 Password=admin_password,
 Description="ParallelCluster AD",
 VpcSettings={"VpcId": vpc_id, "SubnetIds": [subnet1_id, subnet2_id]},
 Edition="Standard",
)
directory_id = ad["DirectoryId"]

Wait for completion
print("Waiting for the directory to be created...")
directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
directory = directories[0]
while directory["Stage"] in {"Requested", "Creating"}:
 time.sleep(3)
 directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
 directory = directories[0]

dns_ip_addrs = directory["DnsIpAddrs"]

pprint({"directory_id": directory_id,
 "vpc_id": vpc_id,
 "subnet1_id": subnet1_id,
 "subnet2_id": subnet2_id,
 "dns_ip_addrs": dns_ip_addrs})
```

Berikut ini adalah contoh output dari script Python.

```
{
 "directory_id": "d-abcdef01234567890",
 "dns_ip_addrs": ["192.0.2.254", "203.0.113.237"],
 "subnet1_id": "subnet-021345abcdef6789",
 "subnet2_id": "subnet-1234567890abcdef0",
 "vpc_id": "vpc-021345abcdef6789"
}
```

Catat nama sumber daya keluaran dan ID. Anda menggunakannya di langkah selanjutnya.

Setelah skrip selesai, lanjutkan ke langkah berikutnya.

## Buat instans EC2

### New EC2 console

1. Masuk ke AWS Management Console.
2. Jika Anda tidak memiliki peran dengan kebijakan yang tercantum di langkah 4 terlampir, buka konsol IAM di <https://console.aws.amazon.com/iam/>. Jika tidak, lewati ke langkah 5.
3. Buat `ResetUserPassword` kebijakan, ganti konten berwarna merah yang disorot dengan ID, ID Akun, dan ID direktori dari output skrip yang Anda jalankan untuk membuat AD. Wilayah AWS

### ResetUserPassword

```
{
 "Statement": [
 {
 "Action": [
 "ds:ResetUserPassword"
],
 "Resource": "arn:aws:ds:region-id:123456789012:directory/d-abcdef01234567890",
 "Effect": "Allow"
 }
]
}
```

4. Buat peran IAM dengan kebijakan berikut terlampir.
  - AWSkebijakan terkelola [AmazonSSM ManagedInstanceCore](#)
  - AWSkebijakan terkelola [AmazonSSM DirectoryServiceAccess](#)
  - `ResetUserPassword` kebijakan
5. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
6. Di Dasbor EC2, pilih Launch Instance.
7. Di Application and OS Images, pilih Amazon Linux 2 AMI terbaru.
8. Untuk jenis Instance, pilih t2.micro.
9. Untuk Key pair, pilih key pair.
10. Untuk pengaturan Jaringan, pilih Edit.
11. Untuk VPC, pilih direktori VPC.

12. Gulir ke bawah dan pilih Detail lanjutan.
13. Di Detail lanjutan, Domain bergabung direktori, pilih **corp.example.com**.
14. Untuk profil Instans IAM, pilih peran yang Anda buat di langkah 1 atau peran dengan kebijakan yang tercantum di langkah 4 terlampir.
15. Dalam Ringkasan pilih Launch instance.
16. Catat ID Instance (misalnya, i-1234567890abcdef0) dan tunggu hingga instance selesai diluncurkan.
17. Setelah instance diluncurkan, lanjutkan ke langkah berikutnya.

## Old EC2 console

1. Masuk ke AWS Management Console.
2. Jika Anda tidak memiliki peran dengan kebijakan yang tercantum di langkah 4 terlampir, buka konsol IAM di <https://console.aws.amazon.com/iam/>. Jika tidak, lewati ke langkah 5.
3. Buat `ResetUserPassword` kebijakan. Ganti konten yang disorot merah dengan Wilayah AWS Akun AWS ID, ID, dan ID direktori Anda dari output skrip yang Anda jalankan untuk membuat Active Directory (AD).

### ResetUserPassword

```
{
 "Statement": [
 {
 "Action": [
 "ds:ResetUserPassword"
],
 "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
 "Effect": "Allow"
 }
]
}
```

4. Buat peran IAM dengan kebijakan berikut terlampir.
  - AWSkebijakan terkelola [AmazonSSM ManagedInstanceCore](#)
  - AWSkebijakan terkelola [AmazonSSM DirectoryServiceAccess](#)
  - `ResetUserPassword`kebijakan

5. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
6. Di Dasbor EC2, pilih Launch Instance.
7. Di Application and OS Images, pilih Amazon Linux 2 AMI terbaru.
8. Untuk jenis Instance, pilih t2.micro.
9. Untuk Key pair, pilih key pair.
10. Di Pengaturan jaringan, pilih Edit.
11. Di Pengaturan jaringan, VPC, pilih direktori VPC.
12. Gulir ke bawah dan pilih Detail lanjutan.
13. Di Detail lanjutan, Domain bergabung direktori, pilih **corp.example.com**.
14. Di Detail lanjutan, Profil instans, pilih peran yang Anda buat di langkah 1 atau peran dengan kebijakan yang tercantum di langkah 4 terlampir.
15. Dalam Ringkasan pilih Launch instance.
16. Catat ID Instance (misalnya, i-1234567890abcdef0) dan tunggu instance selesai diluncurkan.
17. Setelah instance diluncurkan, lanjutkan ke langkah berikutnya.

Bergabunglah dengan instans Anda ke AD

1. Hubungkan ke instans Anda dan bergabunglah dengan ranah AD sebagai **admin**.

Jalankan perintah berikut untuk terhubung ke instance.

```
$ INSTANCE_ID="i-1234567890abcdef0"
```

```
$ PUBLIC_IP=$(aws ec2 describe-instances \
--instance-ids $INSTANCE_ID \
--query "Reservations[0].Instances[0].PublicIpAddress" \
--output text)
```

```
$ ssh -i ~/.ssh/keys/keypair.pem ec2-user@$PUBLIC_IP
```

2. Instal perangkat lunak yang diperlukan dan bergabunglah dengan dunia.

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-
common-tools krb5-workstation openldap-clients polycoreutils-python
```



### 3. Ganti kata sandi admin dengan **admin** kata sandi Anda.

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com
Password for Admin:
```

Jika sebelumnya telah berhasil, Anda bergabung ke ranah dan dapat melanjutkan ke langkah berikutnya.

### Menambahkan pengguna ke AD

#### 1. Buat ReadOnlyUser dan pengguna tambahan.

Pada langkah ini, Anda menggunakan alat [adcli](#) dan [openldap-client](#) yang Anda instal pada langkah sebelumnya.

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=ReadOnlyUser ReadOnlyUser
```

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=user000 user000
```

#### 2. Verifikasi pengguna dibuat:

Direktori alamat IP DNS adalah output dari skrip Python.

```
$ DIRECTORY_IP="192.0.2.254"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=user000,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

Secara default, saat Anda membuat pengguna dengan `adcli`, pengguna dinonaktifkan.

### 3. Setel ulang dan aktifkan kata sandi pengguna dari mesin lokal Anda:

Keluar dari instans EC2 Anda.

#### Note

- `ro-p@ssw0rd` adalah kata sandi dari `ReadOnlyUser`, diambil dari AWS Secrets Manager.
- `user-p@ssw0rd` adalah kata sandi pengguna cluster yang disediakan saat Anda menghubungkan (ssh) ke cluster.

`directory-id` ini adalah output dari skrip Python.

```
$ DIRECTORY_ID="d-abcdef01234567890"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "ReadOnlyUser" \
--new-password "ro-p@ssw0rd" \
--region "region-id"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "user000" \
--new-password "user-p@ssw0rd" \
--region "region-id"
```

### 4. Tambahkan kata sandi ke rahasia Secrets Manager.

Sekarang setelah Anda membuat `ReadOnlyUser` dan mengatur kata sandi, simpan dalam rahasia yang AWS ParallelCluster digunakan untuk memvalidasi login.

Gunakan Secrets Manager untuk membuat rahasia baru untuk menyimpan kata sandi `ReadOnlyUser` sebagai nilai. Format nilai rahasia harus berupa teks biasa saja (bukan format JSON). Catat rahasia ARN untuk langkah-langkah masa depan.

```
$ aws secretsmanager create-secret --name "ADSecretPassword" \
--region region-id \
```

```
--secret-string "ro-p@ssw0rd" \
--query ARN \
--output text
arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
```

LDAPs dengan pengaturan verifikasi sertifikat (disarankan)

Buat catatan ID sumber daya. Anda menggunakannya dalam langkah-langkah nanti.

1. Hasilkan sertifikat domain, secara lokal.

```
$ PRIVATE_KEY="corp-example-com.key"
CERTIFICATE="corp-example-com.crt"
printf ".\n.\n.\n.\n.\ncorp.example.com\n.\n" | openssl req -x509 -sha256 -nodes -
newkey rsa:2048 -keyout $PRIVATE_KEY -days 365 -out $CERTIFICATE
```

2. Simpan sertifikat ke Secrets Manager untuk membuatnya dapat diambil dari dalam cluster nanti.

```
$ aws secretsmanager create-secret --name example-cert \
--secret-string file://$CERTIFICATE \
--region region-id
{
 "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
 "Name": "example-cert",
 "VersionId": "14866070-092a-4d5a-bcdd-9219d0566b9c"
}
```

3. Tambahkan kebijakan berikut ke peran IAM yang Anda buat untuk menggabungkan instans EC2 ke domain AD.

PutDomainCertificateSecrets

```
{
 "Statement": [
 {
 "Action": [
 "secretsmanager:PutSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
```

```

],
 "Effect": "Allow"
 }
]
}

```

#### 4. Impor sertifikat ke AWS Certificate Manager (ACM).

```

$ aws acm import-certificate --certificate fileb://$CERTIFICATE \
 --private-key fileb://$PRIVATE_KEY \
 --region region-id
{
 "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
}

```

#### 5. Buat dan penyeimbang beban yang diletakkan di depan titik akhir Active Directory.

```

$ aws elbv2 create-load-balancer --name CorpExampleCom-NLB \
 --type network \
 --scheme internal \
 --subnets subnet-1234567890abcdef0 subnet-021345abcdef6789 \
 --region region-id
{
 "LoadBalancers": [
 {
 "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
 "CanonicalHostedZoneId": "Z2IFOLAFXWL04F",
 "CreatedTime": "2022-05-05T12:56:55.988000+00:00",
 "LoadBalancerName": "CorpExampleCom-NLB",
 "Scheme": "internal",
 "VpcId": "vpc-021345abcdef6789",
 "State": {
 "Code": "provisioning"
 },
 "Type": "network",
 "AvailabilityZones": [
 {
 "ZoneName": "region-idb",
 "SubnetId": "subnet-021345abcdef6789",
 "LoadBalancerAddresses": []
 }
]
 }
]
}

```

```

 },
 {
 "ZoneName": "region-ida",
 "SubnetId": "subnet-1234567890abcdef0",
 "LoadBalancerAddresses": []
 }
],
 "IpAddressType": "ipv4"
}
]
}

```

6. Buat grup target yang menargetkan titik akhir Active Directory.

```

$ aws elbv2 create-target-group --name CorpExampleCom-Targets --protocol TCP \
 --port 389 \
 --target-type ip \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id
{
 "TargetGroups": [
 {
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
 "TargetGroupName": "CorpExampleCom-Targets",
 "Protocol": "TCP",
 "Port": 389,
 "VpcId": "vpc-021345abcdef6789",
 "HealthCheckProtocol": "TCP",
 "HealthCheckPort": "traffic-port",
 "HealthCheckEnabled": true,
 "HealthCheckIntervalSeconds": 30,
 "HealthCheckTimeoutSeconds": 10,
 "HealthyThresholdCount": 3,
 "UnhealthyThresholdCount": 3,
 "TargetType": "ip",
 "IpAddressType": "ipv4"
 }
]
}

```

## 7. Daftarkan titik akhir Active Directory (AD) ke dalam grup target.

```
$ aws elbv2 register-targets --target-group-arn
arn:aws:elasticloadbalancing:region-id:123456789012:targetgroup/CorpExampleCom-
Targets/44577c583b695e81 \
--targets Id=192.0.2.254,Port=389 Id=203.0.113.237,Port=389 \
--region region-id
```

## 8. Buat pendengar LB dengan sertifikat.

```
$ aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region-id:123456789012:loadbalancer/net/
CorpExampleCom-NLB/3afe296bf4ba80d4 \
--protocol TLS \
--port 636 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 \
--ssl-policy ELBSecurityPolicy-TLS-1-2-2017-01 \
--certificates CertificateArn=arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 \
--region region-id
"Listeners": [
{
 "ListenerArn": "arn:aws:elasticloadbalancing:region-id:123456789012:listener/
net/CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b",
 "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
 "Port": 636,
 "Protocol": "TLS",
 "Certificates": [
 {
 "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
 }
],
 "SslPolicy": "ELBSecurityPolicy-TLS-1-2-2017-01",
 "DefaultActions": [
 {
 "Type": "forward",
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
 "ForwardConfig": {
```

```

 "TargetGroups": [
 {
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81"
 }
]
 }
}
]
}
]
}
}

```

9. Buat zona yang dihosting untuk membuat domain dapat ditemukan di dalam VPC cluster.

```

$ aws route53 create-hosted-zone --name corp.example.com \
 --vpc VPCRegion=region-id,VPCId=vpc-021345abcdef6789 \
 --caller-reference "ParallelCluster AD Tutorial"
{
 "Location": "https://route53.amazonaws.com/2013-04-01/hostedzone/
Z09020002B5MZQNXMSJUB",
 "HostedZone": {
 "Id": "/hostedzone/Z09020002B5MZQNXMSJUB",
 "Name": "corp.example.com.",
 "CallerReference": "ParallelCluster AD Tutorial",
 "Config": {
 "PrivateZone": true
 },
 "ResourceRecordSetCount": 2
 },
 "ChangeInfo": {
 "Id": "/change/C05533343BF3IKSORW1TQ",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T13:21:53.863000+00:00"
 },
 "VPC": {
 "VPCRegion": "region-id",
 "VPCId": "vpc-021345abcdef6789"
 }
}

```

10. Buat file yang diberi nama **recordset-change.json** dengan konten berikut.  
**HostedZoneId** adalah ID zona host kanonik dari penyeimbang beban.

```
{
 "Changes": [
 {
 "Action": "CREATE",
 "ResourceRecordSet": {
 "Name": "corp.example.com",
 "Type": "A",
 "Region": "region-id",
 "SetIdentifier": "example-active-directory",
 "AliasTarget": {
 "HostedZoneId": "Z2IF0LAFXWL04F",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
 "EvaluateTargetHealth": true
 }
 }
 }
]
}
```

11. Kirim perubahan recordset ke zona yang dihosting, kali ini menggunakan ID zona yang dihosting.

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXMSJUB \
--change-batch file://recordset-change.json
{
 "ChangeInfo": {
 "Id": "/change/C0137926I56R3GC7XW2Y",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
 }
}
```

12. Buat dokumen kebijakan **policy.json** dengan konten berikut.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "secretsmanager:GetSecretValue"
]
 }
]
}
```



```
],
 "Resource": [
 "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-abc123"
],
 "Effect": "Allow"
 }
]
```

13. Buat dokumen kebijakan yang diberi nama **policy.json** dengan konten berikut.

```
$ aws iam create-policy --policy-name ReadCertExample \
 --policy-document file://policy.json
{
 "Policy": {
 "PolicyName": "ReadCertExample",
 "PolicyId": "ANPAUUXUVBC42VZSI4LDY",
 "Arn": "arn:aws:iam::123456789012:policy/ReadCertExample-efg456",
 "Path": "/",
 "DefaultVersionId": "v1",
 "AttachmentCount": 0,
 "PermissionsBoundaryUsageCount": 0,
 "IsAttachable": true,
 "CreateDate": "2022-05-05T13:42:18+00:00",
 "UpdateDate": "2022-05-05T13:42:18+00:00"
 }
}
```

14. Terus ikuti langkah-langkah di [\(Opsional\) Langkah 2: Kelola pengguna dan grup AD](#) atau [Langkah 3: Buat cluster](#).

## (Opsional) Langkah 2: Kelola pengguna dan grup AD

Pada langkah ini, Anda mengelola pengguna dan grup dari instans EC2 Amazon Linux 2 yang bergabung dengan domain Pengiriman Aktif (AD).

Jika Anda mengikuti jalur otomatis, mulai ulang dan masuk ke instans gabungan AD yang dibuat sebagai bagian dari otomatisasi.

Jika Anda mengikuti jalur manual, mulai ulang dan masuk ke instans yang Anda buat dan bergabung dengan AD di langkah-langkah sebelumnya.

Dalam langkah-langkah ini, Anda menggunakan alat [adcli](#) dan [openldap-client](#) yang diinstal dalam instance sebagai bagian dari langkah sebelumnya.

Masuk ke instans EC2 yang bergabung dengan domain AD

1. Dari konsol EC2, pilih instans EC2 tanpa judul yang dibuat pada langkah sebelumnya. Status instance mungkin Dihentikan.
2. Jika status instance Dihentikan, pilih status Instance dan kemudian Mulai instance.
3. Setelah pemeriksaan status lulus, pilih instance dan pilih Connect dan SSH masuk ke instance.

Kelola pengguna dan grup saat masuk ke instans EC2 Amazon Linux 2 yang bergabung dengan AD

Ketika Anda menjalankan `adcli` perintah dengan `-U "Admin"` opsi, Anda diminta untuk memasukkan Admin kata sandi AD. Anda menyertakan Admin kata sandi AD sebagai bagian dari `ldapsearch` perintah.

1. Buat pengguna.

```
$ adcli create-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

2. Tetapkan kata sandi pengguna.

```
$ aws --region "region-id" ds reset-user-password --directory-id "d-
abcdef01234567890" --user-name "clusteruser" --new-password "new-p@ssw0rd"
```

3. Buat grup.

```
$ adcli create-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

4. Tambahkan pengguna ke grup.

```
$ adcli add-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

5. Jelaskan pengguna dan grup.

Jelaskan semua pengguna.

```
$ ldapsearch "(&(objectClass=user))" -x -h "192.0.2.254" -b
"DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Jelaskan pengguna tertentu.

```
$ ldapsearch "(&(objectClass=user)(cn=clusteruser))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Jelaskan semua pengguna dengan pola nama.

```
$ ldapsearch "(&(objectClass=user)(cn=user*))" -x -h "192.0.2.254" -b
"DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Jelaskan semua pengguna yang merupakan bagian dari grup tertentu.

```
$ ldapsearch "(&(objectClass=user)
(memberOf=CN=clusterteam,OU=Users,OU=CORP,DC=corp,DC=example,DC=com))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Jelaskan semua kelompok

```
$ ldapsearch "objectClass=group" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com"
-D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Jelaskan kelompok tertentu

```
$ ldapsearch "(&(objectClass=group)(cn=clusterteam))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

6. Hapus pengguna dari grup.

```
$ adcli remove-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

## 7. Hapus pengguna.

```
$ adcli delete-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

## 8. Hapus grup.

```
$ adcli delete-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

## Langkah 3: Buat cluster

Jika Anda belum keluar dari instans EC2, lakukan sekarang.

Lingkungan diatur untuk membuat cluster yang dapat mengautentikasi pengguna terhadap Active Directory (AD).

Buat konfigurasi cluster sederhana dan berikan pengaturan yang relevan untuk menghubungkan ke AD. Untuk informasi selengkapnya, lihat bagian [DirectoryService](#).

Pilih salah satu konfigurasi klaster berikut dan salin ke file yang diberi nama `ldaps_config.yaml`, `ldaps_nocert_config.yaml`, atau `ldap_config.yaml`.

Kami menyarankan Anda memilih konfigurasi LDAPS dengan verifikasi sertifikat. Jika Anda memilih konfigurasi ini, Anda juga harus menyalin skrip bootstrap ke file yang diberi nama `active-directory.head.post.sh`. Dan, Anda harus menyimpannya di bucket Amazon S3 seperti yang ditunjukkan dalam file konfigurasi.

LDAPS dengan konfigurasi verifikasi sertifikat (disarankan)

### Note

Komponen-komponen berikut harus diubah.

- `KeyName`: Salah satu keypair EC2 Anda.
- `SubnetId` / `SubnetIds`: Salah satu ID subnet yang disediakan dalam output dari CloudFormation quick create stack (tutorial otomatis) atau skrip python (tutorial manual).
- `Region`: Wilayah tempat Anda membuat infrastruktur AD.
- `DomainAddr`: Alamat IP ini adalah salah satu alamat DNS dari layanan AD Anda.

- `PasswordSecretArn`: Nama Sumber Daya Amazon (ARN) dari rahasia yang berisi kata sandi untuk file. `DomainReadOnlyUser`
- `BucketName`: Nama bucket yang menyimpan skrip bootstrap.
- `AdditionalPolicies/Policy`: Nama Sumber Daya Amazon (ARN) dari kebijakan sertifikasi domain baca. `ReadCertExample`
- `CustomActions/OnNodeConfigured/Args`: Nama Sumber Daya Amazon (ARN) rahasia yang memegang kebijakan sertifikasi domain.

Untuk postur keamanan yang lebih baik, kami sarankan untuk menggunakan `AllowedIps` konfigurasi `HeadNodeSsh`//untuk membatasi akses SSH ke node kepala.

```

Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ReadCertExample
 S3Access:
 - BucketName: my-bucket
 EnableWriteAccess: false
 KeyName: bootstrap/active-directory/active-directory.head.post.sh
 CustomActions:
 OnNodeConfigured:
 Script: s3://my-bucket/bootstrap/active-directory/active-directory.head.post.sh
 Args:
 - arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc
 - /opt/parallelcluster/shared/directory_service/domain-certificate.crt
 Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro

```

```

 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: corp.example.com
 DomainAddr: ldaps://corp.example.com
 PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 LdapTlsCaCert: /opt/parallelcluster/shared/directory_service/domain-certificate.crt
 LdapTlsReqCert: hard

```

## Skrip Bootstrap

Setelah Anda membuat file bootstrap dan sebelum Anda mengunggahnya ke bucket S3 Anda, jalankan `chmod +x active-directory.head.post.sh` untuk memberikan izin AWS ParallelCluster jalankan.

```

#!/bin/bash
set -e

CERTIFICATE_SECRET_ARN="$1"
CERTIFICATE_PATH="$2"

[[-z $CERTIFICATE_SECRET_ARN]] && echo "[ERROR] Missing CERTIFICATE_SECRET_ARN" &&
exit 1
[[-z $CERTIFICATE_PATH]] && echo "[ERROR] Missing CERTIFICATE_PATH" && exit 1

source /etc/parallelcluster/cfnconfig
REGION="${cfn_region:?}"

mkdir -p $(dirname $CERTIFICATE_PATH)
aws secretsmanager get-secret-value --region $REGION --secret-id
$CERTIFICATE_SECRET_ARN --query SecretString --output text > $CERTIFICATE_PATH

```

## LDAPS tanpa konfigurasi verifikasi sertifikat

### Note

Komponen-komponen berikut harus diubah.

- `KeyName`: Salah satu keypair EC2 Anda.
- `SubnetId` / `SubnetIds`: Salah satu ID subnet yang ada di output dari CloudFormation quick create stack (tutorial otomatis) atau skrip python (tutorial manual).
- `Region`: Wilayah tempat Anda membuat infrastruktur AD.
- `DomainAddr`: Alamat IP ini adalah salah satu alamat DNS dari layanan AD Anda.
- `PasswordSecretArn`: Nama Sumber Daya Amazon (ARN) dari rahasia yang berisi kata sandi untuk file. `DomainReadOnlyUser`

Untuk postur keamanan yang lebih baik, kami sarankan untuk menggunakan `AllowedIps` konfigurasi `HeadNode /Ssh/` untuk membatasi akses SSH ke node kepala.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
```

```
DirectoryService:
 DomainName: corp.example.com
 DomainAddr: ldaps://corp.example.com
 PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 LdapTlsReqCert: never
```

## Konfigurasi LDAP

### Note

Komponen-komponen berikut harus diubah.

- `KeyName`: Salah satu keypair EC2 Anda.
- `SubnetId` / `SubnetIds`: Salah satu ID subnet yang disediakan dalam output dari CloudFormation quick create stack (tutorial otomatis) atau skrip python (tutorial manual).
- `Region`: Wilayah tempat Anda membuat infrastruktur AD.
- `DomainAddr`: Alamat IP ini adalah salah satu alamat DNS dari layanan AD Anda.
- `PasswordSecretArn`: Nama Sumber Daya Amazon (ARN) dari rahasia yang berisi kata sandi untuk file. `DomainReadOnlyUser`

Untuk postur keamanan yang lebih baik, kami sarankan untuk menggunakan `AllowedIps` konfigurasi `HeadNode /Ssh/` untuk membatasi akses SSH ke node kepala.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
```



```

ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: dc=corp,dc=example,dc=com
 DomainAddr: ldap://192.0.2.254,ldap://203.0.113.237
 PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 AdditionalSssdConfigs:
 ldap_auth_disable_tls_never_use_in_production: True

```

Buat kluster Anda dengan perintah berikut.

```

$ pcluster create-cluster --cluster-name "ad-cluster" --cluster-configuration "./ldaps_config.yaml"
{
 "cluster": {
 "clusterName": "pcluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "region-id",
 "version": 3.7.0,
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}

```

#### Langkah 4: Connect ke cluster sebagai pengguna

Anda dapat menentukan status cluster dengan perintah berikut.

```

$ pcluster describe-cluster -n ad-cluster --region "region-id" --query "clusterStatus"

```

Outputnya adalah sebagai berikut.

```

"CREATE_IN_PROGRESS" / "CREATE_COMPLETE"

```

Saat status mencapai "CREATE\_COMPLETE", masuk dengan nama pengguna dan kata sandi yang dibuat.

```
$ HEAD_NODE_IP=$(pcluster describe-cluster -n "ad-cluster" --region "region-id" --query headNode.publicIpAddress | xargs echo)
```

```
$ ssh user000@$HEAD_NODE_IP
```

Anda dapat masuk tanpa kata sandi dengan memberikan SSH kunci yang dibuat untuk pengguna baru di `/home/user000@HEAD_NODE_IP/.ssh/id_rsa`.

Jika ssh perintah berhasil, Anda telah berhasil terhubung ke cluster sebagai pengguna yang diautentikasi untuk menggunakan Active Directory (AD).

## Langkah 5: Bersihkan

1. Dari mesin lokal Anda, hapus klaster.

```
$ pcluster delete-cluster --cluster-name "ad-cluster" --region "region-id"
{
 "cluster": {
 "clusterName": "ad-cluster",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "region-id",
 "version": "3.7.0",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

2. Periksa kemajuan cluster yang dihapus.

```
$ pcluster describe-cluster --cluster-name "ad-cluster" --region "region-id" --query "clusterStatus"
"DELETE_IN_PROGRESS"
```

Setelah cluster berhasil dihapus, lanjutkan ke langkah berikutnya.

## Otomatis

### Hapus sumber daya Direktori Aktif

1. Dari <https://console.aws.amazon.com/cloudformation/>.
2. Di panel navigasi, pilih Stacks (Tumpukan).
3. Dari daftar tumpukan, pilih tumpukan AD (misalnya, pcluster-ad).
4. Pilih Hapus.

## Manual

1. Hapus instans EC2.
  - a. Dari <https://console.aws.amazon.com/ec2/>, pilih Instans di panel navigasi.
  - b. Dari daftar instance, pilih instance yang Anda buat untuk menambahkan pengguna ke direktori.
  - c. Pilih status Instance, lalu Hentikan instance.
2. Hapus zona yang dihosting.
  - a. Buat `recordset-delete.json` dengan konten berikut. Dalam contoh ini, `HostedZoneId` adalah ID zona host kanonik dari penyeimbang beban.

```
{
 "Changes": [
 {
 "Action": "DELETE",
 "ResourceRecordSet": {
 "Name": "corp.example.com",
 "Type": "A",
 "Region": "region-id",
 "SetIdentifier": "pcluster-active-directory",
 "AliasTarget": {
 "HostedZoneId": "Z2IFOLAFXWL04F",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
 "EvaluateTargetHealth": true
 }
 }
 }
]
}
```

```
}

```

- b. Kirim perubahan recordset ke zona yang dihosting menggunakan ID zona yang dihosting.

```
$ aws route53 change-resource-record-sets --hosted-zone-
id Z09020002B5MZQNXMSJUB \
 --change-batch file://recordset-delete.json
{
 "ChangeInfo": {
 "Id": "/change/C04853642A0TH2TJ5NLNI",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T14:25:51.046000+00:00"
 }
}
```

- c. Hapus zona yang dihosting.

```
$ aws route53 delete-hosted-zone --id Z09020002B5MZQNXMSJUB
{
 "ChangeInfo": {
 "Id": "/change/C0468051QFABTVHMDEG9",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T14:26:13.814000+00:00"
 }
}
```

3. Hapus pendengar LB.

```
$ aws elbv2 delete-listener \
 --listener-arn arn:aws:elasticloadbalancing:region-id:123456789012:listener/net/
CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b --region region-id
```

4. Hapus grup target.

```
$ aws elbv2 delete-target-group \
 --target-group-arn arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 --
region region-id
```

5. Hapus penyeimbang beban.

```
$ aws elbv2 delete-load-balancer \
```

```
--load-balancer-arn arn:aws:elasticloadbalancing:region-id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4 --
region region-id
```

6. Hapus kebijakan yang digunakan kluster untuk membaca sertifikat dari Secrets Manager.

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/
ReadCertExample
```

7. Hapus rahasia yang berisi sertifikat domain.

```
$ aws secretsmanager delete-secret \
 --secret-id arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc \
 --region region-id
{
 "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",
 "Name": "example-cert",
 "DeletionDate": "2022-06-04T16:27:36.183000+02:00"
}
```

8. Hapus sertifikat dari ACM.

```
$ aws acm delete-certificate \
 --certificate-arn arn:aws:acm:region-id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 --region region-id
```

9. Hapus sumber daya Active Directory (AD).

- a. Dapatkan ID sumber daya berikut dari output skrip `ad.py` python:

- ID IKLAN
- ID subnet AD
- ID VPC IKLAN

- b. Hapus direktori dengan menjalankan perintah berikut.

```
$ aws ds delete-directory --directory-id d-abcdef0123456789 --region region-id
{
 "DirectoryId": "d-abcdef0123456789"
}
```

- c. Buat daftar grup keamanan di VPC.

```
$ aws ec2 describe-security-groups --filters '[{"Name":"vpc-id","Values":["vpc-07614ade95ebad1bc"]}]' --region region-id
```

- d. Hapus grup keamanan khusus.

```
$ aws ec2 delete-security-group --group-id sg-021345abcdef6789 --region region-id
```

- e. Hapus subnet.

```
$ aws ec2 delete-subnet --subnet-id subnet-1234567890abcdef --region region-id
```

```
$ aws ec2 delete-subnet --subnet-id subnet-021345abcdef6789 --region region-id
```

- f. Jelaskan gateway internet.

```
$ aws ec2 describe-internet-gateways \
 --filters Name=attachment.vpc-id,Values=vpc-021345abcdef6789 \
 --region region-id
{
 "InternetGateways": [
 {
 "Attachments": [
 {
 "State": "available",
 "VpcId": "vpc-021345abcdef6789"
 }
],
 "InternetGatewayId": "igw-1234567890abcdef",
 "OwnerId": "123456789012",
 "Tags": []
 }
]
}
```

- g. Lepaskan gateway internet.

```
$ aws ec2 detach-internet-gateway \
 --internet-gateway-id igw-1234567890abcdef \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id
```

h. Hapus gateway internet.

```
$ aws ec2 delete-internet-gateway \
 --internet-gateway-id igw-1234567890abcdef \
 --region region-id
```

i. Hapus VPC.

```
$ aws ec2 delete-vpc \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id
```

j. Hapus rahasia yang berisi ReadOnlyUser kata sandi.

```
$ aws secretsmanager delete-secret \
 --secret-id arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234" \
 --region region-id
```

## Mengonfigurasi enkripsi penyimpanan bersama dengan kunci AWS KMS

Pelajari cara menyiapkan AWS KMS kunci yang dikelola pelanggan untuk mengenkripsi dan melindungi data Anda di sistem penyimpanan file kluster yang dikonfigurasi. AWS ParallelCluster

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Layanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelClusterBiaya UI](#).

AWS ParallelCluster mendukung opsi konfigurasi penyimpanan bersama berikut:

- [SharedStorage](#) / [EbsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [EfsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [FsxLustreSettings](#) / [KmsKeyId](#)

Anda dapat menggunakan opsi ini untuk menyediakan AWS KMS kunci yang dikelola pelanggan untuk enkripsi sistem penyimpanan bersama Amazon EBS, Amazon EFS, dan FSx for Lustre. Untuk menggunakannya, Anda harus membuat dan mengkonfigurasi kebijakan IAM untuk yang berikut:

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

## Prasyarat

- AWS ParallelCluster [dipasang](#).
- [diinstal dan dikonfigurasi. AWS CLI](#)
- Anda memiliki [pasangan kunci EC2](#).
- Anda memiliki peran IAM dengan [izin](#) yang diperlukan untuk menjalankan CLI. [pcluster](#)

## Topik

- [Buat kebijakan](#)
- [Mengkonfigurasi dan membuat cluster](#)

## Buat kebijakan

Buat kebijakan.

1. Pergi ke konsol IAM: <https://console.aws.amazon.com/iam/home>.
2. Pilih Policies (Kebijakan).
3. Pilih Buat kebijakan.
4. Pilih tab JSON dan tempel di kebijakan berikut. Pastikan untuk mengganti semua kejadian **123456789012** dengan Akun AWS ID Anda dan Amazon Resource Name (ARN) dan Wilayah AWS dengan nama Anda sendiri.

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



```

 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:region-id:123456789012:key/abcd1234-ef56-gh78-ij90-abcd1234efgh5678"
]
}
]
}

```

5. Untuk tutorial ini, beri nama kebijakan `ParallelClusterKmsPolicy`, lalu pilih Create Policy.
6. Buat catatan kebijakan ARN. Anda membutuhkannya untuk mengkonfigurasi kluster Anda.

## Mengkonfigurasi dan membuat cluster

Berikut ini adalah contoh konfigurasi kluster yang mencakup sistem file bersama Amazon Elastic Block Store dengan enkripsi.

```

Region: eu-west-1
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: my-ssh-key
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: q1
 ComputeResources:
 - Name: t2micro
 InstanceType: t2.micro

```

```
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
 SharedStorage:
 - MountDir: /shared/efs1
 Name: shared-efs1
 StorageType: Ebs
 EbsSettings:
 Encrypted: True
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

Ganti item dalam teks merah dengan nilai Anda sendiri. Kemudian, buat klaster yang menggunakan AWS KMS kunci Anda untuk mengenkripsi data Anda di Amazon EBS.

Konfigurasinya serupa untuk sistem file Amazon EFS dan FSx for Lustre.

SharedStorageKonfigurasi Amazon EFS adalah sebagai berikut.

```
...
 SharedStorage:
 - MountDir: /shared/efs1
 Name: shared-efs1
 StorageType: Efs
 EfsSettings:
 Encrypted: True
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

FSx untuk SharedStorage konfigurasi Lustre adalah sebagai berikut.

```
...
 SharedStorage:
 - MountDir: /shared/fsx1
 Name: shared-fsx1
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeploymentType: PERSISTENT_1
 PerUnitStorageThroughput: 200
```

```
KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

## Menjalankan pekerjaan dalam kluster mode antrian ganda

Tutorial ini mencakup cara menjalankan yang pertama "Hello World" pekerjaan di AWS ParallelCluster bersama [beberapa mode antrian](#).

Saat menggunakan AWS ParallelCluster antarmuka baris perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Layanan yang digunakan oleh AWS ParallelCluster](#).

The AWS ParallelCluster UI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya di dalam AWS Kategori Tier Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Biaya UI](#).

### Prasyarat

- AWS ParallelCluster [diinstal](#).
- The AWS CLI [diinstal dan dikonfigurasi](#).
- Anda memiliki [Pasangan kunci EC2](#).
- Anda memiliki peran IAM dengan [izin](#) yang diperlukan untuk menjalankan [pcluster](#) CLI.

## Konfigurasi kluster Anda

Pertama, verifikasi bahwa AWS ParallelCluster diinstal dengan benar dengan menjalankan perintah berikut.

```
$ pcluster version
```

Untuk informasi selengkapnya tentang `pcluster version`, lihat [pcluster version](#).

Perintah ini mengembalikan versi berjalan AWS ParallelCluster.

Selanjutnya, jalankan `pcluster configure` untuk menghasilkan file konfigurasi dasar. Ikuti semua petunjuk yang mengikuti perintah ini.

```
$ pcluster configure --config multi-queue-mode.yaml
```

Untuk informasi lebih lanjut tentang `pcluster configure` perintah, lihat [pcluster configure](#).

Setelah Anda menyelesaikan langkah ini, file konfigurasi dasar bernama `multi-queue-mode.yaml` muncul. File ini berisi konfigurasi cluster dasar.

Pada langkah berikutnya, Anda memodifikasi file konfigurasi baru Anda dan meluncurkan cluster dengan beberapa antrian.

#### Note

Beberapa contoh yang digunakan tutorial ini tidak memenuhi syarat tingkat gratis.

Untuk tutorial ini, ubah file konfigurasi Anda agar sesuai dengan konfigurasi berikut. Item yang disorot dengan warna merah mewakili nilai file konfigurasi Anda. Pertahankan nilai-nilai Anda sendiri.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: c5.xlarge
Networking:
 SubnetId: subnet-abcdef01234567890
Ssh:
 KeyName: yourkeypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: spot
 ComputeResources:
 - Name: c5xlarge
 InstanceType: c5.xlarge
 MinCount: 1
 MaxCount: 10
 - Name: t2micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
 - Name: ondemand
```

```
ComputeResources:
- Name: c52xlarge
 InstanceType: c5.2xlarge
 MinCount: 0
 MaxCount: 10
Networking:
 SubnetIds:
 - subnet-021345abcdef6789
```

## Buat cluster Anda

Buat cluster yang diberi nama `multi-queue-cluster` berdasarkan file konfigurasi Anda.

```
$ pcluster create-cluster --cluster-name multi-queue-cluster --cluster-configuration
multi-queue-mode.yaml
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.7.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Untuk informasi lebih lanjut tentang `pcluster create-cluster` perintah, lihat [pcluster create-cluster](#).

Untuk memeriksa status cluster, jalankan perintah berikut.

```
$ pcluster list-clusters
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.7.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
```

```
}
}
```

Ketika cluster dibuat, `clusterStatus` lapangan menunjukkan `CREATE_COMPLETE`.

## Masuk ke node kepala

Gunakan file kunci SSH pribadi Anda untuk masuk ke node kepala.

```
$ pcluster ssh --cluster-name multi-queue-cluster -i ~/path/to/yourkeyfile.pem
```

Untuk informasi selengkapnya tentang `pcluster ssh`, lihat [pcluster ssh](#).

Setelah masuk, jalankan `sinfo` perintah untuk memverifikasi bahwa antrian penjadwal Anda telah diatur dan dikonfigurasi.

Untuk informasi lebih lanjut tentang `sinfo`, lihat [sinfo](#) di Slurm dokumentasi.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 18 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[1-9]
spot* up infinite 2 idle spot-st-c5xlarge-1,spot-st-t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Output menunjukkan bahwa Anda memiliki satu `2.micro` dan satu `c5.xlarge` menghitung simpul di `idle` status yang tersedia di cluster Anda.

Node lain semuanya dalam keadaan hemat daya, ditunjukkan oleh `~` akhiran dalam keadaan simpul, tanpa instance EC2 yang mendukungnya. Antrian default ditunjukkan oleh `*` akhiran setelah nama antreannya. `spot` adalah antrian pekerjaan default Anda.

## Jalankan pekerjaan dalam beberapa mode antrian

Selanjutnya, cobalah untuk menjalankan pekerjaan untuk tidur sebentar. Pekerjaan itu kemudian mengeluarkan nama hostnya sendiri. Pastikan skrip ini dapat dijalankan oleh pengguna saat ini.

```
$ tee <<EOF hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from \$(hostname)"
EOF
```

```
$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

Kirim pekerjaan menggunakan `sbatch` perintah. Minta dua node untuk pekerjaan ini dengan `-N 2` opsi, dan verifikasi bahwa pekerjaan berhasil dikirimkan. Untuk informasi lebih lanjut tentang `sbatch`, lihat [sbatch](#) di Dokumentasi `slurm`.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 1
```

Anda dapat melihat antrian Anda dan memeriksa status pekerjaan dengan `squeue` perintah. Karena Anda tidak menentukan antrian tertentu, antrian default (`spot`) digunakan. Untuk informasi lebih lanjut tentang `squeue`, lihat [squeue](#) di `Slurm` dokumentasi.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 1 spot wrap ec2-user R 0:10 2 spot-st-c5xlarge-1,spot-st-
t2micro-1
```

Output menunjukkan bahwa pekerjaan saat ini dalam keadaan berjalan. Tunggu pekerjaan selesai. Ini membutuhkan waktu sekitar 30 detik. Kemudian, lihat `squeue` lagi.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Sekarang pekerjaan dalam antrian telah selesai, cari file output yang diberi nama `slurm-1.out` di direktori Anda saat ini.

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```

Output menunjukkan bahwa pekerjaan berjalan dengan sukses di `spot-st-t2micro-1` dan `spot-st-c5xlarge-1` simpul.

Sekarang kirimkan pekerjaan yang sama dengan menentukan batasan untuk instance tertentu dengan perintah berikut.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 2
```

Anda menggunakan parameter ini untuk `sbatch`:

- `-N 3`— meminta tiga node.
- `-p spot`— menyerahkan pekerjaan ke spot antrian. Anda juga dapat mengirimkan pekerjaan ke ondemand antrian dengan menentukan `-p ondemand`.
- `-C "[c5.xlarge*1&t2.micro*2]"`— menentukan kendala node tertentu untuk pekerjaan ini. Ini meminta satu `c5.xlarge` simpul dan dua `t2.micro` node yang akan digunakan untuk pekerjaan ini.

Jalankan `sinfo` perintah untuk melihat node dan antrian. Antrian di AWS ParallelCluster disebut partisi di Slurm.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 1 alloc# spot-dy-t2micro-1
spot* up infinite 17 idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot* up infinite 1 mix spot-st-c5xlarge-1
spot* up infinite 1 alloc spot-st-t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Node menyala. Hal ini ditunjukkan oleh #akhiran pada keadaan simpul. Jalankan `squeue` perintah untuk melihat informasi tentang pekerjaan di cluster.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 spot wrap ec2-user CF 0:04 3 spot-dy-c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```

Pekerjaan Anda ada di CF (CONFIGURING) state, menunggu instance untuk ditingkatkan dan bergabung dengan cluster.

Setelah sekitar tiga menit, node tersedia dan pekerjaan memasuki R (RUNNING) negara.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 spot wrap ec2-user R 0:04 3 spot-dy-c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```



```

2 spot wrap ec2-user R 0:07 3 spot-dy-t2micro-1,spot-st-
c5xlarge-1,spot-st-t2micro-1

```

Pekerjaan selesai, dan ketiga node ada diidlenegara.

```

$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 17 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot* up infinite 3 idle spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]

```

Kemudian, setelah tidak ada pekerjaan yang tersisa dalam antrian, periksa `slurm-2.out` di direktori lokal Anda.

```

$ cat slurm-2.out
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
Hello World from spot-st-c5xlarge-1

```

Ini adalah keadaan akhir dari cluster.

```

$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 17 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot* up infinite 3 idle spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]

```

Setelah log off dari cluster, Anda dapat membersihkan dengan menjalankan `pcluster delete-cluster`. Selengkapnya, lihat [pcluster list-clusters](#) dan [pcluster delete-cluster](#).

```

$ pcluster list-clusters
{
 "clusters": [
 {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_COMPLETE",

```

```
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_COMPLETE"
 }
]
}
$ pcluster delete-cluster -n multi-queue-cluster
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.1.4",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

## Menggunakan API AWS ParallelCluster ini

Dalam tutorial ini, Anda membangun dan menguji API dengan [Amazon API Gateway](#) dan AWS ParallelCluster CloudFormation template. Kemudian, Anda menggunakan contoh klien yang tersedia GitHub untuk menggunakan API. Untuk informasi lebih lanjut tentang menggunakan API, lihat [API AWS ParallelCluster](#).

Tutorial ini dikutip dari [HPC For Public Sector Customers Workshop](#).

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS layanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Biaya UI](#).

### Prasyarat

- [Terinstal](#) dan dikonfigurasi di lingkungan komputasi Anda. AWS CLI

- AWS ParallelCluster dipasang di lingkungan virtual. Untuk informasi selengkapnya, lihat [Menginstal AWS ParallelCluster di lingkungan virtual](#).
- Anda memiliki [pasangan kunci EC2](#).
- Anda memiliki peran IAM dengan [izin](#) yang diperlukan untuk menjalankan CLI. [pcluster](#)

## Langkah 1: Membangun API dengan Amazon API Gateway

Tetap di direktori pengguna rumah Anda dan aktifkan lingkungan virtual Anda:

1. Instal prosesor baris perintah JSON yang bermanfaat.

```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. Jalankan perintah berikut untuk mendapatkan AWS ParallelCluster versi Anda dan menetapkan ke variabel lingkungan.

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" | tee -a ~/.bashrc
```

3. Buat variabel lingkungan dan tetapkan ID Wilayah Anda untuk itu.

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" | tee -a ~/.bashrc
```

4. Jalankan perintah berikut untuk menyebarkan API.

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" | tee -a ~/.bashrc
```

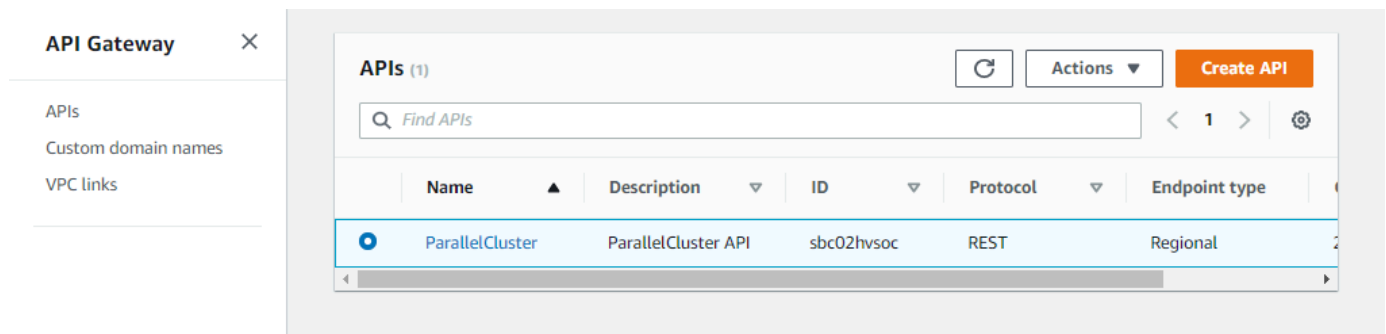
```
aws cloudformation create-stack \
 --region ${AWS_DEFAULT_REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
 ${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
 parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
 --parameters ParameterKey=EnableIamAdminAccess,ParameterValue=true
```

```
{
 "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-api-
stack/abcd1234-ef56-gh78-ei90-1234abcd5678"
}
```

Setelah proses selesai, lanjutkan ke langkah berikutnya.

## Langkah 2: Uji API di konsol Amazon API Gateway

1. Masuk ke AWS Management Console.
2. Arahkan ke [konsol Amazon API Gateway](#).
3. Pilih penerapan API Anda.



4. Pilih Tahapan dan pilih panggung.

Amazon API Gateway APIs > ParallelCluster (sbc02hvsoc) > Stages > prod Show all hints ?

APIs Stages **Create** prod Stage Editor Delete Stage Configure Tags

Invoke URL: <https://sbc02hvsoc.execute-api.us-east-1.amazonaws.com/prod>

Settings Logs/Tracing Stage Variables SDK Generation Export

Deployment History Documentation History Canary

Cache Settings

Enable API cache

Default Method Throttling

Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current account level throttling rate is **10000** requests per second with a burst of 5000 requests. [Read more about API Gateway throttling](#)

Enable throttling  ⓘ

Rate  requests per second

Burst  requests

Web Application Firewall (WAF) [Learn more.](#)

Select the Web ACL to be applied to this stage.

Web ACL  [Create Web ACL](#)

Client Certificate

Select the client certificate that API Gateway will use to call your integration endpoints in this stage.

Certificate

- Perhatikan URL yang disediakan API Gateway untuk mengakses atau menerapkan API Anda. Ini disorot dengan warna biru.
- Pilih Sumber Daya, dan pilih **GET** di bawah/**clusters**.
- Pilih ikon UJI dan kemudian gulir ke bawah dan pilih ikon UJI.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET

Resources Actions **/v3/clusters - GET - Method Execution**

The screenshot displays the AWS API Gateway console for a specific API. The breadcrumb trail at the top indicates the path: APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET. Below this, the 'Resources' section shows a tree view with the following structure:

- /
- /v3
  - /clusters
    - GET (highlighted)
    - POST
    - /{clusterName}
      - DELETE
      - GET
      - PUT
    - /computefleet
      - GET
      - PATCH
    - /instances
      - DELETE
      - GET
    - /logstreams
      - GET
      - /{logStreamName}
        - GET
    - /stackevents
      - GET
  - /images
    - /custom

The main area shows the 'Method Execution' for the GET /v3/clusters endpoint. It features a 'Client' box with a 'TEST' button and a lightning bolt icon. To the right, the 'Method Request' box contains the following details:

- Auth:** AWS IAM
- ARN:** arn:aws:execute-api:us-east-1:123456789012:sbc02hvsoc/\*/GET/v3/clusters
- Query Strings:** region, nextToken, clusterStatus

Below the request box is the 'Method Response' box, which contains the text: 'Select an integration response.' Arrows indicate the flow of data from the client to the request and back to the response.

Respons terhadap Anda /clusters GET muncul.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfwk2) > GET

Show all hints ?

Resources Actions

Method Execution /v3/clusters - GET - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

Path: /v3/clusters  
Request: /v3/clusters  
Status: 200  
Latency: 3203 ms  
Response Body

No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.

Query Strings: {clusters} param1=value1&param2=value2

Headers: {clusters} Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

Stage Variables: No stage variables exist for this method.

Client Certificate: No client certificates have been generated.

```
{
 "clusters": [
 {
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/test-cluster/4450d850-b684-11ec-84a7-0a047567c9f3",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "clusterName": "test-cluster",
 "clusterStatus": "CREATE_COMPLETE",
 "region": "us-east-1",
 "version": "3.1.2"
 }
]
}
```

Response Headers: {"Content-Length": "360", "X-Amzn-Trace-Id": "Root=1-62686455-c1cf243417b2721e33822ac5;Sampled=1", "Content-Type": "application/json"}

Logs

### Langkah 3: Mempersiapkan dan menguji contoh klien untuk memanggil API

Kloning kode AWS ParallelCluster sumber, cd ke api direktori, dan instal pustaka klien Python.

- ```
$ git clone -b v${PCLUSTER_VERSION} https://github.com/aws/aws-parallelcluster aws-parallelcluster-v${PCLUSTER_VERSION}
cd aws-parallelcluster-v${PCLUSTER_VERSION}/api
```

```
$ pip3 install client/src
```

- Arahkan kembali ke direktori pengguna home Anda.

3. Ekspor URL dasar API Gateway yang digunakan klien saat menjalankan.

```
$ export PCLUSTER_API_URL=$( aws cloudformation describe-stacks
--stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiInvokeUrl`.OutputValue' --output text )
echo "export PCLUSTER_API_URL=${PCLUSTER_API_URL}" |tee -a ~/.bashrc
```

4. Ekspor nama cluster yang digunakan klien untuk membuat klaster.

```
$ export CLUSTER_NAME="test-api-cluster"
echo "export CLUSTER_NAME=${CLUSTER_NAME}" |tee -a ~/.bashrc
```

5. Jalankan perintah berikut untuk menyimpan kredensi yang digunakan klien contoh untuk mengakses API.

```
$ export PCLUSTER_API_USER_ROLE=$( aws cloudformation describe-
stacks --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiUserRole`.OutputValue' --output text )
echo "export PCLUSTER_API_USER_ROLE=${PCLUSTER_API_USER_ROLE}" |tee -a ~/.bashrc
```

Langkah 4: Salin skrip kode klien dan jalankan tes klaster

1. Salin contoh kode klien berikut ke `test_pcluster_client.py` direktori pengguna home Anda. Kode klien membuat permintaan untuk melakukan hal berikut:
 - Buat cluster.
 - Jelaskan cluster.
 - Daftar cluster.
 - Jelaskan armada komputasi.
 - Jelaskan contoh cluster.

```
# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# 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.
#
# 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.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)
    fleet_ops = cluster_compute_fleet_api.ClusterComputeFleetApi(api_client)
    instance_ops = cluster_instances_api.ClusterInstancesApi(api_client)
```

```
# Create cluster
build_done = False
try:
    with open('cluster-config.yaml', encoding="utf-8") as f:
        body = CreateClusterRequestContent(cluster_name=cluster_name,
cluster_configuration=f.read())
        api_response = cluster_ops.create_cluster(body, region=region)
except pcluster_client.ApiException as e:
    print("Exception when calling create_cluster: %s\n" % e)
    build_done = True
time.sleep(60)

# Confirm cluster status with describe_cluster
while not build_done:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('CREATE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
        elif api_response.cluster_status == ClusterStatus('CREATE_COMPLETE'):
            print('READY!')
            build_done = True
        else:
            print('ERROR!!!!')
            build_done = True
    except pcluster_client.ApiException as e:
        print("Exception when calling describe_cluster: %s\n" % e)

# List clusters
try:
    api_response = cluster_ops.list_clusters(region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling list_clusters: %s\n" % e)

# DescribeComputeFleet
try:
    api_response = fleet_ops.describe_compute_fleet(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling compute fleet: %s\n" % e)
```

```
# DescribeClusterInstances
try:
    api_response = instance_ops.describe_cluster_instances(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling describe_cluster_instances: %s\n" % e)
```

2. Buat konfigurasi cluster.

```
$ pcluster configure --config cluster-config.yaml
```

3. Pustaka Klien API secara otomatis mendeteksi detail konfigurasi dari variabel lingkungan Anda (misalnya, `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, atau `AWS_SESSION_TOKEN`) atau `$HOME/.aws`. Perintah berikut mengalihkan peran IAM Anda saat ini ke yang ditunjuk `ParallelClusterApiUserRole`.

```
$ eval $(aws sts assume-role --role-arn ${PCLUSTER_API_USER_ROLE} --role-session-name ApiTestSession | jq -r '.Credentials | "export AWS_ACCESS_KEY_ID=\(.AccessKeyId)\nexport AWS_SECRET_ACCESS_KEY=\(.SecretAccessKey)\nexport AWS_SESSION_TOKEN=\(.SessionToken)\n"')
```

Kesalahan untuk diperhatikan:

Jika Anda melihat kesalahan yang mirip dengan berikut ini, Anda sudah berasumsi `ParallelClusterApiUserRole` dan Anda `AWS_SESSION_TOKEN` telah kedaluwarsa.

```
An error occurred (AccessDenied) when calling the AssumeRole operation:
User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/ApiTestSession
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::XXXXXXXXXXXX:role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Jatuhkan peran dan kemudian jalankan kembali `aws sts assume-role` perintah untuk menggunakan `ParallelClusterApiUserRole`

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
```

```
unset AWS_ACCESS_KEY_ID
```

Untuk memberikan izin pengguna saat ini untuk akses API, Anda harus [memperluas Kebijakan Sumber Daya](#).

4. Jalankan perintah berikut untuk menguji contoh klien.

```
$ python3 test_pcluster_client.py
{'cluster_configuration': 'Region: us-east-1\n'
                          'Image:\n'
                          '  Os: alinux2\n'
                          'HeadNode:\n'
                          '  InstanceType: t2.micro\n'
                          '  Networking . . . :\n'
                          '    SubnetId: subnet-1234567890abcdef0\n'
                          '  Ssh:\n'
                          '    KeyName: adpc\n'
                          'Scheduling:\n'
                          '  Scheduler: slurm\n'
                          '  SlurmQueues:\n'
                          '    - Name: queue1\n'
                          '    ComputeResources:\n'
                          '      - Name: t2micro\n'
                          '        InstanceType: t2.micro\n'
                          '        MinCount: 0\n'
                          '        MaxCount: 10\n'
                          '    Networking . . . :\n'
                          '      SubnetIds:\n'
                          '        - subnet-1234567890abcdef0\n',
  'cluster_name': 'test-api-cluster'}
{'cloud_formation_stack_status': 'CREATE_IN_PROGRESS',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
  'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-not-delete...'},
  'cluster_name': 'test-api-cluster',
  'cluster_status': 'CREATE_IN_PROGRESS',
  'compute_fleet_status': 'UNKNOWN',
  'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'region': 'us-east-1',
```

```
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
.
.
.
. . . working . . . {'cloud_formation_stack_status': 'CREATE_COMPLETE',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'CREATE_COMPLETE',
'compute_fleet_status': 'RUNNING',
'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 21, 46,
tzinfo=tzlocal()),
'private_ip_address': '172.31.27.153',
'public_ip_address': '52.90.156.51',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
READY!
```

Langkah 5: Salin skrip kode klien dan hapus cluster

1. Salin contoh kode klien berikut ke `delete_cluster_client.py`. Kode klien membuat permintaan untuk menghapus cluster.

```
# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# 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.
#
# 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.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)

    # Delete the cluster
    gone = False
```

```

try:
    api_response = cluster_ops.delete_cluster(cluster_name, region=region)
except pcluster_client.ApiException as e:
    print("Exception when calling delete_cluster: %s\n" % e)
time.sleep(60)

# Confirm cluster status with describe_cluster
while not gone:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('DELETE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
    except pcluster_client.ApiException as e:
        gone = True
        print("DELETE COMPLETE or Exception when calling describe_cluster: %s
\n" % e)

```

2. Jalankan perintah berikut untuk menghapus cluster.

```

$ python3 delete_cluster_client.py
{'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 53, 48,
tzinfo=tzlocal()),
'private_ip_address': '172.31.17.132',
'public_ip_address': '34.201.100.37',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],

```

```
'version': '3.1.3'}
.
.
.
. . . working . . . {'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
. . . working . . . DELETE COMPLETE or Exception when calling describe_cluster:
(404)
Reason: Not Found
.
.
.
HTTP response body: {"message": "Cluster 'test-api-cluster' does not exist or
belongs to an incompatible ParallelCluster major version."}
```

3. Setelah Anda selesai menguji, unset variabel lingkungan.

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

Langkah 6: Membersihkan

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

1. Dari AWS CloudFormation konsol, pilih tumpukan API dan kemudian pilih Hapus.
2. Jalankan perintah berikut jika menggunakan AWS CLI.

Menggunakan AWS CloudFormation.


```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```

Membuat cluster dengan akuntansi Slurm

Pelajari cara mengkonfigurasi dan membuat cluster dengan Slurm akuntansi. Untuk informasi selengkapnya, lihat [Slurm akuntansi dengan AWS ParallelCluster](#).

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS layanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelCluster Biaya UI](#).

Dalam tutorial ini, Anda menggunakan [CloudFormation cepat-membuat template untuk membuat database MySQL Amazon Aurora serverless](#). Template menginstruksikan CloudFormation untuk membuat semua komponen yang diperlukan untuk menyebarkan database Amazon Aurora tanpa server pada VPC yang sama dengan kluster. Template juga menciptakan jaringan dasar dan konfigurasi keamanan untuk koneksi antara cluster dan database.

Note

[Dimulai dengan versi 3.3.0, AWS ParallelCluster mendukung Slurm akuntansi dengan parameter konfigurasi cluster `SlurmSettings/Database`.](#)

Note

Template pembuatan cepat berfungsi sebagai contoh. Template ini tidak mencakup semua kasus penggunaan yang mungkin untuk server database Slurm akuntansi. Anda bertanggung jawab untuk membuat server database dengan konfigurasi dan kapasitas yang sesuai untuk beban kerja produksi Anda.

Prasyarat:

- AWS ParallelCluster [dipasang](#).
- [diinstal dan dikonfigurasi. AWS CLI](#)
- Anda memiliki [pasangan kunci EC2](#).
- Anda memiliki peran IAM dengan [izin](#) yang diperlukan untuk menjalankan CLI. [pcluster](#)
- Wilayah tempat Anda menerapkan template pembuatan cepat mendukung v2 tanpa server Amazon Aurora MySQL. Untuk informasi selengkapnya, lihat [Aurora Serverless v2 dengan Aurora MySQL](#).

Langkah 1: Buat VPC dan subnet untuk AWS ParallelCluster

Untuk menggunakan CloudFormation template yang disediakan untuk database Slurm akuntansi, Anda harus memiliki VPC untuk cluster siap. Anda dapat melakukan ini secara manual atau sebagai bagian dari [Konfigurasi dan buat cluster dengan antarmuka baris AWS ParallelCluster perintah](#) prosedur. Jika Anda sudah menggunakan AWS ParallelCluster, Anda mungkin memiliki VPC siap untuk penyebaran cluster dan server database.

Langkah 2: Buat tumpukan database

Gunakan [CloudFormation cepat-membuat template](#) untuk membuat tumpukan database untuk Slurm akuntansi. Template membutuhkan input berikut:

- Kredensi server database, khususnya nama pengguna dan kata sandi admin.
- Ukuran cluster tanpa Amazon Aurora server. Hal ini tergantung pada pemuatan cluster yang diharapkan.
- Parameter jaringan, khususnya target VPC dan subnet atau blok CIDR untuk pembuatan subnet.

Pilih kredensi dan ukuran yang sesuai untuk server database Anda. Untuk opsi jaringan, Anda diharuskan untuk menggunakan VPC yang sama dengan yang digunakan AWS ParallelCluster kluster. Anda dapat membuat subnet untuk database dan meneruskannya sebagai masukan ke template. Atau, sediakan dua blok CIDR yang terpisah untuk dua subnet dan biarkan CloudFormation template membuat dua subnet untuk blok CIDR. Pastikan bahwa blok CIDR tidak tumpang tindih dengan subnet yang ada. Jika blok CIDR tumpang tindih dengan subnet yang ada, tumpukan gagal dibuat.

Server database membutuhkan waktu beberapa menit untuk dibuat.

Langkah 3: Buat cluster dengan Slurm akuntansi diaktifkan

CloudFormationTemplate yang disediakan menghasilkan CloudFormation tumpukan dengan beberapa output yang ditentukan. Dari AWS Management Console, Anda dapat melihat output di tab Outputs dalam tampilan CloudFormation tumpukan. Untuk mengaktifkan Slurm akuntansi, beberapa output ini harus digunakan dalam file konfigurasi AWS ParallelCluster cluster:

- DatabaseHost: Digunakan untuk parameter konfigurasi [SlurmSettingsDatabase//Uri](#)cluster.
- DatabaseAdminUser: Digunakan untuk [SlurmSettings/Database/nilai](#) parameter konfigurasi [UserName](#)cluster.
- DatabaseSecretArn: Digunakan untuk parameter konfigurasi [SlurmSettingsDatabase//PasswordSecretArn](#)cluster.
- DatabaseClientSecurityGroup: Ini adalah grup keamanan yang melekat pada node kepala cluster yang didefinisikan dalam parameter [SecurityGroups](#) konfigurasi [HeadNodeNetworking](#)//.

Perbarui Database parameter file konfigurasi klaster Anda dengan nilai keluaran. Gunakan [pcluster](#) CLI untuk membuat cluster.

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

Setelah cluster dibuat, Anda dapat mulai menggunakan perintah Slurm akuntansi seperti `sacctmgr` atau `sacct`.

Mengembalikan ke versi dokumen Manajer AWS Sistem sebelumnya

Pelajari cara kembali ke versi dokumen Manajer AWS Sistem sebelumnya. Untuk informasi selengkapnya tentang dokumen SSM, lihat [Dokumen Manajer AWS Sistem](#) di Panduan Pengguna Manajer AWS Sistem.

Saat menggunakan antarmuka baris AWS ParallelCluster perintah (CLI) atau API, Anda hanya membayar AWS sumber daya yang dibuat saat membuat atau memperbarui AWS ParallelCluster

gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS Slayanan yang digunakan oleh AWS ParallelCluster](#).

AWS ParallelClusterUI dibangun di atas arsitektur tanpa server dan Anda dapat menggunakannya dalam kategori Tingkat AWS Gratis untuk sebagian besar kasus. Untuk informasi selengkapnya, lihat [AWS ParallelClusterBiaya UI](#).

Prasyarat:

- Sebuah Akun AWS dengan izin untuk mengelola dokumen SSM.
- [diinstal dan dikonfigurasi. AWS CLI](#)

Kembalikan ke versi dokumen SSM sebelumnya

1. Di terminal Anda, jalankan perintah berikut untuk mendapatkan daftar dokumen SSM yang ada yang Anda miliki.

```
$ aws ssm list-documents --document-filter "key=Owner,value=Self"
```

2. Kembalikan dokumen SSM ke versi sebelumnya. Dalam contoh ini, kita kembali ke versi `SessionManagerRunShell` dokumen sebelumnya. Anda dapat menggunakan `SessionManagerRunShell` dokumen SSM untuk menyesuaikan setiap sesi shell SSM yang Anda mulai.
 - a. Temukan `DocumentVersion` parameter untuk `SessionManagerRunShell` dengan menjalankan perintah berikut:

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {
        "Name": "linuxcmd",
```

```

        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then
source /opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/
bash"
    }
],
"PlatformTypes": [
    "Windows",
    "Linux",
    "MacOS"
],
"DocumentType": "Session",
"SchemaVersion": "1.0",
"LatestVersion": "2",
"DefaultVersion": "1",
"DocumentFormat": "JSON",
"Tags": []
}
}

```

Versi terbaru adalah 2.

- b. Kembalikan ke versi sebelumnya dengan menjalankan perintah berikut:

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-  
version 2
```

3. Verifikasi bahwa versi dokumen telah dikembalikan dengan menjalankan describe-document perintah lagi:

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {

```

```
        "Name": "linuxcmd",
        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then source /
opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/bash"
    }
],
"PlatformTypes": [
    "Windows",
    "Linux",
    "MacOS"
],
"DocumentType": "Session",
"SchemaVersion": "1.0",
"LatestVersion": "1",
"DefaultVersion": "1",
"DocumentFormat": "JSON",
"Tags": []
}
}
```

Versi terbaru adalah 1.

Membuat cluster dengan AWS CloudFormation

Pelajari cara membuat klaster dengan sumber daya AWS ParallelCluster CloudFormation khusus. Untuk informasi selengkapnya, lihat [AWS CloudFormation sumber daya khusus](#).

Saat menggunakan AWS ParallelCluster, Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWS layanan yang digunakan oleh AWS ParallelCluster](#).

Prasyarat:

- AWS CLI [itu diinstal dan dikonfigurasi](#).
- Sebuah [key pair EC2](#).
- Peran IAM dengan [izin](#) yang diperlukan untuk menjalankan CLI [pcluster](#).

Pembuatan cluster dengan CloudFormation tumpukan cepat

Dalam tutorial ini, Anda menggunakan tumpukan quick-create untuk menyebarkan CloudFormation template yang membuat cluster dan sumber daya berikut: AWS

- CloudFormation Tumpukan root dibuat dengan menggunakan tumpukan CloudFormation quick-create.
- CloudFormation Tumpukan bersarang yang menyertakan kebijakan default, pengaturan VPC default, dan penyedia sumber daya khusus.
- Contoh tumpukan AWS ParallelCluster cluster dan cluster tempat Anda dapat masuk dan menjalankan pekerjaan.

Buat cluster dengan AWS CloudFormation

1. Masuk ke AWS Management Console.
2. Buka [tautan CloudFormation buat cepat untuk membuat](#) sumber daya berikut di konsol: CloudFormation
 - CloudFormation Tumpukan bersarang dengan VPC dengan subnet publik dan pribadi untuk menjalankan node kepala cluster dan node komputasi, masing-masing.
 - CloudFormation Tumpukan bersarang dengan sumber daya AWS ParallelCluster khusus untuk mengelola cluster.
 - CloudFormation Tumpukan bersarang dengan kebijakan default untuk mengelola klaster.
 - CloudFormation Tumpukan root untuk tumpukan bersarang.
 - Sebuah AWS ParallelCluster cluster dengan Slurm scheduler dan jumlah node komputasi yang ditentukan.

CloudFormation > Stacks > Create stack

Quick create stack

Template

Template URL
https://pcluster-cfn-us-east-2.s3.amazonaws.com/parallelcluster/3.5.0/templates/custom_resource/cluster-1-click.yaml

Stack description
AWS ParallelCluster CloudFormation Cluster

Stack name

Stack name
cluster-0

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AvailabilityZone
Availability zone where instances will be launched
us-east-2a

KeyName
KeyPair to login to the head node
Select AWS::EC2::KeyPair::KeyName

Capabilities

The following resource(s) require capabilities: [AWS::CloudFormation::Stack]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: {0}. Check the capabilities of these resources. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND

Cancel Create change set Create stack

3. Di bagian Quick create stack Parameters, masukkan nilai untuk parameter berikut:
 - a. Untuk KeyName, masukkan nama key pair EC2 Anda.
 - b. Untuk AvailabilityZone, pilih AZ untuk node cluster Anda, misalnya, us-east-1a.

4. Centang kotak untuk mengetahui setiap kemampuan akses di bagian bawah halaman.
5. Pilih Buat tumpukan.
6. Tunggu CloudFormation tumpukan mencapai CREATE_COMPLETE status.

Pembuatan cluster dengan AWS CloudFormation Command Line Interface (CLI)

Dalam tutorial ini, Anda menggunakan AWS Command Line Interface (CLI) CloudFormation untuk menyebarkan CloudFormation template yang membuat cluster.

Buat AWS sumber daya berikut:

- CloudFormation Tumpukan root dibuat dengan menggunakan tumpukan CloudFormation quick-create.
- CloudFormation Tumpukan bersarang yang menyertakan kebijakan default, penyiapan VPC default, dan penyedia sumber daya khusus.
- Contoh tumpukan AWS ParallelCluster cluster dan cluster tempat Anda dapat masuk dan menjalankan pekerjaan.

Ganti *input yang disorot dengan warna merah*, seperti *keypair*, dengan nilai Anda sendiri.

Buat cluster dengan AWS CloudFormation

1. Buat CloudFormation template bernama `cluster_template.yaml` dengan konten berikut:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
  KeyName:
    Description: KeyPair to login to the head node
    Type: AWS::EC2::KeyPair::KeyName

  AvailabilityZone:
    Description: Availability zone where instances will be launched
    Type: AWS::EC2::AvailabilityZone::Name
    Default: us-east-2a

Mappings:
```

```
ParallelCluster:
  Constants:
    Version: 3.7.0

Resources:
  PclusterClusterProvider:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
        - { Version: !FindInMap [ParallelCluster, Constants, Version] }

  PclusterVpc:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        PublicCIDR: 10.0.0.0/24
        PrivateCIDR: 10.0.16.0/20
        AvailabilityZone: !Ref AvailabilityZone
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/networking/public-private-
          ${Version}.cfn.json
        - { Version: !FindInMap [ParallelCluster, Constants, Version] }

  PclusterCluster:
    Type: Custom::PclusterCluster
    Properties:
      ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
      ClusterName: !Sub 'c-${AWS::StackName}'
      ClusterConfiguration:
        Image:
          Os: alinux2
        HeadNode:
          InstanceType: t2.medium
          Networking:
            SubnetId: !GetAtt [ PclusterVpc , Outputs.PublicSubnetId ]
          Ssh:
            KeyName: !Ref KeyName
        Scheduling:
          Scheduler: slurm
          SlurmQueues:
            - Name: queue0
```

```

ComputeResources:
  - Name: queue0-cr0
    InstanceType: t2.micro
Networking:
  SubnetIds:
    - !GetAtt [ PclusterVpc , Outputs.PrivateSubnetId ]
Outputs:
  HeadNodeIp:
    Description: The Public IP address of the HeadNode
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]

```

- Jalankan perintah AWS CLI berikut untuk menyebarkan CloudFormation tumpukan untuk pembuatan dan pengelolaan cluster.

```

$ aws cloudformation deploy --template-file ./cluster_template.yaml \
  --stack-name mycluster \
  --parameter-overrides KeyName=keypair \
    AvailabilityZone=us-east-2b \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND

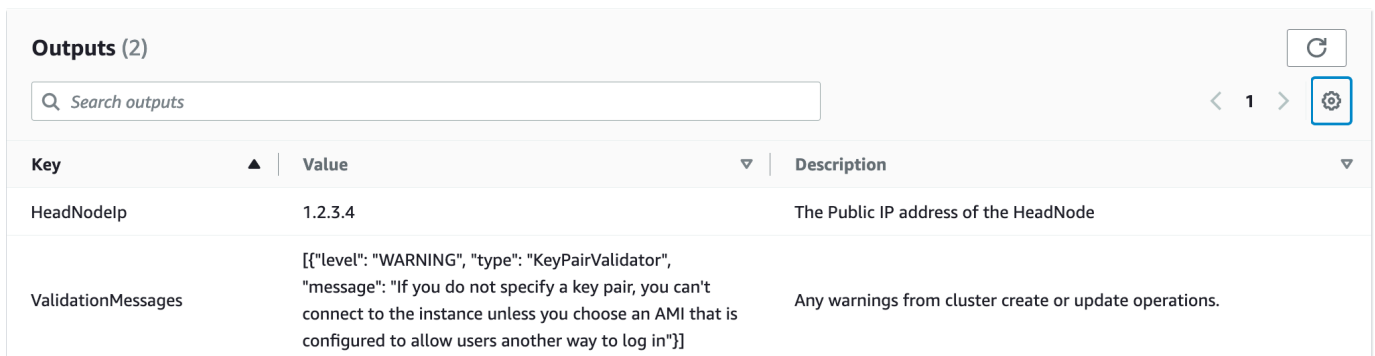
```

Lihat keluaran CloudFormation kluster

Lihat output CloudFormation cluster untuk mendapatkan detail cluster yang berguna.

ValidationMessagesProperti yang ditambahkan menyediakan akses ke pesan validasi dari operasi pembuatan dan pembaruan cluster.

- Arahkan ke [CloudFormation konsol](#) dan pilih tumpukan yang menyertakan sumber daya AWS ParallelCluster kustom Anda.
- Pilih Stack details, dan pilih tab Outputs.



Key	Value	Description
HeadNodeIp	1.2.3.4	The Public IP address of the HeadNode
ValidationMessages	<pre> [{"level": "WARNING", "type": "KeyPairValidator", "message": "If you do not specify a key pair, you can't connect to the instance unless you choose an AMI that is configured to allow users another way to log in"}] </pre>	Any warnings from cluster create or update operations.

Pesan validasi mungkin terpotong. Untuk informasi selengkapnya tentang cara mengambil log, lihat [Memecahkan Masalah AWS ParallelCluster](#).

Akses kluster Anda

Akses cluster.

ssh ke simpul kepala cluster

1. Setelah penyebaran CloudFormation tumpukan selesai, dapatkan alamat IP node kepala dengan perintah berikut:

```
$ HEAD_NODE_IP=$(aws cloudformation describe-stacks --stack-name=mycluster --query "Stacks|[0].Outputs[?OutputKey=='HeadNodeIp']|[0].OutputValue" --output=text)
```

Anda juga dapat mengambil alamat IP node kepala dari HeadNodeIp parameter di tab Output tumpukan cluster di CloudFormation konsol.

Anda dapat menemukan alamat IP node kepala di sini karena ditambahkan di Outputs bagian CloudFormation template cluster, khusus untuk contoh cluster ini.

2. Connect ke node kepala cluster dengan menjalankan perintah berikut:

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

Hapus

Hapus kluster .

1. Jalankan perintah AWS CLI berikut untuk menghapus CloudFormation tumpukan dan cluster.

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. Periksa status penghapusan tumpukan dengan menjalankan perintah berikut.

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

AWS ParallelClusterIntegrasi UI dengan Pusat Identitas

Tujuan dari tutorial ini adalah untuk mendemonstrasikan cara mengintegrasikan AWS ParallelCluster UI dengan IAM Identity Center untuk solusi masuk tunggal yang menyatukan pengguna di Active Directory yang dapat dibagikan dengan cluster. AWS ParallelCluster

Saat menggunakanAWS ParallelCluster, Anda hanya membayar AWS sumber daya yang dibuat saat Anda membuat atau memperbarui AWS ParallelCluster gambar dan cluster. Untuk informasi selengkapnya, lihat [AWSlayanan yang digunakan oleh AWS ParallelCluster](#).

Prasyarat:

- AWS ParallelClusterUI yang ada yang dapat diinstal mengikuti petunjuk di [sini](#).
- Direktori Aktif Terkelola yang ada, lebih disukai yang juga akan Anda gunakan untuk [mengintegrasikan](#). AWS ParallelCluster

Aktifkan Pusat Identitas IAM

Jika Anda sudah memiliki pusat identitas yang terhubung ke AWS Managed Microsoft AD (Direktori Aktif) Anda dapat digunakan dan Anda dapat melompat ke bagian Menambahkan Aplikasi Anda ke Pusat Identitas IAM.

Jika Anda belum memiliki pusat identitas yang terhubung ke sebuahAWS Managed Microsoft AD, ikuti langkah-langkah di bawah ini untuk mengaturnya.

Mengaktifkan Pusat Identitas

1. Di konsol, arahkan ke Pusat Identitas IAM. (Pastikan Anda berada di wilayah tempat Anda memilikiAWS Managed Microsoft AD.)
2. Klik tombol Aktifkan, ini mungkin menanyakan apakah Anda ingin mengaktifkan organisasi, ini adalah persyaratan sehingga Anda dapat memilih untuk mengaktifkannya. Catatan: Ini akan mengirim email ke administrator akun Anda dengan email konfirmasi bahwa Anda harus mengikuti tautan untuk mengonfirmasi.

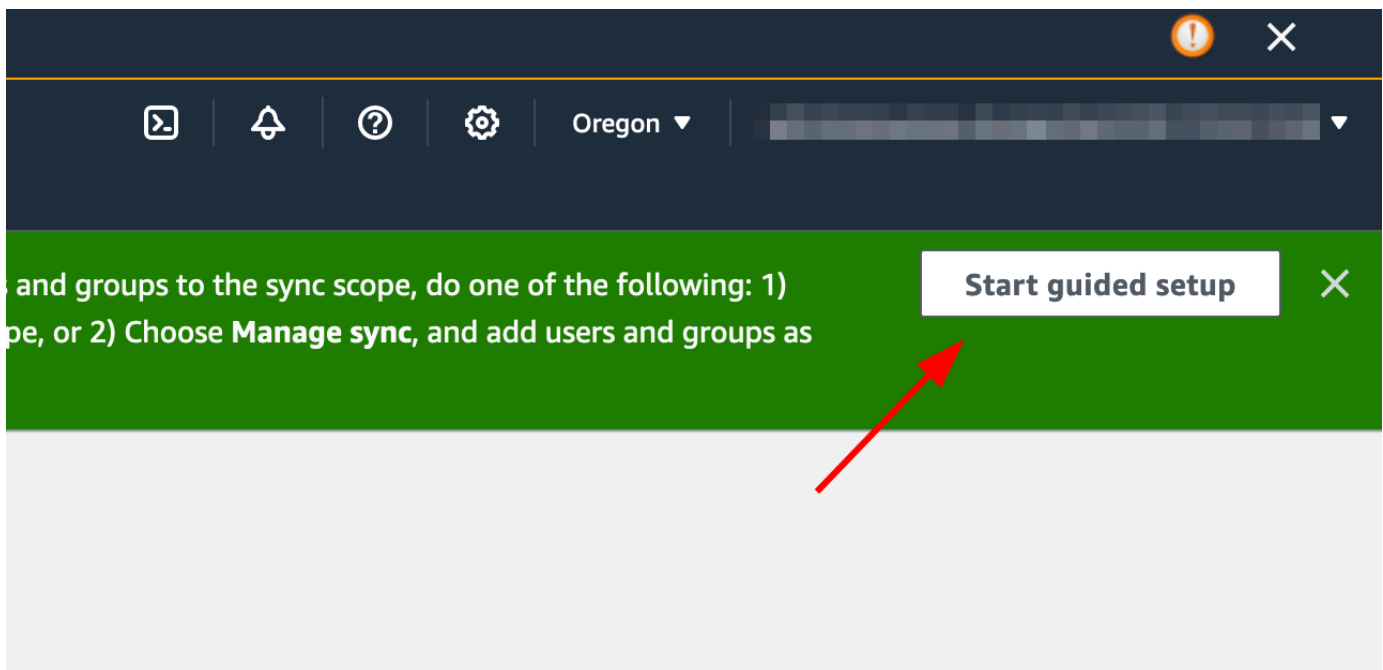
Menghubungkan Pusat Identitas ke AD Terkelola

1. Pada halaman berikutnya setelah mengaktifkan pusat identitas, Anda akan melihat Langkah Pengaturan yang Disarankan, di bawah Langkah 1, pilih Pilih Sumber Identitas Anda.

2. Di bagian Sumber Identitas, klik menu tarik-turun Tindakan (di kanan atas), lalu pilih Ubah Sumber Identitas.
3. Pilih Active Directory.
4. Di bawah Direktori yang Ada, pilih direktori Anda.
5. Klik Selanjutnya.
6. Tinjau perubahan Anda, gulir ke bawah, ketik TERIMA ke dalam kotak teks untuk mengonfirmasi, lalu klik Ubah Sumber Identitas.
7. Tunggu perubahan selesai, maka Anda akan melihat spanduk hijau di bagian atas.

Menyinkronkan pengguna dan grup ke Pusat Identitas

1. Di spanduk hijau klik Start Guided Setup (tombol di kanan atas)



2. Dalam Konfigurasi Pemetaan Atribut, klik Berikutnya
3. Di bagian Konfigurasi cakupan sinkronisasi, ketik nama pengguna yang ingin disinkronkan ke pusat identitas, lalu klik Tambah
4. Setelah selesai menambahkan pengguna dan grup, klik Berikutnya

Users | **Groups**

User

corp.pcluster.com ▼

Added users and groups (4)

<input type="checkbox"/>	Username / Group name	Type	Domain
<input type="checkbox"/>	user1	User	corp.pcluster.com
<input type="checkbox"/>	user2	User	corp.pcluster.com
<input type="checkbox"/>	admin1	User	corp.pcluster.com
<input type="checkbox"/>	admin2	User	corp.pcluster.com

5. Tinjau perubahan Anda, lalu klik Simpan konfigurasi
6. Jika Anda melihat peringatan di layar berikutnya tentang pengguna yang tidak disinkronkan, pilih tombol Lanjutkan sinkronisasi di kanan atas.
7. Selanjutnya, untuk mengaktifkan pengguna, Di tab Pengguna di sebelah kiri, pilih pengguna dan kemudian klik Aktifkan akses pengguna > Aktifkan akses pengguna

Catatan: Anda mungkin perlu memilih Lanjutkan sinkronisasi jika Anda memiliki spanduk peringatan di bagian atas dan kemudian menunggu pengguna untuk menyinkronkan (coba tombol segarkan untuk melihat apakah mereka disinkronkan belum).

IAM Identity Center ×

[IAM Identity Center](#) > [Users](#)

Users (4) Refresh Delete users Manage sync

Users listed here can sign in to the AWS access portal to access AWS accounts and assigned cloud applications. [Learn more](#)

Username < 1 > Settings

<input type="checkbox"/>	Username	Display name	Status	MFA device
<input type="checkbox"/>	user1@corp.pcluster.com	user1@corp.pcluster.com	Enabled	None
<input type="checkbox"/>	admin1@corp.pcluster.com	admin1@corp.pcluster.com	Enabled	None
<input type="checkbox"/>	user2@corp.pcluster.com	user2@corp.pcluster.com	Enabled	None
<input type="checkbox"/>	admin2@corp.pcluster.com	admin2@corp.pcluster.com	Enabled	None

Related consoles
IAM [IAM](#)

Menambahkan Aplikasi Anda ke Pusat Identitas IAM

Setelah Anda menyinkronkan pengguna Anda dengan IAM Identity Center, Anda perlu menambahkan aplikasi baru. Ini mengonfigurasi aplikasi berkemampuan SSO mana yang akan tersedia dari portal Pusat Identitas IAM Anda. Dalam hal ini, kami akan menambahkan AWS ParallelCluster UI sebagai aplikasi sementara IAM Identity Center akan menjadi penyedia identitas.

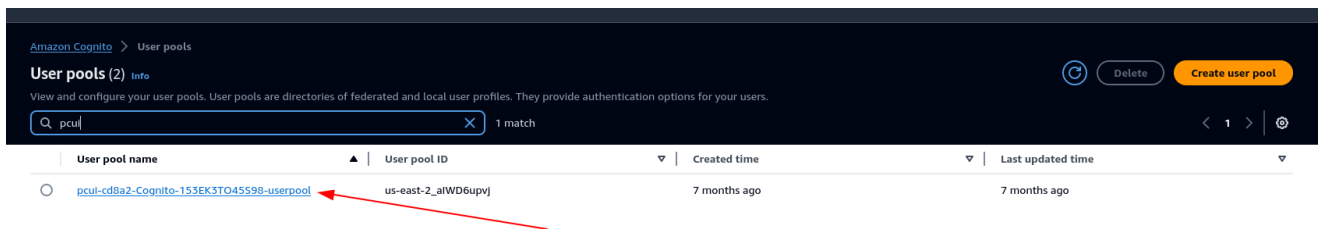
Langkah selanjutnya akan menambahkan AWS ParallelCluster UI sebagai aplikasi di IAM Identity Center. AWS ParallelCluster UI adalah portal web yang membantu pengguna mengelola cluster mereka. Untuk informasi selengkapnya lihat [AWS ParallelClusterUI](#).

Menyiapkan aplikasi di Pusat Identitas

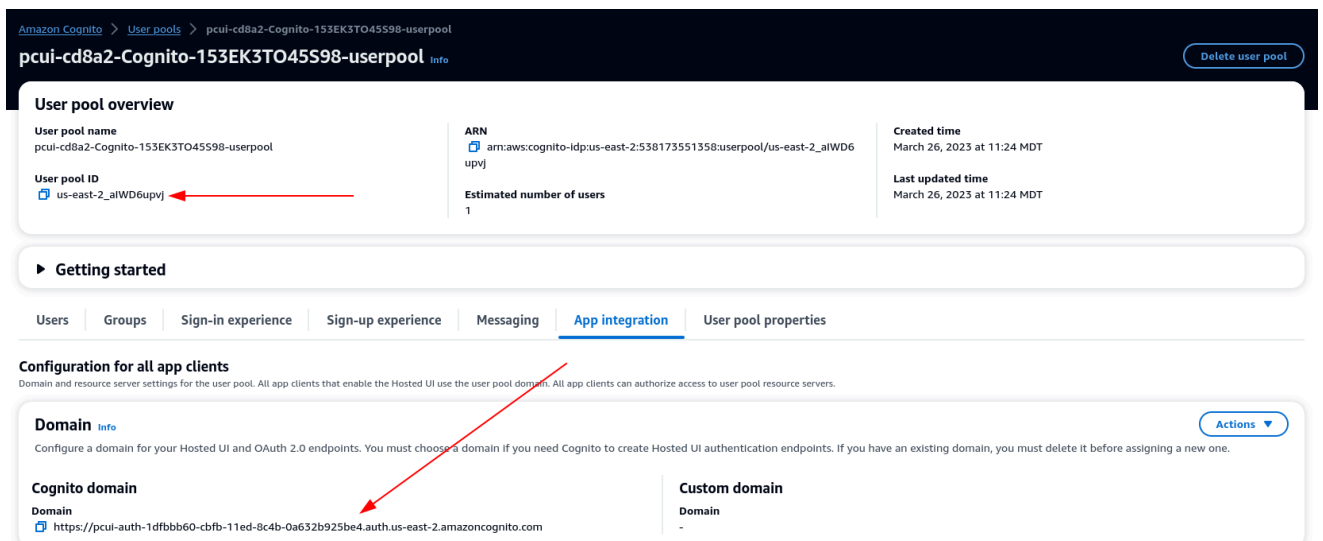
1. Di bawah Pusat Identitas IAM > Aplikasi (ditemukan di bilah menu kiri, klik Aplikasi)
2. Klik Tambah Aplikasi
3. Pilih Tambahkan aplikasi SAMP 2.0 khusus
4. Klik Berikutnya
5. Pilih nama tampilan dan deskripsi yang ingin Anda gunakan (misalnya PCUI dan AWS ParallelCluster UI)
6. Di bawah metadata IAM Identity Center, salin tautan untuk file metadata SAMP IAM Identity Center dan simpan untuk nanti, ini akan digunakan saat mengonfigurasi SSO di aplikasi web

7. Di bawah properti Aplikasi, di URL mulai Aplikasi, masukkan alamat PCUI Anda. Ini dapat ditemukan dengan pergi ke CloudFormation konsol, memilih tumpukan yang sesuai dengan PCUI (misalnya parallelcluster-ui) dan pergi ke tab Outputs untuk menemukan UIUrl ParallelCluster

misalnya <https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com>
8. Di bawah Metadata aplikasi, pilih Ketik nilai metadata Anda secara manual. Kemudian berikan nilai-nilai berikut.
 - a. Penting: Pastikan untuk mengganti nilai domain-prefix, region, dan userpool-id dengan informasi yang spesifik untuk lingkungan Anda.
 - b. Awalan domain, wilayah, dan userpool-id dapat diperoleh dengan membuka konsol Amazon Cognito > User pool



- c. Pilih kumpulan pengguna yang sesuai dengan PCUI (yang akan memiliki nama kumpulan Pengguna seperti PCUI-CD8A2-Cognito-153EK3to45S98-userPool)
- d. Arahkan ke Integrasi Aplikasi



9. <domain-prefix>URL Layanan Konsumen Pernyataan Aplikasi (ACS): <https://.auth.<region>.amazoncognito.com/saml2/idpresponse>

Aplikasi SAMP audiens: urn:amazon:cognito:sp: <userpool-id>

10. Pilih Kirim. Kemudian, buka halaman Detail untuk aplikasi yang Anda tambahkan.
11. Pilih daftar dropdown Tindakan dan pilih Edit pemetaan atribut. Kemudian, berikan atribut berikut.
 - a. Atribut pengguna dalam aplikasi: subjek (Catatan: subjek sudah diisi sebelumnya.) → Peta ke nilai string ini atau atribut pengguna di IAM Identity Center: \$ {user:email}, Format: EmailAddress
 - b. Atribut pengguna dalam aplikasi: email → Peta ke nilai string ini atau atribut pengguna di IAM Identity Center: \$ {user:email}, Format: unspecified

Attribute mappings for PCUI

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. [Learn more](#)

User attribute in the application	Maps to this string value or user attribute in IAM Identity Center	Format	
Subject	\$(user:email)	emailAddress	
email	\$(user:email)	unspecified	Remove
Add new attribute mapping			


Cancel Save changes

12. Simpan perubahan Anda.
13. Pilih tombol Tetapkan Pengguna dan kemudian tetapkan pengguna Anda ke aplikasi. Ini adalah pengguna di Direktori Aktif Anda yang akan memiliki akses ke antarmuka PCUI.

IAM Identity Center > Applications > PCUI

PCUI

Details Actions ▼

	Display name PCUI	Description AWS ParallelCluster UI
---	----------------------	---------------------------------------

Assigned users (1) Remove access Assign Users

The following users and groups from your connected directory can access this application. [Learn more](#)

Search for an assigned user or group

<input type="checkbox"/>	User/Group name	Type
<input type="checkbox"/>	admin1@corp.pcluster.com	User

Konfigurasi Pusat Identitas IAM sebagai IDP SAMP di kumpulan pengguna Anda

1. Di pengaturan kumpulan pengguna, pilih Pengalaman masuk > Tambahkan penyedia identitas

Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool

parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool Info

[Delete user pool](#)

User pool overview

User pool name parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool	ARN am:aws:cognito-idp:us-east-1:538173551358:userpool/us-east-1_Bgyu7LLz6	Created time November 5, 2023 at 12:32 MST
User pool ID us-east-1_Bgyu7LLz6	Estimated number of users 1	Last updated time November 5, 2023 at 12:32 MST

Getting started

[Users](#) | [Groups](#) | [Sign-in experience](#) | [Sign-up experience](#) | [Messaging](#) | [App integration](#) | [User pool properties](#)

Cognito user pool sign-in Info

Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.

Cognito user pool sign-in options
Email

Federated identity provider sign-in (0) Info

Your app users can sign-in through external social identity providers like Facebook, Google, Amazon, or Apple, and through your on-prem directories via SAML or Open ID Connect.

Identity provider	Identity provider type	Created time	Last updated time
No identity providers			

[Add identity provider](#)

[Delete](#) [Add identity provider](#) [View signing certificate](#)

2. Pilih SAMP iDP
3. Untuk nama Penyedia berikan IdentityCenter
4. Di bawah Sumber dokumen metadata pilih Masukkan URL titik akhir dokumen metadata dan berikan URL yang disalin selama penyiapan Aplikasi Pusat Identitas
5. Di bawah Atribut, untuk email pilih email

SAML

Configure a SAML 2.0 identity provider for your user pool.

Register your app with your SAML provider

To connect a SAML provider to Cognito, add your user pool as a relying party or application with your SAML 2.0 identity provider, and upload a metadata document to Cognito.

Set up SAML federation with this user pool

Provider name [Info](#)
Enter a friendly name for your SAML 2.0 identity provider.

Identifiers - optional [Info](#)
Enter identifiers for this provider. Identifiers can be used to redirect users to the correct IdP in multitenant apps.

Separate each identifier by a comma

Sign-out flow [Info](#)
 Add sign-out flow
Enable simultaneous sign-out from the SAML provider and Cognito.

Metadata document source [Info](#)
Provide a SAML metadata document. This document is issued by your SAML provider. It includes the issuer's name, expiration information, and keys that can be used to validate the response from the identity provider.

Upload metadata document
 Enter metadata document endpoint URL

Enter metadata document endpoint URL [Info](#)

Map attributes between your SAML provider and your user pool

Your required attributes are mapped to the equivalent SAML attributes. Each attribute you add must be mapped to a SAML attribute.

User pool attribute	SAML attribute
email	email

[Add another attribute](#)

[Cancel](#) [Add identity provider](#)

6. Pilih Tambahkan penyedia identitas.

Integrasikan iDP dengan klien aplikasi kumpulan pengguna

1. Selanjutnya, di bawah bagian Integrasi Aplikasi dari kumpulan pengguna Anda, pilih klien yang tercantum di bawah Daftar klien Aplikasi

Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool

parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool Info

[Delete user pool](#)

User pool overview

User pool name parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool	ARN arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_Bgyu7Lz6	Created time November 5, 2023 at 12:32 MST
User pool ID us-east-1_Bgyu7Lz6	Estimated number of users 1	Last updated time November 5, 2023 at 12:32 MST

Getting started

Users | Groups | Sign-in experience | Sign-up experience | Messaging | **App integration** | User pool properties

Configuration for all app clients

Domain and resource server settings for the user pool. All app clients that enable the Hosted UI use the user pool domain. All app clients can authorize access to user pool resource servers.

Domain Info

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. You must choose a domain if you need Cognito to create Hosted UI authentication endpoints. If you have an existing domain, you must delete it before assigning a new one.

[Actions](#)

Cognito domain	Custom domain
Domain https://pcul-auth-06f29200-7c12-11ee-b755-0e11297ecb0d.auth.us-east-1.amazoncognito.com	Domain -

Resource servers (0) Info

Configure resource servers. A resource server is a remote server that authorizes access based on OAuth 2.0 scopes in an access token.

[Edit](#) [Delete](#) [Create resource server](#)

Resource server name	Resource server identifier	Custom scopes
No resource servers		

[Create resource server](#)

App client defaults

Hosted UI customization and advanced security settings for the user pool. You can customize the Hosted UI and advanced security in app clients to override the defaults.

Hosted UI customization Info

Customize the hosted sign-up and sign-in pages to match your app's style and branding by uploading your own logo and customized CSS.

[Edit](#)

Logo -	Custom CSS -
------------------	------------------------

Advanced security Info

Configure advanced security features, including Cognito's automatic responses to suspicious user activity. Advanced security adds cost to your bill. [See pricing](#)

[Enable](#)

Status
 Disabled

App client list

The app clients that integrate your apps with your user pool. Configure client overrides to user pool default configurations, and configure Amazon Pinpoint analytics.

App clients and analytics (1) Info

Configure an app client. App clients are the user pool authentication resources attached to your app. Select an app client to configure the permitted authentication actions for an app.

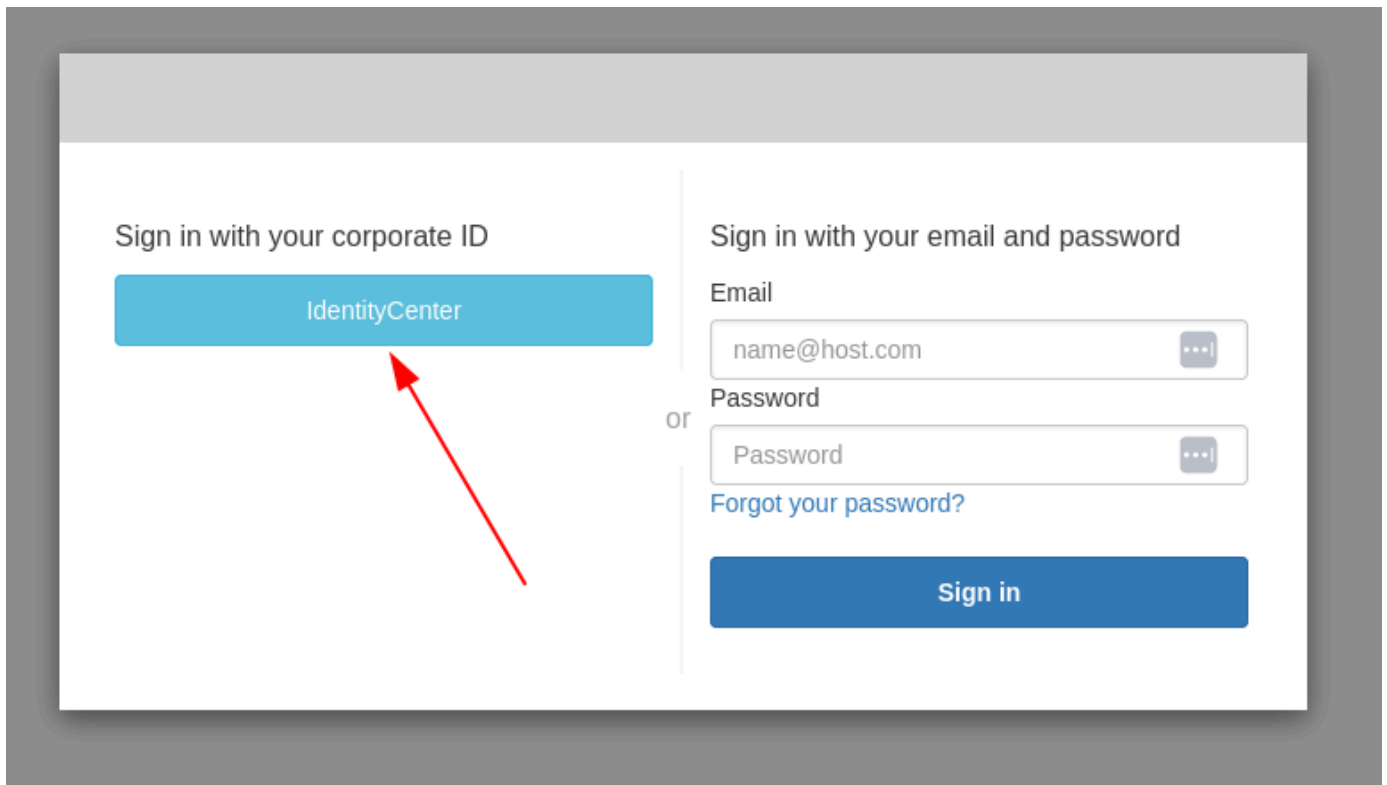
[Refresh](#) [Delete](#) [Create app client](#)

App client name	Client ID
<input type="radio"/> CognitoAppClient-ETqXbqL5wRVs	...

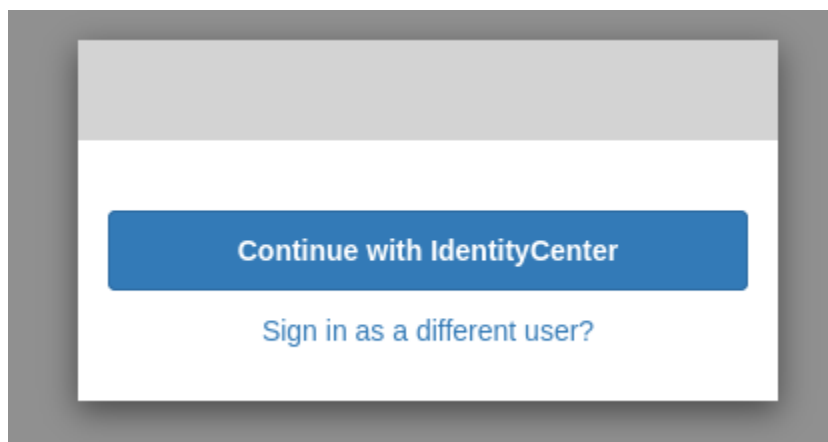
2. Di bawah UI yang Dihosting pilih Edit
3. Di bawah penyedia Identitas memilih IdentityCenterjuga.
4. Pilih Save changes (Simpan perubahan)

Validasi penyiapan Anda

1. Selanjutnya kita akan memvalidasi pengaturan yang baru saja kita buat dengan masuk ke PCUI. Masuk ke portal PCUI Anda dan sekarang Anda akan melihat opsi untuk masuk dengan ID Perusahaan Anda:



2. Mengklik IdentityCentertombol akan membawa Anda ke IAM Identity Center iDP login diikuti oleh halaman dengan aplikasi Anda di dalamnya yang mencakup PCUI, buka aplikasi itu.
3. Setelah Anda masuk ke layar berikut, pengguna Anda akan ditambahkan ke kumpulan pengguna Cognito.



Jadikan pengguna Anda administrator

1. Sekarang navigasikan ke konsol Amazon Cognito > User pool dan pilih pengguna yang baru dibuat yang seharusnya memiliki awalan identitycenter

The screenshot shows the Amazon Cognito console for a user pool named 'parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool'. The 'Users (2)' section is active, displaying a table of users. A red arrow points to the user 'identitycenter_admin1@corp.pcluster.c...'.

User name	Email address	Email verified	Confirmation status	Status
c8f7eccc-e647-4227-afa2-080274ebfefe	user@amazon.com	Yes	Confirmed	Enabled
identitycenter_admin1@corp.pcluster.c...	admin1@corp.pcluster.com	No	External provider	Enabled

2. Di bawah Keanggotaan grup pilih Tambahkan pengguna ke grup, pilih admin dan klik Tambah.
3. Sekarang ketika Anda mengklik Lanjutkan dengan IdentityCenter Anda akan dinavigasi ke halaman AWS ParallelCluster UI.

Memecahkan Masalah AWS ParallelCluster

AWS ParallelCluster [Komunitas memelihara halaman Wiki yang menyediakan banyak tips pemecahan masalah di Wiki. AWS ParallelCluster GitHub](#) Untuk daftar masalah yang diketahui, lihat [Masalah yang diketahui](#).

Topik

- [Mencoba membuat cluster](#)
- [Mencoba menjalankan pekerjaan](#)
- [Mencoba memperbarui cluster](#)
- [Mencoba mengakses penyimpanan](#)
- [Mencoba menghapus cluster](#)
- [Mencoba memutakhirkan tumpukan AWS ParallelCluster API](#)
- [Melihat kesalahan dalam inialisasi node komputasi](#)
- [Pemecahan masalah metrik metrik metrik metrik metrik Kondisi Kondisi](#)
- [Memecahkan masalah penerapan klaster](#)
- [Memecahkan masalah penskalaan](#)
- [Grup penempatan dan masalah peluncuran instance](#)
- [Direktori yang tidak dapat diganti](#)
- [Memecahkan masalah di NICE DCV](#)
- [Memecahkan masalah dalam cluster dengan integrasi AWS Batch](#)
- [Memecahkan masalah integrasi multi-pengguna dengan Active Directory](#)
- [Memecahkan masalah AMI khusus](#)
- [Memecahkan masalah batas waktu pembaruan klaster saat cfn-hup tidak berjalan](#)
- [Pemecahan masalah jaringan](#)
- [Pembaruan cluster gagal pada tindakan onNodeUpdated kustom](#)
- [Melihat kesalahan dengan Slurm konfigurasi khusus](#)
- [Alarm Cluster](#)
- [Dukungan tambahan](#)

Mencoba membuat cluster

Saat menggunakan AWS ParallelCluster versi 3.5.0 dan yang lebih baru untuk membuat cluster, dan pembuatan cluster gagal dengan `--rollback-on-failure` set to `false`, gunakan perintah [`pcluster describe-cluster`](#) CLI untuk mendapatkan informasi status dan kegagalan. Dalam hal ini, `pcluster describe-cluster` output `clusterStatus` yang diharapkan adalah `CREATE_FAILED`. Periksa `failures` bagian dalam output untuk menemukan `failureCode` dan `failureReason`. Kemudian, di bagian berikut, temukan pencocokan `failureCode` untuk bantuan pemecahan masalah tambahan. Untuk informasi selengkapnya, lihat [`pcluster describe-cluster`](#).

Di bagian berikut, kami sarankan Anda memeriksa log pada node kepala, seperti `/var/log/chef-client.log` file `/var/log/cfn-init.log` dan. Untuk informasi selengkapnya tentang AWS ParallelCluster log dan cara melihatnya, lihat [Log kunci untuk debugging](#) dan [Mengambil dan melestarikan log](#).

Jika Anda tidak memiliki `failureCode`, navigasikan ke AWS CloudFormation konsol untuk melihat tumpukan cluster. Periksa `Status Reason` untuk `HeadNodeWaitCondition` atau kegagalan pada sumber daya lain untuk menemukan rincian kegagalan tambahan. Untuk informasi selengkapnya, lihat [Lihat AWS CloudFormation acara di CREATE_FAILED](#). Periksa `/var/log/chef-client.log` file `/var/log/cfn-init.log` dan pada node kepala.

failureCode adalah **OnNodeConfiguredExecutionFailure**

- Mengapa gagal?

Anda menyediakan skrip kustom di `OnNodeConfigured` bagian simpul kepala dalam konfigurasi untuk membuat cluster. Namun, skrip khusus gagal dijalankan.

- Bagaimana cara mengatasinya?

Periksa `/var/log/cfn-init.log` file untuk mempelajari lebih lanjut tentang kegagalan dan cara memperbaiki masalah dalam skrip kustom Anda. Menjelang akhir log ini, Anda mungkin melihat menjalankan informasi yang terkait dengan `OnNodeConfigured` skrip setelah `Running` command `runpostinstall` pesan.

failureCode adalah **OnNodeConfiguredDownloadFailure**

- Mengapa gagal?

Anda menyediakan skrip kustom di `OnNodeConfigured` bagian simpul kepala dalam konfigurasi untuk membuat cluster. Namun, skrip khusus gagal diunduh.

- Bagaimana cara mengatasinya?

Pastikan URL valid dan akses dikonfigurasi dengan benar. Untuk informasi lebih lanjut tentang konfigurasi skrip bootstrap kustom, lihat [Tindakan bootstrap kustom](#).

Periksa `/var/log/cfn-init.log` file. Menjelang akhir log ini, Anda mungkin melihat menjalankan informasi yang terkait dengan pemrosesan `OnNodeConfigured` skrip, termasuk mengunduh, setelah `Running command runpostinstall` pesan.

failureCode adalah **OnNodeConfiguredFailure**

- Mengapa gagal?

Anda menyediakan skrip kustom di `OnNodeConfigured` bagian simpul kepala dalam konfigurasi untuk membuat cluster. Namun, penggunaan skrip khusus gagal dalam penerapan cluster. Penyebab langsung tidak dapat ditentukan dan penyelidikan tambahan diperlukan.

- Bagaimana cara mengatasinya?

Periksa `/var/log/cfn-init.log` file. Menjelang akhir log ini, Anda mungkin melihat menjalankan informasi yang terkait dengan pemrosesan `OnNodeConfigured` skrip setelah `Running command runpostinstall` pesan.

failureCode adalah **OnNodeStartExecutionFailure**

- Mengapa gagal?

Anda menyediakan skrip kustom di `OnNodeStart` bagian simpul kepala dalam konfigurasi untuk membuat cluster. Namun, skrip khusus gagal dijalankan.

- Bagaimana cara mengatasinya?

Periksa `/var/log/cfn-init.log` file untuk mempelajari lebih lanjut tentang kegagalan dan cara memperbaiki masalah dalam skrip kustom Anda. Menjelang akhir log ini, Anda mungkin melihat menjalankan informasi yang terkait dengan `OnNodeStart` skrip setelah `Running command runpreinstall` pesan.

failureCode adalah **OnNodeStartDownloadFailure**

- Mengapa gagal?

Anda menyediakan skrip kustom di `OnNodeStart` bagian simpul kepala dalam konfigurasi untuk membuat cluster. Namun, skrip khusus gagal diunduh.

- Bagaimana cara mengatasinya?

Pastikan URL valid dan akses dikonfigurasi dengan benar. Untuk informasi lebih lanjut tentang konfigurasi skrip bootstrap kustom, lihat [Tindakan bootstrap kustom](#).

Periksa `/var/log/cfn-init.log` file. Menjelang akhir log ini, Anda mungkin melihat menjalankan informasi yang terkait dengan pemrosesan `OnNodeStart` skrip, termasuk mengunduh, setelah `Running command runpreinstall` pesan.

failureCode adalah **OnNodeStartFailure**

- Mengapa gagal?

Anda menyediakan skrip kustom di `OnNodeStart` bagian node kepala dalam konfigurasi untuk membuat cluster. Namun, penggunaan skrip khusus gagal dalam penerapan cluster. Penyebab langsung tidak dapat ditentukan dan penyelidikan tambahan diperlukan.

- Bagaimana cara mengatasinya?

Periksa `/var/log/cfn-init.log` file. Menjelang akhir log ini, Anda mungkin melihat menjalankan informasi yang terkait dengan pemrosesan `OnNodeStart` skrip setelah `Running command runpreinstall` pesan.

failureCode adalah **EbsMountFailure**

- Mengapa gagal?

Volume EBS yang ditentukan dalam konfigurasi cluster gagal dipasang.

- Bagaimana cara mengatasinya?

Periksa `/var/log/chef-client.log` file untuk rincian kegagalan.

failureCode adalah **EfsMountFailure**

- Mengapa gagal?

Volume Amazon EFS yang ditentukan dalam konfigurasi cluster gagal dipasang.

- Bagaimana cara mengatasinya?

Jika Anda mendefinisikan sistem file Amazon EFS yang ada, pastikan lalu lintas diizinkan antara cluster dan sistem file. Untuk informasi lebih lanjut, lihat [SharedStorage/EfsSettings/FileSystemId](#).

Periksa `/var/log/chef-client.log` file untuk rincian kegagalan.

failureCode adalah **FsxMountFailure**

- Mengapa gagal?

Sistem file Amazon FSx yang ditentukan dalam konfigurasi cluster gagal dipasang.

- Bagaimana cara mengatasinya?

Jika Anda mendefinisikan sistem file Amazon FSx yang ada, pastikan lalu lintas diizinkan antara cluster dan sistem file. Untuk informasi lebih lanjut, lihat [SharedStorage/FsxLustreSettings/FileSystemId](#).

Periksa `/var/log/chef-client.log` file untuk rincian kegagalan.

failureCode adalah **RaidMountFailure**

- Mengapa gagal?

Volume RAID yang ditentukan dalam konfigurasi cluster gagal dipasang.

- Bagaimana cara mengatasinya?

Periksa `/var/log/chef-client.log` file untuk rincian kegagalan.

failureCode adalah **AmiVersionMismatch**

- Mengapa gagal?

AWS ParallelClusterVersi yang digunakan untuk membuat AMI kustom berbeda dari AWS ParallelCluster versi yang digunakan untuk mengkonfigurasi cluster. Di CloudFormation konsol, lihat detail CloudFormation tumpukan cluster dan periksa Status Reason HeadNodeWaitCondition untuk mendapatkan detail tambahan tentang AWS ParallelCluster versi dan AMI. Untuk informasi selengkapnya, lihat [Lihat AWS CloudFormation acara di CREATE_FAILED](#).

- Bagaimana cara mengatasinya?

Pastikan AWS ParallelCluster versi yang digunakan untuk membuat AMI kustom adalah AWS ParallelCluster versi yang sama yang digunakan untuk mengkonfigurasi cluster. Anda dapat mengubah versi AMI kustom atau versi `pcluster` CLI untuk membuatnya sama.

failureCode adalah **InvalidAmi**

- Mengapa gagal?

AMI kustom tidak valid karena tidak dibuat menggunakan AWS ParallelCluster

- Bagaimana cara mengatasinya?

Gunakan `pcluster build-image` perintah untuk membuat AMI dengan menjadikan AMI Anda sebagai gambar induk. Untuk informasi selengkapnya, lihat [pcluster build-image](#).

failureCode adalah **HeadNodeBootstrapFailure** dengan **failureReason** Gagal mengatur simpul kepala.

- Mengapa gagal?

Penyebab langsung tidak dapat ditentukan dan penyelidikan tambahan diperlukan. Misalnya, bisa jadi cluster dalam status dilindungi, dan ini bisa disebabkan oleh kegagalan untuk menyediakan armada komputasi statis.

- Bagaimana cara mengatasinya?

Periksa `/var/log/chef-client.log` file untuk rincian kegagalan.

Note

Jika Anda melihat `RuntimeError` pengecualian `Cluster state has been set to PROTECTED mode due to failures detected in static node provisioning`, kluster dalam status dilindungi. Untuk informasi selengkapnya, lihat [Cara men-debug mode terlindungi](#).

failureCodeHeadNodeBootstrapFailure dengan waktu pembuatan **failureReason** Cluster habis.

- Mengapa gagal?

Secara default, ada batas waktu 30 menit untuk menyelesaikan pembuatan cluster. Jika pembuatan kluster belum selesai dalam jangka waktu ini, pembuatan kluster gagal dengan kesalahan batas waktu. Pembuatan cluster dapat batas waktu karena berbagai alasan. Misalnya, kegagalan batas waktu dapat disebabkan oleh kegagalan pembuatan node kepala, masalah jaringan, skrip khusus yang terlalu lama untuk dijalankan di node kepala, kesalahan dalam skrip khusus yang berjalan di node komputasi, atau waktu tunggu yang lama untuk penyediaan node komputasi. Penyebab langsung tidak dapat ditentukan dan penyelidikan tambahan diperlukan.

- Bagaimana cara mengatasinya?

Periksa `/var/log/chef-client.log` file `/var/log/cfn-init.log` dan untuk rincian kegagalan. Untuk informasi lebih lanjut tentang AWS ParallelCluster log dan cara mendapatkannya, lihat [Log kunci untuk debugging](#) dan [Mengambil dan melestarikan log](#).

Anda mungkin menemukan yang berikut di log ini.

- Melihat **Waiting for static fleet capacity provisioning** dekat akhir **chef-client.log**

Ini menunjukkan bahwa waktu pembuatan cluster habis saat menunggu node statis dinyalakan. Untuk informasi selengkapnya, lihat [Melihat kesalahan dalam inisialisasi node komputasi](#).

- Melihat **OnNodeConfigured** atau skrip node **OnNodeStart** kepala belum selesai di akhir **cfn-init.log**

Ini menunjukkan bahwa skrip `OnNodeConfigured` atau `OnNodeStart` kustom membutuhkan waktu lama untuk dijalankan dan menyebabkan kesalahan batas waktu. Periksa skrip kustom Anda untuk masalah yang mungkin menyebabkannya berjalan untuk waktu yang lama. Jika skrip kustom Anda memerlukan waktu yang lama untuk dijalankan, pertimbangkan untuk mengubah batas waktu dengan menambahkan `DevSettings` bagian ke file konfigurasi cluster Anda, seperti yang ditunjukkan pada contoh berikut:

```
DevSettings:
  Timeouts:
    HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- Tidak dapat menemukan log, atau node kepala tidak berhasil dibuat

Ada kemungkinan bahwa node kepala tidak berhasil dibuat dan log tidak dapat ditemukan. Di CloudFormation konsol, lihat detail tumpukan cluster untuk memeriksa detail kegagalan tambahan.

failureCodeHeadNodeBootstrapFailed dengan **failureReason** Gagal mem-bootstrap simpul kepala.

- Mengapa gagal?

Penyebab langsung tidak dapat ditentukan dan penyelidikan tambahan diperlukan.

- Bagaimana cara mengatasinya?

Periksa `/var/log/chef-client.log` file `/var/log/cfn-init.log` dan.

failureCode adalah **ResourceCreationFailure**

- Mengapa gagal?

Penciptaan beberapa sumber daya gagal selama proses pembuatan cluster. Kegagalan dapat terjadi karena berbagai alasan. Misalnya, kegagalan pembuatan sumber daya dapat disebabkan oleh masalah kapasitas atau kebijakan IAM yang salah konfigurasi.

- Bagaimana cara mengatasinya?

Di CloudFormation konsol, lihat tumpukan cluster untuk memeriksa detail kegagalan pembuatan sumber daya tambahan.

failureCode adalah **ClusterCreationFailure**

- Mengapa gagal?

Penyebab langsung tidak dapat ditentukan dan penyelidikan tambahan diperlukan.

- Bagaimana cara mengatasinya?

Di CloudFormation konsol, lihat tumpukan cluster dan periksa Status Reason HeadNodeWaitCondition untuk menemukan rincian kegagalan tambahan.

Periksa `/var/log/chef-client.log` file `/var/log/cfn-init.log` dan.

Melihat **WaitCondition timed out...** di CloudFormation tumpukan

Untuk informasi selengkapnya, lihat [failureCodeHeadNodeBootstrapFailed dengan waktu pembuatan failureReason Cluster habis..](#)

Melihat **Resource creation cancelled** di CloudFormation tumpukan

Untuk informasi selengkapnya, lihat [failureCode adalah ResourceCreationFailure.](#)

Melihat **Failed to run cfn-init...** atau kesalahan lain di AWS CloudFormation tumpukan

Periksa `/var/log/cfn-init.log` dan `/var/log/chef-client.log` untuk rincian kegagalan tambahan.

Melihat **chef-client.log** berakhir dengan **INFO: Waiting for static fleet capacity provisioning**

Ini terkait dengan batas waktu pembuatan cluster saat menunggu node statis dinyalakan. Untuk informasi selengkapnya, lihat [Melihat kesalahan dalam inisialisasi node komputasi.](#)

Melihat **Failed to run preinstall or postinstall in cfn-init.log**

Anda memiliki OnNodeStart skrip OnNodeConfigured atau di HeadNode bagian konfigurasi cluster. Script tidak berfungsi dengan benar. Periksa `/var/log/cfn-init.log` file untuk detail kesalahan skrip kustom.

Melihat **This AMI was created with xxx, but is trying to be used with xxx...** di CloudFormation tumpukan

Untuk informasi selengkapnya, lihat [failureCodeadalah AmiVersionMismatch](#).

Melihat **This AMI was not baked by AWS ParallelCluster...** di CloudFormation tumpukan

Untuk informasi selengkapnya, lihat [failureCodeadalah InvalidAmi](#).

Melihat **pcluster create-cluster** perintah gagal dijalankan secara lokal

Periksa `~/.parallelcluster/pcluster-cli.log` di sistem file lokal Anda untuk rincian kegagalan.

Dukungan tambahan

Ikuti panduan pemecahan masalah di [Memecahkan masalah penerapan klaster](#)

Periksa untuk melihat apakah skenario Anda tercakup dalam [Masalah GitHub yang Diketahui](#) di AWS ParallelCluster on GitHub.

Untuk dukungan tambahan, lihat [Dukungan tambahan](#).

Mencoba menjalankan pekerjaan

srun pekerjaan interaktif gagal dengan kesalahan **srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf**

- Mengapa gagal?

Anda menjalankan `srun` perintah untuk mengirimkan pekerjaan, dan kemudian Anda meningkatkan ukuran antrian dengan menggunakan `pcluster update-cluster` perintah tanpa memulai ulang Slurm daemon setelah pembaruan selesai.

Slurm mengatur Slurm daemon dalam hierarki pohon untuk mengoptimalkan komunikasi. Hirarki ini hanya diperbarui saat daemon dimulai.

Misalkan Anda menggunakan `srun` untuk meluncurkan pekerjaan dan kemudian menjalankan `pcluster update-cluster` perintah untuk meningkatkan ukuran antrian. Node komputasi baru diluncurkan sebagai bagian dari pembaruan. Kemudian, Slurm antrian pekerjaan Anda ke salah satu node komputasi baru. Dalam hal ini, baik Slurm daemon dan `srun` tidak mendeteksi node komputasi baru. `srun` mengembalikan kesalahan karena tidak mendeteksi node baru.

- Bagaimana cara mengatasinya?

Mulai ulang Slurm daemon pada semua node komputasi, dan kemudian gunakan `srun` untuk mengirimkan pekerjaan Anda. Anda dapat menjadwalkan Slurm daemon restart dengan menjalankan `scontrol reboot` perintah yang memulai ulang node komputasi. Untuk informasi selengkapnya, lihat [scontrol reboot](#) di Slurm dokumentasi. Anda juga dapat me-restart Slurm daemon secara manual pada node komputasi dengan meminta restart layanan terkait. `systemd`

Job terjebak dalam CF keadaan dengan **squeue** perintah

Ini mungkin menjadi masalah dengan node dinamis yang menyala. Untuk informasi selengkapnya, lihat [Melihat kesalahan dalam inisialisasi node komputasi](#).

Menjalankan pekerjaan skala besar dan melihat **nfsd: too many open connections, consider increasing the number of threads in /var/log/messages**

Dengan sistem file jaringan, ketika batas jaringan tercapai, waktu tunggu I/O juga meningkat. Ini dapat menghasilkan penguncian lunak karena jaringan digunakan untuk menulis data untuk jaringan dan metrik I/O.

Dengan instance generasi ke-5, kami menggunakan driver ENA untuk mengekspos penghitung paket. Penghitung ini menghitung paket yang dibentuk oleh AWS ketika jaringan mencapai batas bandwidth instance. Anda dapat memeriksa penghitung ini untuk melihat apakah mereka lebih besar dari 0. Jika ya, maka Anda telah melampaui batas bandwidth Anda. Anda dapat melihat penghitung ini dengan menjalankan `ethtool -S eth0 | grep exceeded`.

Melebihi batas jaringan seringkali merupakan hasil dari mendukung terlalu banyak koneksi NFS. Ini adalah salah satu hal pertama yang harus diperiksa ketika Anda mencapai atau melampaui batas jaringan.

Misalnya, output berikut menunjukkan paket yang dijatuhkan:

```
$ ethtool -S eth0 | grep exceeded
bw_in_allowance_exceeded: 38750610
bw_out_allowance_exceeded: 1165693
pps_allowance_exceeded: 103
contrack_allowance_exceeded: 0
linklocal_allowance_exceeded: 0
```

Untuk menghindari pesan ini, pertimbangkan untuk mengubah tipe instance node head ke tipe instance yang lebih berkinerja. Pertimbangkan untuk memindahkan penyimpanan data Anda ke sistem file penyimpanan bersama yang tidak diekspor sebagai berbagi NFS, seperti Amazon EFS atau Amazon FSx. Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#) dan [Praktik Terbaik](#) di AWS ParallelCluster Wiki di GitHub.

Menjalankan pekerjaan MPI

Mengaktifkan mode debug

Untuk mengaktifkan mode debug OpenMpi, [lihat Kontrol apa yang dimiliki Open MPI](#) yang membantu dalam debugging.

Untuk mengaktifkan mode debug IntelMpi, lihat Variabel Lingkungan [Lainnya](#).

Melihat **MPI_ERRORS_ARE_FATAL** dan **OPAL ERROR** dalam output pekerjaan

Kode kesalahan ini berasal dari lapisan MPI dalam aplikasi Anda. Untuk mempelajari cara mendapatkan log debug MPI dari aplikasi Anda, lihat [Mengaktifkan mode debug](#)

Kemungkinan penyebab kesalahan ini adalah aplikasi Anda telah dikompilasi untuk implementasi MPI tertentu, seperti OpenMPI, dan Anda mencoba menjalankannya dengan implementasi MPI yang berbeda, seperti IntelMPI. Pastikan Anda berdua mengkompilasi dan menjalankan aplikasi Anda dengan implementasi MPI yang sama.

Menggunakan **mpirun** dengan DNS terkelola dinonaktifkan

Untuk cluster yang dibuat [SlurmSettingsdengan/Dns/DisableManagedDns](#) dan [UseEc2Hostnames](#) disetel ke `true`, nama Slurm node tidak diselesaikan oleh DNS. Slurmdapat mem-bootstrap proses MPI saat nodenames tidak diaktifkan dan jika pekerjaan MPI dijalankan dalam suatu Slurm konteks. Kami merekomendasikan mengikuti panduan dalam [Panduan Pengguna Slurm MPI](#) untuk menjalankan pekerjaan MPI dengan Slurm

Mencoba memperbarui cluster

pcluster update-cluster perintah gagal dijalankan secara lokal

Periksa `~/.parallelcluster/pcluster-cli.log` di sistem file lokal Anda untuk rincian kegagalan.

Melihat **clusterStatus** adalah **UPDATE_FAILED** dengan **pcluster describe-cluster** perintah

Jika pembaruan tumpukan cluster diputar kembali, periksa `/var/log/chef-client.logs` file untuk detail kesalahan.

Periksa untuk melihat apakah masalah Anda disebutkan dalam [Masalah yang GitHub Diketahui](#) di AWS ParallelCluster on GitHub.

Waktu pembaruan cluster habis

Ini bisa menjadi masalah yang terkait dengan `cf-n-hup` tidak berjalan. Jika `cf-n-hup` iblis diakhiri oleh penyebab eksternal, itu tidak dimulai ulang secara otomatis. Jika `cf-n-hup` tidak berjalan,

selama pembaruan cluster, CloudFormation tumpukan memulai proses pembaruan seperti yang diharapkan, tetapi prosedur pembaruan tidak diaktifkan pada node kepala dan penerapan tumpukan akhirnya habis waktu. Untuk informasi selengkapnya, lihat [Memecahkan masalah batas waktu pembaruan klaster saat cfn-hup tidak berjalan](#) untuk memecahkan masalah dan memulihkan masalah.

Mencoba mengakses penyimpanan

Menggunakan sistem file Amazon FSx for Lustre eksternal

Pastikan lalu lintas diperbolehkan antara cluster dan sistem file. Sistem file harus dikaitkan dengan grup keamanan yang memungkinkan lalu lintas TCP masuk dan keluar melalui port 988, 1021, 1022, dan 1023. Untuk informasi selengkapnya tentang cara mengatur grup keamanan, lihat [FileSystemId](#).

Menggunakan sistem file Amazon Elastic File System eksternal

Pastikan lalu lintas diperbolehkan antara cluster dan sistem file. Sistem file harus dikaitkan dengan grup keamanan yang memungkinkan lalu lintas TCP masuk dan keluar melalui port 988, 1021, 1022, dan 1023. Untuk informasi selengkapnya tentang cara mengatur grup keamanan, lihat [FileSystemId](#).

Mencoba menghapus cluster

pcluster delete-clusterPerintah gagal dijalankan secara lokal

Periksa `~/ .parallelcluster/pcluster-cli.log` file di sistem file lokal Anda.

Tumpukan cluster gagal dihapus

Jika tumpukan cluster gagal dihapus, periksa pesan peristiwa CloudFormation tumpukan.

Periksa apakah masalah Anda disebutkan dalam [Masalah yang GitHub Diketahui](#) saat AWS ParallelCluster aktif GitHub.

Mencoba memutakhirkan tumpukan AWS ParallelCluster API

Periksa apakah masalah Anda disebutkan dalam [Masalah yang GitHub Diketahui](#) saat AWS ParallelCluster aktif GitHub.

Melihat kesalahan dalam inisialisasi node komputasi

Melihat **Node bootstrap error** di **clustermgtd.log**

Masalahnya terkait dengan node komputasi yang gagal bootstrap. Untuk informasi tentang cara men-debug masalah mode yang dilindungi klaster, lihat [Cara men-debug mode terlindungi](#).

Saya mengonfigurasi reservasi kapasitas permintaan (ODCR) atau Instans Cadangan zona

ODCR yang menyertakan instance yang memiliki beberapa antarmuka jaringan, seperti P4d, P4de, dan Trainium (Trn) AWS

Dalam file konfigurasi cluster, periksa apakah HeadNode ada di subnet publik dan node komputasi berada dalam subnet pribadi.

ODCR ditargetkan ODCRS

Melihat **Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'**. meskipun saya sudah memiliki **/opt/slurm/etc/pcluster/run_instances_overrides.json** tempat dengan mengikuti instruksi yang diberikan di [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#)

Jika Anda menggunakan AWS ParallelCluster versi 3.1.1 hingga 3.2.1 dengan ODCR yang ditargetkan, dan Anda juga menggunakan [instance run override file JSON, mungkin saja Anda tidak memiliki file JSON](#) yang diformat dengan benar. Anda bisa melihat kesalahan di `clustermgtd.log`, seperti berikut ini:

```
Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'.
Using default: {} in /var/log/parallelcluster/clustermgtd.
```

Validasi bahwa format file JSON sudah benar dengan menjalankan yang berikut ini:

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

Melihat **Found RunInstances parameters override.clustermgtd.log** kapan pembuatan kluster gagal, atau **slurm_resume.log** saat menjalankan pekerjaan gagal

Jika Anda menggunakan [instance run override file JSON](#), periksa apakah Anda benar mengatur nama antrian dan nama sumber daya komputasi dalam file. `/opt/slurm/etc/pcluster/run_instances_overrides.json`

Melihat **An error occurred (InsufficientInstanceCapacity)slurm_resume.log** ketika saya gagal menjalankan pekerjaan, atau **clustermgtd.log** ketika saya gagal membuat cluster

Menggunakan PG-ODCR (Placement Group ODCR)

Saat membuat ODCR dengan grup penempatan terkait, nama grup penempatan yang sama harus digunakan dalam file konfigurasi. Tetapkan [nama grup penempatan](#) yang sesuai dalam konfigurasi cluster.

Menggunakan Instans Cadangan Zonal

Jika Anda menggunakan Instans Cadangan zona dengan `PlacementGroup/Enabledto true` dalam konfigurasi cluster, Anda mungkin melihat kesalahan, seperti berikut ini:

```
We currently do not have sufficient trn1.32xlarge capacity in the Availability Zone
you requested (us-east-1d). Our system will be working on provisioning additional
capacity.
You can currently get trn1.32xlarge capacity by not specifying an Availability Zone in
your request or choosing us-east-1a, us-east-1b, us-east-1c, us-east-1e, us-east-1f.
```

Anda mungkin melihat ini karena Instans Cadangan zona tidak ditempatkan di UC (atau tulang belakang) yang sama, yang dapat menyebabkan kesalahan kapasitas (ICE) tidak mencukupi saat menggunakan grup penempatan. Anda dapat memeriksa kasus ini dengan menonaktifkan pengaturan `PlacementGroup` dalam konfigurasi cluster untuk menentukan apakah cluster dapat mengalokasikan instance.

Melihat **An error occurred**

(VcpuLimitExceeded)slurm_resume.log ketika saya gagal menjalankan pekerjaan, atau masuk **clustermgtd.log**, ketika saya gagal membuat cluster

Periksa batas vCPU pada akun Anda untuk jenis instans EC2 tertentu yang Anda gunakan. Jika Anda melihat nol atau lebih sedikit vCPU daripada yang Anda minta, mintalah peningkatan batas Anda.

Untuk informasi tentang cara melihat batas saat ini dan meminta batas baru, lihat [kuota layanan Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Melihat **An error occurred**

(InsufficientInstanceCapacity)slurm_resume.log ketika saya gagal menjalankan pekerjaan, atau masuk **clustermgtd.log**, ketika saya gagal membuat cluster

Anda mengalami masalah kapasitas yang tidak mencukupi. Ikuti <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors> untuk memecahkan masalah.

Melihat node dalam **DOWN** keadaan dengan **Reason (Code:InsufficientInstanceCapacity)...**

Anda mengalami masalah kapasitas yang tidak mencukupi. Ikuti <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors> untuk memecahkan masalah. Untuk informasi selengkapnya AWS ParallelCluster tentang mode fail-over kapasitas cepat tidak mencukupi, lihat. [Slurmcluster cepat kapasitas tidak mencukupi fail-over](#)

Melihat **cannot change locale (en_US.utf-8) because it has an invalid name** di **slurm_resume.log**

Ini dapat terjadi jika Anda memiliki proses yum instalasi yang gagal yang meninggalkan pengaturan lokal dalam keadaan tidak konsisten. Misalnya, ini dapat disebabkan ketika pengguna menghentikan proses penginstalan.

Untuk memverifikasi penyebabnya, lakukan tindakan berikut:

- Jalankan `su - pcluster-admin`.

Shell menunjukkan kesalahan, seperti, `cannot change locale...no such file or directory`.

- Jalankan `localedef --list`.

Mengembalikan daftar kosong atau tidak berisi lokal default.

- Periksa yum perintah terakhir dengan `yum history` dan `yum history info #ID`. Apakah ID terakhir memiliki `Return-Code: Success?`

Jika ID terakhir tidak memiliki `Return-Code: Success`, skrip pasca-instal mungkin tidak berhasil berjalan.

Untuk memperbaiki masalah, coba bangun kembali lokal dengan `yum reinstall glibc-all-langpacks`. Setelah pembangunan kembali, `su - pcluster-admin` tidak menampilkan kesalahan atau peringatan jika masalah telah diperbaiki.

Tak satu pun dari skenario sebelumnya berlaku untuk situasi saya

Untuk memecahkan masalah inisialisasi node komputasi, lihat [Memecahkan masalah inisialisasi node](#)

Periksa untuk melihat apakah skenario Anda tercakup dalam [Masalah GitHub yang Diketahui](#) di AWS ParallelCluster on GitHub.

Untuk dukungan tambahan, lihat [Dukungan tambahan](#).

Pemecahan masalah metrik metrik metrik metrik metrik Kondisi Kondisi

Metrik kesehatan kluster ditambahkan ke CloudWatch dasbor AWS ParallelCluster Amazon dimulai dengan AWS ParallelCluster versi 3.6.0. Di bagian berikut, Anda dapat mempelajari tentang metrik kesehatan dasbor, dan tindakan yang dapat Anda lakukan untuk memecahkan masalah dan menyelesaikan masalah.

Topik

- [Melihat grafik Kesalahan Penyediaan Instans](#)
- [Melihat grafik Kesalahan Instans Tidak Sehat](#)
- [Melihat grafik Compute Fleet Idle Time](#)

Melihat grafik Kesalahan Penyediaan Instans

Jika Anda melihat nilai bukan nol dalam `Instance Provisioning Errors` grafik, maka itu berarti bahwa contoh EC2 untuk mendukung node slurm gagal diluncurkan pada API `or. CreateFleet RunInstance`

Melihat **IAMPolicyErrors**

- Apa yang terjadi?

Sejumlah instance gagal diluncurkan, yang disebabkan oleh izin yang tidak memadai dengan kode kesalahan. `UnauthorizedOperation`

- Bagaimana cara mengatasinya?

Jika Anda telah mengkonfigurasi kustom [InstanceRole](#) atau [InstanceProfile](#), periksa kebijakan IAM Anda dan memverifikasi bahwa Anda menggunakan kredensial yang benar.

Periksa `clustermgtd` file untuk rincian kesalahan simpul statis. Periksa `slurm_resume.log` file untuk rincian kesalahan simpul dinamis. Gunakan detail untuk mempelajari lebih lanjut tentang izin yang hilang yang harus ditambahkan.

Melihat **VcpuLimitErrors**

- Apa yang terjadi?

AWS ParallelCluster gagal meluncurkan instans karena mencapai batas vCPU pada Anda Akun AWS untuk jenis instans EC2 tertentu yang Anda konfigurasi untuk node komputasi kluster.

- Bagaimana cara mengatasinya?

Periksa `VcpuLimitExceeded` kesalahan dalam `clustermgtd` file untuk node statis, dan periksa di `slurm_resume.log` file untuk node dinamis untuk mendapatkan detail tambahan. Untuk mengatasi masalah ini, Anda dapat meminta peningkatan batas vCPU Anda. Untuk informasi selengkapnya tentang cara melihat batas saat ini dan meminta batas baru, lihat [kuota layanan Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Melihat **VolumeLimitErrors**

- Apa yang terjadi?

Anda telah mencapai batas volume Amazon EBS pada Anda Akun AWS, dan AWS ParallelCluster tidak dapat meluncurkan instans dengan kode `InsufficientVolumeCapacity` kesalahan atau `VolumeLimitExceeded`

- Bagaimana cara mengatasinya?

Periksa `clustermgtd` file untuk node statis, dan periksa `slurm_resume.log` file untuk node dinamis untuk mendapatkan rincian batas volume tambahan. Untuk mengatasi masalah ini, Anda dapat menggunakan yang berbeda Wilayah AWS, membersihkan volume yang ada, atau menghubungi Pusat AWS Support untuk mengirimkan permintaan guna meningkatkan batas volume Amazon EBS Anda.

Melihat **InsufficientCapacityErrors**

- Apa yang terjadi?

AWS ParallelClustertidak memiliki kapasitas yang cukup untuk meluncurkan instans EC2 untuk mendukung node.

- Bagaimana cara mengatasinya?

Periksa `clustermgtd` file untuk node statis, dan periksa `slurm_resume.log` file untuk node dinamis untuk mendapatkan rincian kesalahan kapasitas yang tidak mencukupi. Untuk memecahkan masalah, ikuti panduan di <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>.

OtherInstanceLaunchFailures

- Apa yang terjadi?

Instance EC2 untuk mendukung node komputasi gagal diluncurkan dengan API `or. CreateFleet` `RunInstance`

- Bagaimana cara mengatasinya?

Periksa `clustermgtd` file untuk node statis, dan periksa `slurm_resume.log` file untuk node dinamis untuk mendapatkan detail kesalahan.

Melihat grafik Kesalahan Instans Tidak Sehat

- Apa yang terjadi?

Sejumlah instance komputasi diluncurkan tetapi kemudian diakhiri sebagai tidak sehat.

- Bagaimana cara mengatasinya?

Untuk informasi lebih lanjut tentang pemecahan masalah node yang tidak sehat, lihat.

[Memecahkan masalah penggantian dan penghentian node yang tidak terduga](#)

Melihat **InstanceBootstrapTimeoutError**

- Apa yang terjadi?

Sebuah instance tidak dapat bergabung dengan cluster dalam `resume_timeout` (untuk node dinamis) atau `node_replacement_timeout` (untuk node statis). Ini dapat terjadi jika jaringan tidak dikonfigurasi dengan benar untuk node komputasi, atau dapat terjadi jika skrip kustom yang berjalan pada node komputasi membutuhkan waktu terlalu lama untuk diselesaikan.

- Bagaimana cara mengatasinya?

Untuk node dinamis, periksa `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) untuk alamat IP node komputasi dan kesalahan seperti berikut:

```
Node bootstrap error: Resume timeout expires for node
```

Untuk node statis, periksa `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) untuk alamat IP node komputasi dan kesalahan seperti berikut:

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

Untuk detail tambahan, periksa `/var/log/cloud-init-output.log` file untuk kesalahan. Anda dapat mengambil alamat IP node komputasi bermasalah dari `clustermgtd` dan `slurm_resume` file log.

Melihat **EC2HealthCheckErrors**

- Apa yang terjadi?

Instance gagal pemeriksaan kesehatan EC2.

- Bagaimana cara mengatasinya?

Untuk informasi tentang cara memecahkan masalah ini, lihat [Memecahkan masalah instans dengan pemeriksaan status gagal](#).

Melihat **ScheduledEventHealthCheckErrors**

- Apa yang terjadi?

Sebuah instans gagal pemeriksaan kesehatan acara terjadwal EC2, dan itu tidak sehat.

- Bagaimana cara mengatasinya?

Untuk informasi tentang memecahkan masalah ini, lihat [Peristiwa terjadwal untuk instans Anda](#).

Melihat **NoCorrespondingInstanceErrors**

- Apa yang terjadi?

AWS ParallelClustertidak dapat menemukan instance backing node.

Node memiliki kemungkinan self-terminated selama operasi bootstrap.

[SlurmQueues/CustomActions/OnNodeStart| OnNodeConfigured](#)skrip, atau kesalahan jaringan dapat menghasilkanNoCorrespondingInstanceErrors.

- Bagaimana cara mengatasinya?

Untuk detail tambahan, periksa `/var/log/cloud-init-output.log` untuk node komputasi.

Melihat grafik Compute Fleet Idle Time

Melihat **MaxDynamicNodeIdleTime** yang jauh lebih panjang dari ambang batas Idle Time Scaledown

- Apa yang terjadi?

Instans Anda tidak berakhir dengan benar. MaxDynamicNodeIdleTimemenunjukkan waktu maksimum dalam hitungan detik bahwa node dinamis, didukung oleh instans EC2, menganggur. Ambang batas Scaledown Waktu Idle berasal dari parameter konfigurasi cluster.

[ScaledownIdleTime](#) Ketika node komputasi telah menganggur selama lebih dari detik Scaledown Waktu Idle, matikan Slurm node dan AWS ParallelCluster mengakhiri instance backing. Dalam hal ini, ada sesuatu yang mencegah penghentian instance.

- Bagaimana cara mengatasinya?

Untuk informasi selengkapnya tentang masalah ini, lihat [Mengganti, mengakhiri, atau mematikan instance dan node yang bermasalah](#) di [Memecahkan masalah penskalaan](#).

Memecahkan masalah penerapan klaster

Jika cluster Anda gagal dibuat dan memutar kembali pembuatan tumpukan, Anda dapat melihat file log untuk mendiagnosis masalah. Pesan kegagalan kemungkinan terlihat seperti output berikut:

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
--cluster-configuration cluster-config.yaml
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}

$ pcluster describe-cluster --cluster-name mycluster --region eu-west-1
{
  "creationTime": "2021-09-06T11:03:47.696Z",
  ...
  "cloudFormationStackStatus": "ROLLBACK_IN_PROGRESS",
  "clusterName": "mycluster",
  "computeFleetStatus": "UNKNOWN",
  "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
  "lastUpdatedTime": "2021-09-06T11:03:47.696Z",
  "region": "eu-west-1",
  "clusterStatus": "CREATE_FAILED"
}
```

Topik

- [Lihat AWS CloudFormation acara di CREATE_FAILED](#)
- [Gunakan CLI untuk melihat aliran log](#)
- [Buat ulang cluster yang gagal dengan rollback-on-failure](#)

Lihat AWS CloudFormation acara di **CREATE_FAILED**

Anda dapat menggunakan konsol atau AWS ParallelCluster CLI untuk melihat CloudFormation peristiwa pada CREATE_FAILED kesalahan untuk membantu menemukan akar penyebabnya.

Topik

- [Lihat acara di CloudFormation konsol](#)
- [Gunakan CLI untuk melihat dan memfilter acara CloudFormation CREATE_FAILED](#)

Lihat acara di CloudFormation konsol

Untuk melihat informasi lebih lanjut tentang apa yang menyebabkan "CREATE_FAILED" status, Anda dapat menggunakan CloudFormation konsol.

Lihat pesan CloudFormation kesalahan dari konsol.

1. Masuk ke AWS Management Console dan navigasikan ke <https://console.aws.amazon.com/cloudformation>.
2. Pilih tumpukan bernama *cluster_name*.
3. Pilih tab Acara.
4. Periksa Status sumber daya yang gagal dibuat dengan menggulir daftar peristiwa sumber daya berdasarkan ID Logis. Jika subtugas gagal dibuat, kerjakan mundur untuk menemukan peristiwa sumber daya yang gagal.
5. Sebagai contoh, jika Anda melihat pesan status berikut, Anda harus menggunakan jenis instans yang tidak akan melebihi batas vCPU Anda saat ini atau meminta lebih banyak kapasitas vCPU.

```
2022-02-04 16:09:44 UTC-0800 HeadNode CREATE_FAILED You have requested more vCPU
capacity than your current vCPU limit of 0 allows
    for the instance bucket that the specified instance type belongs to. Please
visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to
this limit.
    (Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request
ID: a9876543-b321-c765-d432-dcba98766789; Proxy: null).
```

Gunakan CLI untuk melihat dan memfilter acara CloudFormation **CREATE_FAILED**

Untuk mendiagnosis masalah pembuatan cluster, Anda dapat menggunakan [pcluster get-cluster-stack-events](#) perintah dengan memfilter **CREATE_FAILED** status. Untuk informasi selengkapnya, lihat [Memfilter AWS CLI output](#) di Panduan AWS Command Line Interface Pengguna.

```
$ pcluster get-cluster-stack-events --cluster-name mycluster --region eu-west-1 \
  --query 'events[?resourceStatus==`CREATE_FAILED`]'
[
  {
    "eventId": "3ccdedd0-0f03-11ec-8c06-02c352fe2ef9",
    "physicalResourceId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "The following resource(s) failed to create: [HeadNode].",
  },
  {
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "mycluster",
    "resourceType": "AWS::CloudFormation::Stack",
    "timestamp": "2021-09-06T11:11:51.780Z"
  },
  {
    "eventId": "HeadNode-CREATE_FAILED-2021-09-06T11:11:50.127Z",
    "physicalResourceId": "i-04e91cc1f4ea796fe",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "Received FAILURE signal with UniqueId
i-04e91cc1f4ea796fe",
    "resourceProperties": "{\"LaunchTemplate\":{\"Version\":\"1\",\"LaunchTemplateId
\":\"lt-057d2b1e687f05a62\"}}",
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "HeadNode",
    "resourceType": "AWS::EC2::Instance",
    "timestamp": "2021-09-06T11:11:50.127Z"
  }
]
```

Pada contoh sebelumnya, kegagalan ada di setup node head.

Gunakan CLI untuk melihat aliran log

Untuk men-debug masalah semacam ini, Anda dapat membuat daftar aliran log yang tersedia dari node kepala dengan [pcluster list-cluster-log-streams](#) dengan memfilter untuk node-type dan kemudian menganalisis konten aliran log.

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
--filters 'Name=node-type,Values=HeadNode'
{
  "logStreams": [
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      ...
    },
    ...
  ]
}
```

Dua aliran log utama yang dapat Anda gunakan untuk menemukan kesalahan inisialisasi adalah sebagai berikut:

- `cfn-init` adalah log untuk `cfn-init` skrip. Pertama periksa aliran log ini. Anda mungkin melihat `Command chef failed` kesalahan di log ini. Lihatlah baris segera sebelum baris ini untuk lebih spesifik yang terkait dengan pesan kesalahan. Untuk informasi lebih lanjut, lihat [cfn-init](#).
- `cloud-init` adalah log untuk [cloud-init](#). Jika Anda tidak melihat apa pun `cfn-init`, coba periksa log ini selanjutnya.

Anda dapat mengambil konten aliran log dengan menggunakan [pcluster get-cluster-log-events](#) (perhatikan `--limit 5` opsi untuk membatasi jumlah peristiwa yang diambil):

```
$ pcluster get-cluster-log-events --cluster-name mycluster \  
  --region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init \  
  --limit 5  
{  
  "nextToken": "f/36370880979637159565202782352491087067973952362220945409/s",  
  "prevToken": "b/36370880752972385367337528725601470541902663176996585497/s",  
  "events": [  
    {  
      "message": "2021-09-06 11:11:39,049 [ERROR] Unhandled exception during build:  
Command runpostinstall failed",  
      "timestamp": "2021-09-06T11:11:39.049Z"  
    },  
    {  
      "message": "Traceback (most recent call last):\n File \"/opt/aws/bin/  
cfn-init\", line 176, in <module>\n   worklog.build(metadata, configSets)\n File \"/usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line  
135, in build\n   Contractor(metadata).build(configSets, self)\n File \"/  
usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 561, in  
build\n   self.run_config(config, worklog)\n File \"/usr/lib/python3.7/  
site-packages/cfnbootstrap/construction.py\", line 573, in run_config\n   CloudFormationCarpenter(config, self._auth_config).build(worklog)\n File \"/usr/  
lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 273, in build\n   self._config.commands)\n File \"/usr/lib/python3.7/site-packages/cfnbootstrap/  
command_tool.py\", line 127, in apply\n   raise ToolError(u\"Command %s failed\" %  
name)",  
      "timestamp": "2021-09-06T11:11:39.049Z"  
    },  
    {  
      "message": "cfnbootstrap.construction_errors.ToolError: Command runpostinstall  
failed",  
      "timestamp": "2021-09-06T11:11:39.049Z"  
    },  
    {  
      "message": "2021-09-06 11:11:49,212 [DEBUG] CloudFormation client initialized  
with endpoint https://cloudformation.eu-west-1.amazonaws.com",  
      "timestamp": "2021-09-06T11:11:49.212Z"  
    },  
    {  
      "message": "2021-09-06 11:11:49,213 [DEBUG] Signaling resource HeadNode in stack  
mycluster with unique ID i-04e91cc1f4ea796fe and status FAILURE",
```

```
    "timestamp": "2021-09-06T11:11:49.213Z"
  }
]
}
```

Pada contoh sebelumnya, kegagalan disebabkan oleh `runpostinstall` kegagalan, sehingga sangat terkait dengan konten skrip bootstrap khusus yang digunakan dalam parameter `OnNodeConfigured` konfigurasi file [CustomActions](#).

Buat ulang cluster yang gagal dengan **rollback-on-failure**

AWS ParallelCluster membuat aliran CloudWatch log cluster di grup log. Anda dapat melihat log ini di CloudWatch konsol Dasbor Kustom atau grup Log. Untuk informasi selengkapnya, silakan lihat [Integrasi dengan Amazon CloudWatch Logs](#) dan [CloudWatchDasbor Amazon](#). Jika tidak ada aliran log yang tersedia, kegagalan mungkin disebabkan oleh skrip bootstrap [CustomActions](#) khusus atau masalah terkait AMI. Untuk mendiagnosis masalah pembuatan dalam kasus ini, buat cluster lagi menggunakan `pcluster create-cluster`, termasuk `--rollback-on-failure` parameter yang disetel ke `false`. Kemudian, gunakan SSH untuk melihat cluster, seperti yang ditunjukkan pada berikut ini:

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
  --cluster-configuration cluster-config.yaml --rollback-on-failure false
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
$ pcluster ssh --cluster-name mycluster
```

Setelah Anda masuk ke node kepala, Anda harus menemukan tiga file log utama yang dapat Anda gunakan untuk menemukan kesalahan.

- `/var/log/cfn-init.log` adalah log untuk `cfn-init` skrip. Pertama periksa log ini. Anda mungkin melihat kesalahan seperti `Command chef failed` di log ini. Lihatlah baris segera

sebelum baris ini untuk lebih spesifik yang terkait dengan pesan kesalahan. Untuk informasi lebih lanjut, lihat [cfn-init](#).

- `/var/log/cloud-init.log` adalah log untuk [cloud-init](#). Jika Anda tidak melihat apa pun `cfn-init.log`, coba periksa log ini selanjutnya.
- `/var/log/cloud-init-output.log` adalah output dari perintah yang dijalankan oleh [cloud-init](#). Ini termasuk output dari `cfn-init`. Dalam kebanyakan kasus, Anda tidak perlu melihat log ini untuk memecahkan masalah jenis ini.

Memecahkan masalah penskalaan

Bagian ini relevan dengan cluster yang diinstal menggunakan AWS ParallelCluster versi 3.0.0 dan yang lebih baru dengan penjadwal pekerjaan Slurm. Untuk informasi selengkapnya tentang mengonfigurasi beberapa antrian, lihat [Konfigurasi beberapa antrian](#)

Jika salah satu cluster yang sedang berjalan mengalami masalah, tempatkan kluster dalam STOPPED status dengan menjalankan perintah berikut sebelum Anda mulai memecahkan masalah. Ini mencegah timbulnya biaya tak terduga.

```
$ pcluster update-compute-fleet --cluster-name mycluster \  
--status STOP_REQUESTED
```

Anda dapat membuat daftar aliran log yang tersedia dari node cluster dengan menggunakan [pcluster list-cluster-log-streams](#) perintah dan pemfilteran dengan menggunakan `private-dns-name` salah satu node yang gagal atau node kepala:

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \  
--filters 'Name=private-dns-name,Values=ip-10-0-0-101'
```

Kemudian, Anda dapat mengambil konten aliran log untuk menganalisisnya dengan menggunakan [pcluster get-cluster-log-events](#) perintah dan meneruskan `--log-stream-name` yang sesuai dengan salah satu log kunci yang disebutkan di bagian berikut:

```
$ pcluster get-cluster-log-events --cluster-name mycluster \  
--region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init
```

AWS ParallelCluster membuat aliran CloudWatch log cluster di grup log. Anda dapat melihat log ini di CloudWatch konsol Dasbor Kustom atau grup Log. Untuk informasi selengkapnya, silakan lihat [Integrasi dengan Amazon CloudWatch Logs](#) dan [CloudWatchDasbor Amazon](#).

Topik

- [Log kunci untuk debugging](#)
- [Melihat InsufficientInstanceCapacity kesalahan slurm_resume.log ketika saya gagal menjalankan pekerjaan, atau clustermgtd.log ketika saya gagal membuat cluster](#)
- [Memecahkan masalah inisialisasi node](#)
- [Memecahkan masalah penggantian dan penghentian node yang tidak terduga](#)
- [Mengganti, mengakhiri, atau mematikan instance dan node yang bermasalah](#)
- [Status antrian \(partisi\) Inactive](#)
- [Memecahkan masalah node dan pekerjaan lain yang diketahui](#)

Log kunci untuk debugging

Tabel berikut memberikan ikhtisar log kunci untuk node kepala:

- `/var/log/cfn-init.log`- Ini adalah log AWS CloudFormation init. Ini berisi semua perintah yang dijalankan ketika sebuah instance disiapkan. Gunakan untuk memecahkan masalah inisialisasi.
- `/var/log/chef-client.log`- Ini adalah log klien Chef. Ini berisi semua perintah yang dijalankan melalui chef/CINC. Gunakan untuk memecahkan masalah inisialisasi.
- `/var/log/parallelcluster/slurm_resume.log`- Ini adalah ResumeProgram log. Ini meluncurkan instance untuk node dinamis. Gunakan untuk memecahkan masalah peluncuran node dinamis.
- `/var/log/parallelcluster/slurm_suspend.log`- Ini SuspendProgram log. Ini disebut ketika instance dihentikan untuk node dinamis. Gunakan untuk memecahkan masalah penghentian node dinamis. Saat Anda memeriksa log ini, Anda juga harus memeriksa `clustermgtd` log.
- `/var/log/parallelcluster/clustermgtd`- Ini `clustermgtd` log. Ini berjalan sebagai daemon terpusat yang mengelola sebagian besar tindakan operasi cluster. Gunakan untuk memecahkan masalah peluncuran, penghentian, atau operasi klaster apa pun.
- `/var/log/slurmctld.log`- Ini adalah log daemon kontrol Slurm. AWS ParallelClustertidak membuat keputusan penskalaan. Sebaliknya, ia hanya mencoba meluncurkan sumber daya untuk

memenuhi persyaratan Slurm. Ini berguna untuk masalah penskalaan dan alokasi, masalah terkait pekerjaan, dan masalah peluncuran dan penghentian terkait penjadwal.

- `/var/log/parallelcluster/compute_console_output`- Log ini merekam output konsol dari subset sampel node komputasi statis yang tiba-tiba dihentikan. Gunakan log ini jika node komputasi statis berakhir dan log node komputasi tidak tersedia di `CloudWatch compute_console_output` logKonten yang Anda terima sama saat Anda menggunakan konsol EC2 atau AWS CLI untuk mengambil output konsol instans.

Ini adalah log kunci untuk node komputasi:

- `/var/log/cloud-init-output.log`- Ini adalah log [cloud-init](#). Ini berisi semua perintah yang dijalankan ketika sebuah instance disiapkan. Gunakan untuk memecahkan masalah inisialisasi.
- `/var/log/parallelcluster/computemgtd`- Ini `computemgtd` log. Ini berjalan pada setiap node komputasi untuk memantau node dalam peristiwa yang tidak biasa bahwa `clustermgtd` daemon pada node kepala sedang offline. Gunakan untuk memecahkan masalah penghentian yang tidak terduga.
- `/var/log/slurmd.log`- Ini adalah log daemon komputasi Slurm. Gunakan untuk memecahkan masalah inisialisasi dan kegagalan komputasi.

Melihat **InsufficientInstanceCapacity** kesalahan **slurm_resume.log** ketika saya gagal menjalankan pekerjaan, atau **clustermgtd.log** ketika saya gagal membuat cluster

Jika cluster menggunakan Slurm penjadwal, Anda mengalami masalah kapasitas yang tidak mencukupi. Jika tidak ada cukup instance yang tersedia saat permintaan peluncuran instance dibuat, **InsufficientInstanceCapacity** kesalahan akan dikembalikan.

Untuk kapasitas instance statis, Anda dapat menemukan kesalahan dalam `clustermgtd` log di `/var/log/parallelcluster/clustermgtd`.

Untuk kapasitas instance dinamis, Anda dapat menemukan kesalahan di `ResumeProgram` log di `/var/log/parallelcluster/slurm_resume.log`.

Pesannya terlihat mirip dengan contoh berikut:

```
An error occurred (InsufficientInstanceCapacity) when calling the RunInstances/  
CreateFleet operation...
```

Berdasarkan kasus penggunaan Anda, pertimbangkan untuk menggunakan salah satu metode berikut untuk menghindari jenis pesan kesalahan ini:

- Nonaktifkan grup penempatan jika diaktifkan. Untuk informasi selengkapnya, lihat [Grup penempatan dan masalah peluncuran instance](#).
- Cadangan kapasitas untuk instans dan luncurkan dengan ODCR (Pemesanan Kapasitas Sesuai Permintaan). Untuk informasi selengkapnya, lihat [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#).
- Konfigurasi beberapa sumber daya komputasi dengan tipe instans yang berbeda. Jika beban kerja Anda tidak memerlukan jenis instans tertentu, Anda dapat memanfaatkan kegagalan kapasitas yang tidak mencukupi dengan beberapa sumber daya komputasi. Untuk informasi selengkapnya, lihat [Slurmcluster cepat kapasitas tidak mencukupi fail-over](#).
- Konfigurasi beberapa tipe instans dalam sumber daya komputasi yang sama, dan manfaatkan alokasi tipe instans ganda. Untuk informasi selengkapnya tentang mengonfigurasi beberapa instance, lihat [Beberapa alokasi tipe instans dengan Slurm dan Scheduling//SlurmQueues/ComputeResources. Instances](#)
- Pindahkan antrian ke Availability Zone yang berbeda dengan mengubah subnet ID dalam konfigurasi cluster [Scheduling//SlurmQueues/Networking. SubnetIds](#)
- Jika beban kerja Anda tidak digabungkan dengan erat, rentangkan antrian di berbagai Availability Zone. Untuk informasi selengkapnya tentang mengonfigurasi beberapa subnet, lihat [Scheduling//SlurmQueues/Networking. SubnetIds](#)

Memecahkan masalah inisialisasi node

Bagian ini mencakup bagaimana Anda dapat memecahkan masalah inisialisasi node. Ini termasuk masalah di mana node gagal diluncurkan, dinyalakan, atau bergabung dengan cluster.

Topik

- [Node kepala](#)
- [Hitung node](#)

Node kepala

Log yang berlaku:

- `/var/log/cfn-init.log`
- `/var/log/chef-client.log`
- `/var/log/parallelcluster/clustermgtd`
- `/var/log/parallelcluster/slurm_resume.log`
- `/var/log/slurmctld.log`

Periksa `/var/log/cfn-init.log` dan `/var/log/chef-client.log` log atau aliran log yang sesuai. Log ini berisi semua tindakan yang dijalankan saat node kepala disiapkan. Sebagian besar kesalahan yang terjadi selama pengaturan harus memiliki pesan kesalahan yang terletak di `/var/log/chef-client.log` log. Jika `OnNodeStart` atau `OnNodeConfigured` skrip ditentukan dalam konfigurasi cluster, periksa kembali apakah skrip berjalan dengan sukses melalui pesan log.

Ketika sebuah cluster dibuat, node kepala harus menunggu node komputasi untuk bergabung dengan cluster sebelum dapat bergabung dengan cluster. Karena itu, jika node komputasi gagal bergabung dengan cluster, maka node kepala juga gagal. Anda dapat mengikuti salah satu rangkaian prosedur ini, tergantung pada jenis catatan komputasi yang Anda gunakan, untuk memecahkan masalah jenis ini:

Hitung node

- Log yang berlaku:
 - `/var/log/cloud-init-output.log`
 - `/var/log/slurmd.log`
- Jika node komputasi diluncurkan, periksa terlebih dahulu `/var/log/cloud-init-output.log`, yang harus berisi log pengaturan yang mirip dengan `/var/log/chef-client.log` log pada node kepala. Sebagian besar kesalahan yang terjadi selama pengaturan harus memiliki pesan kesalahan yang terletak di `/var/log/cloud-init-output.log` log. Jika skrip pra-instal atau pasca-instal ditentukan dalam konfigurasi cluster, periksa apakah skrip tersebut berhasil dijalankan.
- Jika Anda menggunakan AMI kustom dengan modifikasi Slurm konfigurasi, maka mungkin ada kesalahan Slurm terkait yang mencegah node komputasi bergabung dengan cluster. Untuk kesalahan terkait penjadwal, periksa log `/var/log/slurmd.log`

Node komputasi dinamis:

- Cari ResumeProgram log (`/var/log/parallelcluster/slurm_resume.log`) untuk nama node komputasi Anda untuk melihat apakah pernah ResumeProgram dipanggil dengan node. (Jika ResumeProgram tidak pernah dipanggil, Anda dapat memeriksa `slurmctld` log (`/var/log/slurmctld.log`) untuk menentukan apakah Slurm pernah mencoba memanggil ResumeProgram dengan node).
- Perhatikan bahwa izin yang salah untuk ResumeProgram dapat menyebabkan kegagalan ResumeProgram secara diam-diam. Jika Anda menggunakan AMI kustom dengan modifikasi untuk ResumeProgram penyiapan, periksa apakah AMI dimiliki oleh `slurm` pengguna dan memiliki izin `744 (rwxr--r--)`. ResumeProgram
- Jika ResumeProgram dipanggil, periksa untuk melihat apakah sebuah instance diluncurkan untuk node. Jika tidak ada instance yang diluncurkan, Anda dapat melihat pesan kesalahan yang menjelaskan kegagalan peluncuran.
- Jika instance diluncurkan, maka mungkin ada masalah selama proses penyiapan. Anda akan melihat alamat IP pribadi dan ID instance yang sesuai dari ResumeProgram log. Selain itu, Anda dapat melihat log pengaturan yang sesuai untuk contoh tertentu. Untuk informasi selengkapnya tentang pemecahan masalah kesalahan penyiapan dengan node komputasi, lihat bagian selanjutnya.

Node komputasi statis:

- Periksa log `clustermgtd` (`/var/log/parallelcluster/clustermgtd`) untuk melihat apakah instance diluncurkan untuk node. Jika tidak diluncurkan, harus ada pesan kesalahan yang jelas yang merinci kegagalan peluncuran.
- Jika instance diluncurkan, ada beberapa masalah selama proses penyiapan. Anda akan melihat alamat IP pribadi dan ID instance yang sesuai dari ResumeProgram log. Selain itu, Anda dapat melihat log pengaturan yang sesuai untuk instance tertentu.

Node komputasi yang didukung oleh Instans Spot:

- Jika ini pertama kalinya Anda menggunakan Instans Spot dan pekerjaan tetap dalam PD (status tertunda), periksa kembali `/var/log/parallelcluster/slurm_resume.log` file tersebut. Anda mungkin akan menemukan kesalahan seperti berikut:

```
2022-05-20 13:06:24,796 - [slurm_plugin.common:add_instances_for_nodes] - ERROR -  
Encountered exception when launching instances for nodes (x1) ['spot-dy-t2micro-2']:  
An error occurred (AuthFailure.ServiceLinkedRoleCreationNotPermitted) when calling  
the RunInstances operation: The provided credentials do not have permission to  
create the service-linked role for EC2 Spot Instances.
```

Saat menggunakan Instans Spot, peran `AWSServiceRoleForEC2Spot` terkait layanan harus ada di akun Anda. Untuk membuat peran ini di akun Anda menggunakan AWS CLI, jalankan perintah berikut:

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

Untuk informasi selengkapnya, lihat [Mengerjakan Instans ITSpot](#) di Panduan AWS ParallelCluster Pengguna dan [peran terkait Layanan untuk permintaan Instans Spot](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Memecahkan masalah penggantian dan penghentian node yang tidak terduga

Bagian ini terus mengeksplorasi bagaimana Anda dapat memecahkan masalah terkait node, khususnya ketika node diganti atau dihentikan secara tidak terduga.

- Log yang berlaku:
 - `/var/log/parallelcluster/clustermgtd`(simpul kepala)
 - `/var/log/slurmctld.log`(simpul kepala)
 - `/var/log/parallelcluster/computemgtd`(simpul komputasi)

Node diganti atau dihentikan secara tak terduga

- Periksa `clustermgtd` log (`/var/log/parallelcluster/clustermgtd`) untuk melihat apakah `clustermgtd` diganti atau dihentikan node. Perhatikan bahwa `clustermgtd` menangani semua tindakan pemeliharaan node normal.
- Jika `clustermgtd` diganti atau dihentikan node, harus ada pesan yang merinci mengapa tindakan ini diambil pada node. Jika alasannya terkait penjadwal (misalnya, karena node masuk DOWN),

periksa `slurmctld` log untuk informasi lebih lanjut. Jika alasannya terkait Amazon EC2, harus ada pesan informatif yang merinci masalah terkait Amazon EC2 yang memerlukan penggantian.

- Jika `clustermgtd` tidak menghentikan node, periksa terlebih dahulu apakah ini merupakan penghentian yang diharapkan oleh Amazon EC2, lebih khusus lagi penghentian spot. `computemgtd`, berjalan pada node komputasi, juga dapat mengakhiri node jika `clustermgtd` ditentukan sebagai tidak sehat. Periksa `computemgtd` log (`/var/log/parallelcluster/computemgtd`) untuk melihat apakah node `computemgtd` dihentikan.

Node gagal

- Periksa `slurmctld` log (`/var/log/slurmctld.log`) untuk melihat mengapa pekerjaan atau node gagal. Perhatikan bahwa pekerjaan secara otomatis diantrian ulang jika node gagal.
- Jika `slurm_resume` melaporkan bahwa node diluncurkan dan `clustermgtd` melaporkan setelah beberapa menit bahwa tidak ada instance yang sesuai di Amazon EC2 untuk node tersebut, node mungkin gagal selama persiapan. Untuk mengambil log dari compute (`/var/log/cloud-init-output.log`), lakukan langkah-langkah berikut:
 - Kirim pekerjaan untuk membiarkan Slurm putaran node baru.
 - Setelah node dimulai, aktifkan perlindungan terminasi menggunakan perintah ini.

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --disable-api-termination
```

- Ambil output konsol dari node dengan perintah ini.

```
$ aws ec2 get-console-output --instance-id i-1234567890abcdef0 --output text
```

Mengganti, mengakhiri, atau mematikan instance dan node yang bermasalah

- Log yang berlaku:
 - `/var/log/parallelcluster/clustermgtd`(simpul kepala)
 - `/var/log/parallelcluster/slurm_suspend.log`(simpul kepala)
- Dalam kebanyakan kasus, `clustermgtd` menangani semua tindakan penghentian instance yang diharapkan. Periksa di `clustermgtd` log untuk melihat mengapa gagal mengganti atau mengakhiri node.

- Untuk node dinamis gagal [SlurmSettingsProperti](#), periksa SuspendProgram log untuk melihat apakah SuspendProgram dipanggil `slurmctld` dengan node tertentu sebagai argumen. Perhatikan bahwa SuspendProgram tidak benar-benar melakukan tindakan apa pun. Sebaliknya, itu hanya log ketika dipanggil. Semua penghentian dan NodeAddr reset instance dilakukan oleh `clusterimgtd`. Slurm menempatkan node kembali ke POWER_SAVING keadaan setelah SuspendTimeout secara otomatis.
- Jika node komputasi gagal terus menerus karena kegagalan bootstrap, verifikasi apakah node tersebut diluncurkan dengan [Slurmmodus cluster dilindungi](#) diaktifkan. Jika mode terlindungi tidak diaktifkan, ubah pengaturan mode terlindungi untuk mengaktifkan mode terlindungi. Memecahkan masalah dan memperbaiki skrip bootstrap.

Status antrian (partisi) **Inactive**

Jika Anda menjalankan `sinfo` dan output menunjukkan antrian dengan AVAIL status `inact`, kluster Anda mungkin telah [Slurmmodus cluster dilindungi](#) diaktifkan dan antrian telah disetel ke INACTIVE status untuk jangka waktu yang telah ditentukan sebelumnya.

Memecahkan masalah node dan pekerjaan lain yang diketahui

Jenis lain dari masalah yang diketahui adalah yang AWS ParallelCluster mungkin gagal mengalokasikan pekerjaan atau membuat keputusan penskalaan. Dengan jenis masalah ini, AWS ParallelCluster hanya meluncurkan, mengakhiri, atau memelihara sumber daya sesuai dengan instruksi Slurm. Untuk masalah ini, periksa `slurmctld` log untuk memecahkan masalah mereka.

Grup penempatan dan masalah peluncuran instance

Untuk mendapatkan latensi antar simpul terendah, gunakan grup penempatan. Grup penempatan memastikan bahwa instance Anda berada di tulang punggung jaringan yang sama. Jika tidak ada cukup instance yang tersedia saat permintaan dibuat, `InsufficientInstanceCapacity` kesalahan akan dikembalikan. Untuk mengurangi kemungkinan menerima kesalahan ini saat menggunakan grup penempatan cluster, atur [Enabled](#) parameter [SlurmQueues/NetworkingPlacementGroup](#)//`kefalse`.

Untuk kontrol tambahan atas akses kapasitas, pertimbangkan untuk [meluncurkan instans dengan ODCR \(Pemesanan Kapasitas Sesuai Permintaan\)](#).

Untuk informasi selengkapnya, lihat [Memecahkan masalah peluncuran instans](#) serta [peran dan batasan grup Penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Direktori yang tidak dapat diganti

Direktori berikut dibagi antara node dan tidak dapat diganti.

- `/home`- Ini termasuk folder home pengguna default (`/home/ec2_user` di Amazon Linux dan RedHat, `/home/centos` di CentOS, dan `/home/ubuntu` di Ubuntu).
- `/opt/intel`- Ini termasuk Intel MPI, Intel Parallel Studio, dan file terkait.
- `/opt/slurm`- Ini termasuk Slurm Workload Manager dan file terkait. (Bersyarat, hanya jika `Scheduler: slurm`.)

Memecahkan masalah di NICE DCV

Topik

- [Log untuk NICE DCV](#)
- [Masalah Ubuntu NICE DCV](#)

Log untuk NICE DCV

Log untuk NICE DCV ditulis ke file di direktori `/var/log/dcv/`. Meninjau log ini dapat membantu memecahkan masalah.

Jenis instance harus memiliki setidaknya 1,7 gibibytes (GiB) RAM untuk menjalankan NICE DCV. Jenis instans nano dan mikro tidak memiliki cukup memori untuk menjalankan NICE DCV.

AWS ParallelCluster membuat aliran log NICE DCV di grup log. Anda dapat melihat log ini di CloudWatch konsol Dasbor Kustom atau grup Log. Untuk informasi selengkapnya, silakan lihat [Integrasi dengan Amazon CloudWatch Logs](#) dan [CloudWatchDasbor Amazon](#).

Masalah Ubuntu NICE DCV

Saat menjalankan Terminal Gnome melalui sesi NICE DCV di Ubuntu, Anda mungkin tidak secara otomatis memiliki akses ke lingkungan pengguna AWS ParallelCluster yang tersedia melalui shell login. Lingkungan pengguna menyediakan modul lingkungan seperti `openmpi` atau `intelmpi`, dan pengaturan pengguna lainnya.

Pengaturan default Terminal Gnome mencegah shell dimulai sebagai shell login. Ini berarti bahwa profil shell tidak bersumber secara otomatis dan lingkungan AWS ParallelCluster pengguna tidak dimuat.

Untuk mendapatkan sumber profil shell dengan benar dan mengakses lingkungan AWS ParallelCluster pengguna, lakukan salah satu hal berikut:

- Ubah pengaturan terminal default:
 1. Pilih menu Edit di terminal Gnome.
 2. Pilih Preferensi, lalu Profil.
 3. Pilih Command dan pilih Run Command sebagai shell login.
 4. Buka terminal baru.
- Gunakan baris perintah untuk sumber profil yang tersedia:

```
$ source /etc/profile && source $HOME/.bashrc
```

Memecahkan masalah dalam cluster dengan integrasi AWS Batch

Bagian ini relevan dengan cluster dengan integrasi AWS Batch scheduler.

Topik

- [Masalah simpul kepala](#)
- [Masalah komputasi](#)
- [Kegagalan Job](#)
- [Connect timeout pada kesalahan URL endpoint](#)

Masalah simpul kepala

Anda dapat memecahkan masalah penyiapan node kepala dengan cara yang sama seperti Slurm cluster (kecuali untuk log Slurm tertentu). Untuk informasi lebih lanjut tentang masalah ini, lihat [Node kepala](#).

Masalah komputasi

AWS Batch mengelola aspek penskalaan dan komputasi layanan Anda. Jika Anda mengalami masalah terkait komputasi, lihat dokumentasi AWS Batch [pemecahan masalah](#) untuk mendapatkan bantuan.

Kegagalan Job

Jika pekerjaan gagal, Anda dapat menjalankan [awsbout](#) perintah untuk mengambil output pekerjaan. Anda juga dapat menjalankan [awsbstat](#) perintah untuk mendapatkan tautan ke log pekerjaan yang disimpan oleh Amazon CloudWatch.

Connect timeout pada kesalahan URL endpoint

Jika pekerjaan paralel multi-node gagal dengan kesalahan:: Connect timeout on endpoint URL

- Di log `awsbout` keluaran, periksa apakah pekerjaannya paralel multi-node dari output: `Detected 3/3 compute nodes. Waiting for all compute nodes to start.`
- Verifikasi apakah subnet node komputasi bersifat publik.

Pekerjaan paralel multi-node tidak mendukung penggunaan subnet publik saat menggunakan AWS Batch in. AWS ParallelCluster Gunakan subnet pribadi untuk node komputasi dan pekerjaan Anda. Untuk informasi selengkapnya, lihat [Menghitung pertimbangan lingkungan](#) di AWS Batch Panduan Pengguna. Untuk mengonfigurasi subnet pribadi untuk node komputasi Anda, lihat. [AWS ParallelCluster dengan AWS Batch scheduler](#)

Memecahkan masalah integrasi multi-pengguna dengan Active Directory

Bagian ini relevan dengan cluster yang terintegrasi dengan Active Directory.

Jika fitur integrasi Active Directory tidak berfungsi seperti yang diharapkan, log SSSD dapat memberikan informasi diagnostik yang berguna. Log ini terletak di `/var/log/sss` pada node cluster. Secara default, mereka juga disimpan dalam grup CloudWatch log Amazon cluster.

Topik

- [Pemecahan masalah khusus Active Directory](#)
- [Aktifkan mode debug](#)
- [Cara pindah dari LDAPS ke LDAP](#)
- [Cara menonaktifkan verifikasi sertifikat server LDAPS](#)
- [Cara masuk dengan kunci SSH daripada kata sandi](#)
- [Cara mengatur ulang kata sandi pengguna dan kata sandi yang kedaluwarsa](#)
- [Cara memverifikasi domain yang bergabung](#)
- [Cara memecahkan masalah dengan sertifikat](#)
- [Cara memverifikasi bahwa integrasi dengan Active Directory berfungsi](#)
- [Cara memecahkan masalah masuk untuk menghitung node](#)
- [Masalah yang diketahui dengan pekerjaan SimCenter StarCCM+ di lingkungan multi-pengguna](#)
- [Masalah yang diketahui dengan resolusi nama pengguna](#)
- [Cara mengatasi masalah pembuatan direktori home](#)

Pemecahan masalah khusus Active Directory

Bagian ini relevan untuk pemecahan masalah khusus untuk jenis Active Directory.

Simple AD

- `DomainReadOnlyUser` Nilai harus cocok dengan pencarian basis direktori Simple AD untuk pengguna:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

Catatan `cn` untuk `Users`.

- Pengguna admin default adalah `Administrator`.
- `Ldapsearch` membutuhkan nama NetBIOS sebelum nama pengguna.

`Ldapsearch` sintaks harus sebagai berikut:

```
$ ldapsearch -x -D "corp\\Administrator" -w "Password" -H ldap://192.0.2.103 \  
-b "cn=Users,dc=corp,dc=example,dc=com"
```


AWS Managed Microsoft AD

- DomainReadOnlyUserNilai harus sesuai dengan pencarian basis AWS Managed Microsoft AD direktori untuk pengguna:

```
cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

- Pengguna admin default adalahAdmin.
- Ldapsearchsintaks harus sebagai berikut:

```
$ ldapsearch -x -D "Admin" -w "Password" -H ldap://192.0.2.103 \  
-b "ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

Aktifkan mode debug

Log debug dari SSSD dapat berguna untuk memecahkan masalah. Untuk mengaktifkan mode debug, Anda harus memperbarui cluster dengan perubahan berikut yang dibuat pada konfigurasi cluster:

```
DirectoryService:  
  AdditionalSssdConfigs:  
    debug_level: "0x1ff"
```

Cara pindah dari LDAPS ke LDAP

Pindah dari LDAP (LDAP dengan TLS/SSL) ke LDAP tidak disarankan karena LDAP saja tidak menyediakan enkripsi apa pun. Namun demikian, ini dapat berguna untuk tujuan pengujian dan pemecahan masalah.

Anda dapat mengembalikan cluster ke konfigurasi sebelumnya dengan memperbarui cluster dengan definisi konfigurasi sebelumnya.

Untuk berpindah dari LDAPS ke LDAP, Anda harus memperbarui cluster dengan perubahan berikut dalam konfigurasi cluster:

```
DirectoryService:  
  LdapTlsReqCert: never  
  AdditionalSssdConfigs:  
    ldap_auth_disable_tls_never_use_in_production: True
```

Cara menonaktifkan verifikasi sertifikat server LDAPS

Ini dapat berguna untuk menonaktifkan sementara verifikasi sertifikat server LDAPS pada node kepala, untuk tujuan pengujian atau pemecahan masalah.

Anda dapat mengembalikan cluster ke konfigurasi sebelumnya dengan memperbarui cluster dengan definisi konfigurasi sebelumnya.

Untuk menonaktifkan verifikasi sertifikat server LDAPS, Anda harus memperbarui cluster dengan perubahan berikut dalam konfigurasi cluster:

```
DirectoryService:  
  LdapTlsReqCert: never
```

Cara masuk dengan kunci SSH daripada kata sandi

Kunci SSH dibuat `/home/$user/.ssh/id_rsa` setelah pertama kali Anda masuk dengan kata sandi. Untuk masuk dengan kunci SSH, Anda harus masuk dengan kata sandi Anda, menyalin kunci SSH secara lokal, dan kemudian menggunakannya ke SSH tanpa kata sandi seperti biasa:

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY $username@$head_node_ip
```

Cara mengatur ulang kata sandi pengguna dan kata sandi yang kedaluwarsa

Jika pengguna kehilangan akses ke kluster, [AWS Managed Microsoft AD kata sandi mereka mungkin telah kedaluwarsa](#).

Untuk mengatur ulang kata sandi, jalankan perintah berikut dengan pengguna dan peran yang memiliki izin tulis di direktori:

```
$ aws ds reset-user-password \  
  --directory-id "d-abcdef01234567890" \  
  --user-name "USER_NAME" \  
  --new-password "NEW_PASSWORD" \  
  --region "region-id"
```

Jika Anda mengatur ulang kata sandi untuk [DirectoryService/DomainReadOnlyUser](#):

1. Pastikan untuk memperbarui [PasswordSecretArn](#) [rahasia DirectoryService](#)/dengan kata sandi baru.
2. Perbarui cluster untuk nilai rahasia baru:
 - a. Hentikan armada komputasi dengan `pcluster update-compute-fleet` perintah.
 - b. Jalankan perintah berikut dari dalam node kepala cluster.

```
$ sudo /opt/parallelcluster/scripts/directory_service/  
update_directory_service_password.sh
```

Setelah reset kata sandi dan pembaruan cluster, akses cluster pengguna harus dipulihkan.

Untuk informasi selengkapnya, lihat [Mengatur ulang kata sandi pengguna](#) di Panduan AWS Directory Service Administrasi.

Cara memverifikasi domain yang bergabung

Perintah berikut harus dijalankan dari instance yang bergabung ke domain, bukan node kepala.

```
$ realm list corp.example.com \  
type: kerberos \  
realm-name: CORP.EXAMPLE.COM \  
domain-name: corp.example.com \  
configured: kerberos-member \  
server-software: active-directory \  
client-software: sssd \  
required-package: oddjob \  
required-package: oddjob-mkhomedir \  
required-package: sssd \  
required-package: adcli \  
required-package: samba-common-tools \  
login-formats: %U \  
login-policy: allow-realm-logins
```

Cara memecahkan masalah dengan sertifikat

Ketika komunikasi LDAPS tidak berfungsi, itu bisa disebabkan oleh kesalahan dalam komunikasi TLS, yang pada gilirannya dapat disebabkan oleh masalah dengan sertifikat.

Catatan tentang sertifikat:

- Sertifikat yang ditentukan dalam konfigurasi cluster `LdapTlsCaCert` harus berupa bundel sertifikat PEM yang berisi sertifikat untuk seluruh rantai sertifikat otoritas (CA) yang mengeluarkan sertifikat untuk pengontrol domain.
- Bundel sertifikat PEM adalah file yang terbuat dari rangkaian sertifikat PEM.
- Sertifikat dalam format PEM (biasanya digunakan di Linux) setara dengan sertifikat dalam format DER base64 (biasanya diekspor oleh Windows).
- Jika sertifikat untuk pengontrol domain dikeluarkan oleh CA bawahan, maka bundel sertifikat harus berisi sertifikat bawahan dan root CA.

Langkah verifikasi pemecahan masalah:

Langkah-langkah verifikasi berikut mengasumsikan bahwa perintah dijalankan dari dalam node kepala cluster dan bahwa pengontrol domain dapat dijangkau di `SERVER:PORT`

Untuk memecahkan masalah yang terkait dengan sertifikat, ikuti langkah-langkah verifikasi berikut:

Langkah verifikasi:

1. Periksa koneksi ke pengontrol domain Active Directory:

Verifikasi bahwa Anda dapat terhubung ke pengontrol domain. Jika langkah ini berhasil, maka koneksi SSL ke pengontrol domain berhasil dan sertifikat diverifikasi. Masalah Anda tidak terkait dengan sertifikat.

Jika langkah ini gagal, lanjutkan dengan verifikasi berikutnya.

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

2. Periksa verifikasi sertifikat:

Verifikasi bahwa bundel sertifikat CA lokal dapat memvalidasi sertifikat yang disediakan oleh pengontrol domain. Jika langkah ini berhasil, maka masalah Anda tidak terkait dengan sertifikat, tetapi masalah jaringan lainnya.

Jika langkah ini gagal, lanjutkan dengan verifikasi berikutnya.

```
$ openssl verify -verbose -  
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

3. Periksa sertifikat yang disediakan oleh pengontrol domain Active Directory:

Verifikasi bahwa konten sertifikat yang disediakan oleh pengontrol domain adalah seperti yang diharapkan. Jika langkah ini berhasil, Anda mungkin memiliki masalah dengan sertifikat CA yang digunakan untuk memverifikasi pengontrol, lanjutkan ke langkah pemecahan masalah berikutnya.

Jika langkah ini gagal, Anda harus memperbaiki sertifikat yang dikeluarkan untuk pengontrol domain dan menjalankan kembali langkah-langkah pemecahan masalah.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

4. Periksa konten sertifikat:

Verifikasi bahwa konten sertifikat yang disediakan oleh pengontrol domain adalah seperti yang diharapkan. Jika langkah ini berhasil, Anda mungkin memiliki masalah dengan sertifikat CA yang digunakan untuk memverifikasi controller, lanjutkan ke langkah pemecahan masalah berikutnya.

Jika langkah ini gagal, Anda harus memperbaiki sertifikat yang dikeluarkan untuk pengontrol domain dan menjalankan kembali langkah-langkah pemecahan masalah.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

5. Periksa konten bundel sertifikat CA lokal:

Verifikasi bahwa konten bundel sertifikat CA lokal yang digunakan untuk memvalidasi sertifikat pengontrol domain adalah seperti yang diharapkan. Jika langkah ini berhasil, Anda mungkin memiliki masalah dengan sertifikat yang disediakan oleh pengontrol domain.

Jika langkah ini gagal, Anda harus memperbaiki paket sertifikat CA yang dikeluarkan untuk pengontrol domain dan menjalankan kembali langkah-langkah pemecahan masalah.

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

Cara memverifikasi bahwa integrasi dengan Active Directory berfungsi

Jika dua pemeriksaan berikut berhasil, integrasi dengan Active Directory berfungsi.

Cek:

1. Anda dapat menemukan pengguna yang ditentukan dalam direktori:

Dari dalam simpul kepala cluster, sebagai `ec2-user`:

```
$ getent passwd $ANY_AD_USER
```

2. Anda dapat SSH ke node kepala yang menyediakan kata sandi pengguna:

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

Jika cek satu gagal, kami berharap cek dua gagal juga.

Pemeriksaan pemecahan masalah tambahan:

- Verifikasi bahwa pengguna ada di direktori.
- Aktifkan [logging debug](#).
- Pertimbangkan untuk menonaktifkan enkripsi sementara dengan [berpindah dari LDAPS ke LDAP untuk mengesampingkan masalah LDAPS](#).

Cara memecahkan masalah masuk untuk menghitung node

Bagian ini relevan untuk masuk untuk menghitung node dalam cluster yang terintegrasi dengan Active Directory.

Dengan AWS ParallelCluster, login kata sandi ke node komputasi cluster dinonaktifkan berdasarkan desain.

Semua pengguna harus menggunakan kunci SSH mereka sendiri untuk masuk untuk menghitung node.

Pengguna dapat mengambil kunci SSH mereka di node kepala setelah otentikasi pertama (misalnya login), jika [GenerateSshKeysForUsers](#) diaktifkan dalam konfigurasi cluster.

Ketika pengguna mengautentikasi pada node kepala untuk pertama kalinya, mereka dapat mengambil kunci SSH yang secara otomatis dihasilkan untuk mereka sebagai pengguna direktori. Direktori rumah untuk pengguna juga dibuat. Ini juga bisa terjadi saat pertama kali sudo-user beralih ke pengguna di head node.

Jika pengguna belum masuk ke node kepala, kunci SSH tidak dihasilkan dan pengguna tidak akan dapat masuk untuk menghitung node.

Masalah yang diketahui dengan pekerjaan SimCenter StarCCM+ di lingkungan multi-pengguna

Bagian ini relevan dengan pekerjaan yang diluncurkan di lingkungan multi-pengguna oleh Simcenter StarCCM+ perangkat lunak dinamika fluida komputasi dari Siemens.

Jika Anda menjalankan pekerjaan starCCM+v16 yang dikonfigurasi untuk menggunakan IntelMPI yang disematkan, secara default proses MPI di-bootstrap menggunakan SSH.

Karena [bug Slurm](#) yang diketahui yang menyebabkan resolusi nama pengguna salah, pekerjaan mungkin gagal dengan kesalahan seperti `error setting up the bootstrap proxies` Bug ini hanya memengaruhi AWS ParallelCluster versi 3.1.1 dan 3.1.2.

Untuk mencegah hal ini terjadi, paksa IntelMpi untuk menggunakan Slurm sebagai metode bootstrap MPI. [Eksport variabel lingkungan `I_MPI_HYDRA_BOOTSTRAP=slurm` ke dalam skrip pekerjaan yang meluncurkan starCCM+, seperti yang dijelaskan dalam dokumentasi resmi IntelMpi.](#)

Masalah yang diketahui dengan resolusi nama pengguna

Bagian ini relevan untuk mengambil nama pengguna dalam pekerjaan.

Karena [bug yang diketahui di Slurm](#), nama pengguna yang diambil dalam proses pekerjaan mungkin nobody jika Anda menjalankan pekerjaan tanpa `srun` Bug ini hanya memengaruhi AWS ParallelCluster versi 3.1.1 dan 3.1.2.

Misalnya, jika Anda menjalankan perintah `sbatch --wrap 'srun id'` sebagai pengguna direktori, nama pengguna yang benar dikembalikan. Namun, jika Anda menjalankan `sbatch --wrap 'id'` sebagai direktori pengguna, nobody mungkin dikembalikan sebagai nama pengguna.

Anda dapat menggunakan solusi berikut.

1. Luncurkan pekerjaan Anda dengan `'srun'` alih-alih `'sbatch'`, jika memungkinkan.

2. Aktifkan enumerasi SSSD dengan mengatur konfigurasi [AdditionalSssdConfigs](#)in cluster sebagai berikut.

```
AdditionalSssdConfigs:  
  enumerate: true
```

Cara mengatasi masalah pembuatan direktori home

Bagian ini relevan dengan masalah pembuatan direktori home.

Jika Anda melihat kesalahan seperti yang ditunjukkan pada contoh berikut, direktori home tidak dibuat untuk Anda saat pertama kali masuk ke node kepala. Atau, direktori home tidak dibuat untuk Anda saat pertama kali beralih dari sudoer ke pengguna Active Directory di node kepala.

```
$ ssh AD_USER@$HEAD_NODE_IP  
/opt/parallelcluster/scripts/generate_ssh_key.sh failed: exit code 1  
  
  _|  _|_ )  
 _| (    /  Amazon Linux 2 AMI  
__|\___|__|  
  
https://aws.amazon.com/amazon-linux-2/  
Could not chdir to home directory /home/PclusterUser85: No such file or directory
```

Kegagalan pembuatan direktori home dapat disebabkan oleh `oddjob-mkhomedir` paket `oddjob` dan yang diinstal di node kepala cluster.

Tanpa direktori home dan kunci SSH, pengguna tidak dapat mengirimkan pekerjaan atau SSH ke node cluster.

Jika Anda memerlukan `oddjob` paket di sistem Anda, verifikasi bahwa `oddjobd` layanan sedang berjalan dan segarkan file konfigurasi PAM untuk memastikan bahwa direktori home dibuat. Untuk melakukan ini, jalankan perintah di node kepala seperti yang ditunjukkan pada contoh berikut.

```
sudo systemctl start oddjobd  
sudo authconfig --enablemkhomedir --updateall
```

Jika Anda tidak memerlukan `oddjob` paket di sistem Anda, hapus instalannya dan segarkan file konfigurasi PAM untuk memastikan bahwa direktori home dibuat. Untuk melakukan ini, jalankan perintah di node kepala seperti yang ditunjukkan pada contoh berikut.


```
sudo yum remove -y oddjob oddjob-mkhomedir
sudo authconfig --enablemkhomedir --updateall
```

Memecahkan masalah AMI khusus

Saat Anda menggunakan AMI kustom, Anda dapat melihat peringatan berikut:

```
"validationMessages": [
  {
    "level": "WARNING",
    "type": "CustomAmiTagValidator",
    "message": "The custom AMI may not have been created by pcluster. You can ignore
this warning if the AMI is shared or copied from another pcluster AMI. If the
AMI is indeed not created by pcluster, cluster creation will fail. If the cluster
creation fails, please go to https://docs.aws.amazon.com/parallelcluster/latest/ug/troubleshooting.html#troubleshooting-stack-creation-failures for troubleshooting."
  },
  {
    "level": "WARNING",
    "type": "AmiOsCompatibleValidator",
    "message": "Could not check node AMI ami-0000012345 OS and cluster OS alinux2
compatibility, please make sure they are compatible before cluster creation and update
operations."
  }
]
```

Jika Anda yakin bahwa AMI yang benar sedang digunakan, Anda dapat mengabaikan peringatan ini.

Jika Anda tidak ingin melihat peringatan ini di masa mendatang, beri tag AMI khusus dengan tag berikut, di mana *my-os* salah satu dari, *alinux2*, *ubuntu2204* *ubuntu2004centos7*, atau *rhel8* dan **"3.7.0"** adalah versi yang pcluster digunakan:

```
$ aws ec2 create-tags \
  --resources ami-yourcustomAmi \
  --tags Key="parallelcluster:version",Value="3.7.0"
Key="parallelcluster:os",Value="my-os"
```

Memecahkan masalah batas waktu pembaruan klaster saat **cfn-hup** tidak berjalan

`cfn-hupHelper` adalah daemon yang mendeteksi perubahan dalam metadata sumber daya dan menjalankan tindakan yang ditentukan pengguna saat perubahan terdeteksi. Ini adalah cara Anda membuat pembaruan konfigurasi pada instans Amazon EC2 yang sedang berjalan melalui tindakan API. `UpdateStack`

Saat ini `cfn-hup` daemon diluncurkan oleh `supervisord`. Tetapi setelah peluncuran, `cfn-hup` prosesnya terlepas dari `supervisord` kontrol. Jika `cfn-hup` iblis dibunuh oleh aktor eksternal, itu tidak dimulai ulang secara otomatis. Jika `cfn-hup` tidak berjalan, selama pembaruan cluster, CloudFormation tumpukan memulai proses pembaruan seperti yang diharapkan tetapi prosedur pembaruan tidak diaktifkan pada node kepala dan tumpukan akhirnya masuk ke batas waktu. Dari log `cluster/var/log/chef-client`, Anda dapat melihat bahwa resep pembaruan tidak pernah dipanggil.

Periksa dan mulai ulang **cfn-hup** jika terjadi kegagalan

1. Pada node kepala, periksa apakah `cfn-hup` sedang berjalan:

```
$ ps aux | grep cfn-hup
```

2. Periksa `cfn-hup` log `/var/log/cfn-hup.log` dan `/var/log/supervisord.log` pada node kepala.
3. Jika `cfn-hup` tidak berjalan, coba mulai ulang dengan menjalankan:

```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl start cfn-hup
```

Pemecahan masalah jaringan

Cluster dalam satu masalah subnet publik

Periksa `cloud-init-output.log` dari salah satu node komputasi. Jika Anda menemukan sesuatu seperti berikut yang menunjukkan node macet dalam Slurm inisialisasi, kemungkinan besar karena titik akhir DynamoDB VPC yang hilang. Tambahkan titik akhir DynamoDB. Untuk informasi selengkapnya, lihat [AWS ParallelCluster dalam subnet tunggal tanpa akses internet](#).

```
ruby_block[retrieve compute node info] action run[2022-03-11T17:47:11+00:00] INFO:
  Processing ruby_block[retrieve compute node info] action run (aws-parallelcluster-
slurm::init line 31)
```

Pembaruan cluster gagal pada tindakan **onNodeUpdated** kustom

Ketika [OnNodeUpdated](#) skrip [HeadNode/CustomActions](#)/gagal, pembaruan gagal dan skrip tidak dijalankan pada waktu rollback. Anda bertanggung jawab untuk melakukan pembersihan yang diperlukan secara manual setelah rollback selesai. Misalnya, jika OnNodeUpdated skrip mengubah status bidang dalam file konfigurasi (misalnya, dari true ke false) dan kemudian gagal, Anda perlu mengembalikan nilai bidang itu secara manual ke status pra-pembaruan (misalnya, false ke true). Untuk informasi selengkapnya, lihat [Tindakan bootstrap kustom](#).

Melihat kesalahan dengan Slurm konfigurasi khusus

Mulai AWS ParallelCluster versi 3.6.0, Anda tidak dapat lagi menargetkan satu prolog atau epilog skrip dengan memasukkannya ke dalam konfigurasi khusus. Slurm Di AWS ParallelCluster versi 3.6.0 dan versi yang lebih baru, Anda harus menemukan kustom prolog dan epilog skrip di masing-masing Prolog dan folder. Epilog Folder ini dikonfigurasi secara default untuk menunjuk ke:

- Prologmenunjuk ke/opt/slurm/etc/scripts/prolog.d/.
- Epilogmenunjuk ke/opt/slurm/etc/scripts/epilog.d/.

Kami menyarankan Anda menyimpan skrip `90_plcluster_health_check_manager` prolog dan skrip `90_pcluster_noop` epilog di tempatnya.

Slurmmenjalankan skrip dalam urutan abjad terbalik. Kedua folder Prolog dan Epilog folder harus berisi setidaknya satu file. Untuk informasi selengkapnya, silakan lihat [Slurm prolog dan epilog](#) dan [Slurmkustomisasi konfigurasi](#).

Alarm Cluster

Pemantauan kesehatan cluster sangat penting untuk memastikan kinerja yang optimal. AWS ParallelClustermemungkinkan Anda untuk memantau beberapa alarm CloudWatch berbasis untuk node kepala cluster.

Bagian ini memberikan detail untuk setiap jenis alarm kluster node Head termasuk konvensi penamaannya, kondisi spesifik yang memicu alarm, dan langkah-langkah pemecahan masalah yang disarankan.

Konvensi penamaan untuk alarm cluster adalah `CLUSTER_NAME-COMPONENT-METRIC`, mis. `mycluster-HeadNode-Cpu`

- `CLUSTER_NAME-HeadNode`: menandakan status keseluruhan simpul kepala. Warna merah jika setidaknya salah satu alarm di bawah ini.
- `CLUSTER_NAME-HeadNode-Health`: merah jika setidaknya ada satu kegagalan Pemeriksaan Kesehatan EC2. Jika terjadi alarm, kami sarankan untuk melihat [contoh Pemecahan Masalah dengan](#) pemeriksaan status yang gagal.
- `CLUSTER_NAME-HeadNode-Cpu`: merah jika pemanfaatan CPU lebih besar dari 90%. Jika terjadi alarm, periksa proses yang paling banyak mengkonsumsi CPUs `-aux --sort=-%cpu | head -n 10`.
- `CLUSTER_NAME-HeadNode-Mem`: merah jika pemanfaatan memori lebih besar dari 90%. Jika terjadi alarm, periksa proses yang paling banyak menghabiskan memori `ps -aux --sort=-%mem | head -n 10`.
- `CLUSTER_NAME-HeadNode-Disk`: merah jika ruang disk yang ditempati lebih besar dari 90% di jalur `/`. Dalam kasus alarm, periksa folder yang menghabiskan sebagian besar ruang dengan `df -h --max-depth=2 / 2> /dev/null | sort -hr`.

Dukungan tambahan

Untuk daftar masalah yang diketahui, lihat halaman [GitHub Wiki](#) utama atau halaman [masalah](#).

Untuk masalah yang lebih mendesak, hubungi AWS Support atau buka [GitHub masalah baru](#).

AWS ParallelCluster kebijakan dukungan

AWS ParallelCluster mendukung beberapa rilis pada saat yang sama. Setiap AWS ParallelCluster rilis memiliki tanggal End of Support Life (EOSL) yang dijadwalkan. Setelah tanggal EOSL, tidak ada dukungan atau pemeliharaan lebih lanjut yang diberikan untuk rilis tersebut.

AWS ParallelCluster menggunakan skema `major.minor.patch` versi. Fitur baru, peningkatan kinerja, pembaruan keamanan, dan perbaikan bug disertakan dalam rilis versi minor baru untuk rilis versi utama terbaru. Versi minor kompatibel ke belakang dalam versi utama. Untuk masalah kritis, AWS menyediakan perbaikan melalui rilis patch, tetapi hanya untuk versi minor terbaru dari rilis yang belum mencapai EOSL. Jika Anda ingin menggunakan pembaruan dari rilis versi baru, Anda perlu memutakhirkan ke versi minor atau patch yang baru.

AWS ParallelCluster versi	Tanggal akhir masa pakai yang didukung (EOSL)
3.0. <i>x</i>	3/31/2023
3.1. <i>x</i>	8/31/2023
3.2. <i>x</i>	1/31/2024
3.3. <i>x</i>	5/31/2024
3.4. <i>x</i>	6/28/2024
3.5. <i>x</i>	8/31/2024
3.6. <i>x</i>	11/30/2024
3.7. <i>x</i>	2/28/2025
3.8. <i>x</i>	6/30/2025
3.9. <i>x</i>	09/05/2025

Keamanan di AWS ParallelCluster

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

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. [Model tanggung jawab bersama](#) model menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di AWS Cloud. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi secara berkala efektivitas keamanan kami sebagai bagian dari [Program Kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku AWS ParallelCluster, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan atau layanan khusus yang Anda gunakan. Anda juga bertanggung jawab atas beberapa faktor terkait lainnya termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini menjelaskan bagaimana Anda harus menerapkan model tanggung jawab bersama saat menggunakan AWS ParallelCluster. Topik berikut menunjukkan cara mengonfigurasi AWS ParallelCluster untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar bagaimana menggunakan dengan AWS ParallelCluster cara yang membantu Anda memantau dan mengamankan AWS sumber daya Anda.

Topik

- [Informasi keamanan untuk layanan yang digunakan oleh AWS ParallelCluster](#)
- [Perlindungan data di AWS ParallelCluster](#)
- [Identity and Access Management untuk AWS ParallelCluster](#)
- [Validasi kepatuhan untuk AWS ParallelCluster](#)
- [Memberlakukan Versi Minimum dari TLS 1.2](#)

Informasi keamanan untuk layanan yang digunakan oleh AWS ParallelCluster

- [Keamanan di Amazon EC2](#)
- [Keamanan di Amazon API Gateway](#)
- [Keamanan di AWS Batch](#)
- [Keamanan di AWS CloudFormation](#)
- [Keamanan di Amazon CloudWatch](#)
- [Keamanan di AWS CodeBuild](#)
- [Keamanan di Amazon DynamoDB](#)
- [Keamanan di Amazon ECR](#)
- [Keamanan di Amazon ECS](#)
- [Keamanan di Amazon EFS](#)
- [Keamanan di FSx untuk Lustre](#)
- [Keamanan di AWS Identity and Access Management \(IAM\)](#)
- [Keamanan di EC2 Gambar Builder](#)
- [Keamanan di AWS Lambda](#)
- [Keamanan di Amazon Route 53](#)
- [Keamanan di Amazon SNS](#)
- [Keamanan di Amazon SQS \(Untuk AWS ParallelCluster versi 2.x.\)](#)
- [Keamanan di Amazon S3](#)
- [Keamanan di Amazon VPC](#)

Perlindungan data di AWS ParallelCluster

[Model tanggung jawab bersama](#) AWS diterapkan untuk perlindungan data AWS ParallelCluster. Sebagaimana dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Anda juga bertanggung jawab atas konfigurasi keamanan dan tugas manajemen untuk Layanan AWS yang Anda gunakan. Untuk informasi lebih lanjut tentang privasi data, lihat [FAQ tentang Privasi Data](#). Untuk informasi tentang

perlindungan data di Eropa, lihat postingan blog [Model Tanggung Jawab Bersama AWS dan GDPR](#) 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 tersebut, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tugas pekerjaan mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- 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 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat menyarankan agar Anda tidak pernah memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan Anda, ke dalam tag atau bidang teks bentuk bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS ParallelCluster atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tag atau bidang teks bentuk bebas yang digunakan untuk nama dapat digunakan untuk penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, sebaiknya jangan menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi data

Fitur utama dari setiap layanan aman adalah bahwa informasi dienkripsi ketika tidak aktif digunakan.

Enkripsi saat tidak aktif

AWS ParallelCluster tidak menyimpan data pelanggan selain kredensial yang dibutuhkan untuk berinteraksi dengan AWS layanan atas nama pengguna.

Untuk data pada node di cluster, data dapat dienkripsi saat istirahat.

Untuk volume Amazon EBS, enkripsi dikonfigurasi menggunakan `KmsKeyId` pengaturan [EbsSettingsEbsSettings/Encrypted](#) dan di [EbsSettings](#) bagian. Untuk informasi selengkapnya, lihat [enkripsi Amazon EBS](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk volume Amazon EFS, enkripsi dikonfigurasi menggunakan `KmsKeyId` pengaturan [EfsSettingsEfsSettings/Encrypted](#) dan di [EfsSettings](#) bagian. Untuk informasi selengkapnya, lihat [Cara kerja enkripsi saat istirahat](#) di Panduan Pengguna Amazon Elastic File System.

Untuk sistem file FSx for Lustre, enkripsi data saat istirahat diaktifkan secara otomatis saat membuat sistem file Amazon FSx. Untuk informasi selengkapnya, lihat [Mengenkripsi data saat istirahat di Panduan](#) Pengguna Amazon FSx for Lustre.

Misalnya tipe dengan volume NVMe, data pada volume penyimpanan instance NVMe dienkripsi menggunakan cipher XTS-AES-256 yang diimplementasikan pada modul perangkat keras pada instance. Kunci enkripsi dibuat menggunakan modul perangkat keras dan unik untuk setiap perangkat penyimpanan instans NVMe. Semua kunci enkripsi akan dihancurkan ketika instans dihentikan atau dihapus dan tidak dapat dipulihkan. Anda tidak dapat menonaktifkan enkripsi ini dan Anda tidak dapat menyediakan kunci enkripsi Anda sendiri. Untuk informasi selengkapnya, lihat [Enkripsi saat istirahat](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda menggunakan AWS ParallelCluster AWS layanan yang mentransmisikan data pelanggan ke komputer lokal Anda untuk penyimpanan, lihat bagian Keamanan dan Kepatuhan dalam Panduan Pengguna layanan tersebut untuk informasi tentang cara data tersebut disimpan, dilindungi, dan dienkripsi.

Enkripsi dalam transit

Secara default, semua data yang dikirimkan dari komputer klien yang berjalan AWS ParallelCluster dan titik akhir AWS layanan dienkripsi dengan mengirimkan semuanya melalui koneksi HTTPS/TLS. Lalu lintas antar node di cluster dapat dienkripsi secara otomatis, tergantung pada jenis instance yang dipilih. Untuk informasi selengkapnya, lihat [Enkripsi saat transit di](#) Panduan Pengguna Amazon EC2 untuk Instans Linux.

Lihat juga

- [Perlindungan data di Amazon EC2](#)

- [Perlindungan data di EC2 Image Builder](#)
- [Perlindungan data di AWS CloudFormation](#)
- [Perlindungan data di Amazon EFS](#)
- [Perlindungan data di Amazon S3](#)
- [Perlindungan data di FSx for Lustre](#)

Identity and Access Management untuk AWS ParallelCluster

AWS ParallelCluster menggunakan peran untuk mengakses AWS sumber daya Anda dan layanan mereka. Kebijakan instans dan pengguna yang AWS ParallelCluster digunakan untuk memberikan izin didokumentasikan di [AWS Identity and Access Management izin di AWS ParallelCluster](#).

Satu-satunya perbedaan utama adalah bagaimana Anda mengautentikasi saat menggunakan pengguna standar dan kredensial jangka panjang. Meskipun pengguna memerlukan kata sandi untuk mengakses konsol AWS layanan, pengguna yang sama memerlukan pasangan kunci akses untuk melakukan operasi yang sama dengan menggunakan AWS ParallelCluster. Semua kredensial jangka pendek lainnya digunakan dengan cara yang sama dengan yang digunakan bersama konsol.

Kredensial yang digunakan oleh AWS ParallelCluster disimpan dalam file teks biasa dan tidak dienkripsi.

- File `$HOME/.aws/credentials` menyimpan kredensial jangka panjang yang diperlukan untuk mengakses sumber daya AWS Anda. Ini termasuk access key ID dan secret access key Anda.
- Kredensial jangka pendek, seperti untuk peran yang Anda asumsikan, atau yang untuk AWS IAM Identity Center layanan, juga disimpan dalam folder `$HOME/.aws/cli/cache` dan `$HOME/.aws/SSO/Cache`, masing-masing.

Mitigasi Risiko

- Kami sangat merekomendasikan agar Anda mengkonfigurasi izin sistem file pada direktori `$HOME/.aws` dan direktori dan file anaknya untuk membatasi akses hanya untuk pengguna yang berwenang.
- Gunakan peran dengan kredensial sementara sedapat mungkin untuk mengurangi peluang terjadinya kerusakan jika kredensialnya dikurangi. Gunakan kredensial jangka panjang hanya untuk meminta dan menyegarkan kredensial peran jangka pendek.

Validasi kepatuhan untuk AWS ParallelCluster

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS sebagai bagian dari beberapa program kepatuhan AWS. Menggunakan AWS ParallelCluster untuk mengakses layanan tidak akan mengubah kepatuhan layanan tersebut.

Untuk daftar layanan AWS dalam cakupan program kepatuhan tertentu, lihat [AWS layanan dalam cakupan menurut program kepatuhan](#). Untuk informasi umum, lihat [AWS program kepatuhan](#).

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS ParallelCluster ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu dengan kepatuhan:

- [Panduan quick start keamanan dan kepatuhan](#) – Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS.
- [Laporan resmi perancangan untuk keamanan dan kepatuhan AWS HIPAA — Laporan resmi](#) ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang mematuhi HIPAA.
- [AWS sumber daya kepatuhan](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi sumber daya dengan aturan](#) dalam AWS Config Panduan Developer – Layanan AWS Config akan menilai seberapa patuh konfigurasi sumber daya Anda terhadap praktik internal, panduan industri, dan aturan.
- [AWS Security Hub](#) – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda dalam AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Memberlakukan Versi Minimum dari TLS 1.2

Untuk menambahkan peningkatan keamanan saat berkomunikasi dengan AWS layanan, Anda harus mengkonfigurasi AWS ParallelCluster untuk menggunakan TLS 1.2 atau yang lebih baru. Saat Anda menggunakan AWS ParallelCluster, Python digunakan untuk mengatur versi TLS.

Untuk memastikan tidak AWS ParallelCluster menggunakan versi TLS lebih awal dari TLS 1.2, Anda mungkin perlu mengkompilasi ulang OpenSSL untuk menegakkan minimum ini dan kemudian mengkompilasi ulang Python untuk menggunakan OpenSSL yang baru dibangun.

Tentukan Protokol yang Didukung Saat Ini

Pertama, buat sertifikat yang ditandatangani sendiri untuk digunakan untuk server pengujian dan SDK Python menggunakan OpenSSL.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Kemudian putar server uji menggunakan OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

Di jendela terminal baru, buat lingkungan virtual dan instal Python SDK.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Buat skrip Python baru bernama `check.py` yang menggunakan pustaka HTTP yang mendasari SDK.

```
$ import urllib3
URL = 'https://localhost:4433/'

http = urllib3.PoolManager(
    ca_certs='cert.pem',
    cert_reqs='CERT_REQUIRED',
)
r = http.request('GET', URL)
print(r.data.decode('utf-8'))
```

Jalankan skrip baru Anda.

```
$ python check.py
```

Ini menampilkan detail tentang koneksi yang dibuat. Cari "Protocol:" di output. Jika output adalah "TLSv1.2" atau yang lebih baru, SDK default ke TLS v1.2 atau yang lebih baru. Jika versi sebelumnya, Anda perlu mengkompilasi ulang OpenSSL dan mengkompilasi ulang Python.

Namun, meskipun instalasi Python Anda secara default ke TLS v1.2 atau yang lebih baru, Python masih memungkinkan untuk menegosiasikan ulang ke versi yang lebih awal dari TLS v1.2 jika server tidak mendukung TLS v1.2 atau yang lebih baru. Untuk memeriksa bahwa Python tidak secara otomatis menegosiasikan ulang ke versi sebelumnya, restart server pengujian dengan yang berikut ini.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Jika Anda menggunakan versi sebelumnya dari OpenSSL, Anda mungkin tidak memiliki `-no_tls1_3` bendera yang tersedia. Jika ini masalahnya, hapus flag karena versi OpenSSL yang Anda gunakan tidak mendukung TLS v1.3. Kemudian jalankan kembali skrip Python.

```
$ python check.py
```

Jika instalasi Python Anda dengan benar tidak menegosiasikan ulang untuk versi yang lebih awal dari TLS 1.2, Anda akan menerima kesalahan SSL.

```
$ urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='localhost',  
port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL:  
UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

Jika Anda dapat membuat koneksi, Anda perlu mengkompilasi ulang OpenSSL dan Python untuk menonaktifkan negosiasi protokol lebih awal dari TLS v1.2.

Kompilasi OpenSSL dan Python

Untuk memastikan bahwa AWS ParallelCluster tidak bernegosiasi untuk apa pun lebih awal dari TLS 1.2, Anda perlu mengkompilasi ulang OpenSSL dan Python. Untuk melakukan ini, salin konten berikut untuk membuat skrip dan menjalankannya.

```
#!/usr/bin/env bash  
set -e  
  
OPENSSL_VERSION="1.1.1d"  
OPENSSL_PREFIX="/opt/openssl-with-min-tls1_2"
```

```
PYTHON_VERSION="3.8.1"
PYTHON_PREFIX="/opt/python-with-min-tls1_2"

curl -O "https://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz"
tar -xzf "openssl-$OPENSSL_VERSION.tar.gz"
cd openssl-$OPENSSL_VERSION
./config --prefix=$OPENSSL_PREFIX no-ssl3 no-tls1 no-tls1_1 no-shared
make > /dev/null
sudo make install_sw > /dev/null

cd /tmp
curl -O "https://www.python.org/ftp/python/$PYTHON_VERSION/Python-$PYTHON_VERSION.tgz"
tar -xzf "Python-$PYTHON_VERSION.tgz"
cd Python-$PYTHON_VERSION
./configure --prefix=$PYTHON_PREFIX --with-openssl=$OPENSSL_PREFIX --disable-shared > /dev/null
make > /dev/null
sudo make install > /dev/null
```

Ini mengkompilasi versi Python yang memiliki OpenSSL yang terhubung secara statis yang tidak secara otomatis menegosiasikan apa pun lebih awal dari TLS 1.2. Ini juga menginstal OpenSSL di/opt/openssl-with-min-tls1_2 direktori dan menginstal Python di/opt/python-with-min-tls1_2 direktori. Setelah Anda menjalankan skrip ini, konfirmasi penginstalan versi baru Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Ini harus mencetak berikut ini.

```
Python 3.8.1
```

Untuk mengonfirmasi versi baru Python ini tidak menegosiasikan versi lebih awal dari TLS 1.2, jalankan kembali langkah-langkah dari [Tentukan Protokol yang Didukung Saat Ini](#) menggunakan versi Python yang baru diinstal (yaitu, /opt/python-with-min-tls1_2/bin/python3).

Catatan rilis dan riwayat dokumen

Tabel berikut menjelaskan pembaruan utama dan fitur baru untuk Panduan AWS ParallelCluster Pengguna. Kami juga rutin memperbarui dokumentasi untuk menjawab umpan balik yang Anda kirimkan kepada kami.

Perubahan	Deskripsi	Tanggal
AWS ParallelCluster versi 3.9.0 dirilis	<p>Kami sangat senang mengumumkan rilis 3.9.0 AWS ParallelCluster</p> <p>Untuk memutakhirkan, masukkan yang berikut ini:</p> <pre>sudo pip install --upgrade aws-parallelcluster</pre> <p>Penyempurnaan:</p> <ul style="list-style-type: none">• Tambahkan parameter konfigurasi <code>DeploymentSettings/DefaultUserHome</code> untuk memungkinkan pengguna memindahkan direktori home pengguna default ke <code>/local/home</code> bukan <code>/home</code> (default).• Izin untuk memperbarui <code>MinCountMaxCount</code>, <code>Queue</code> dan parameter <code>ComputeResource</code> konfigurasi tanpa perlu menghentikan armada komputasi. Sekarang mungkin untuk	Maret 12, 2024

memperbaruinya dengan menyetel `Scheduling/SlurmSettings/QueueUpdateStrategy` ke `TERMINATE`. AWS ParallelCluster hanya akan menghentikan node yang dihapus selama perubahan ukuran kapasitas cluster yang dilakukan melalui pembaruan cluster.

- Izinkan untuk memperbarui penyimpanan bersama eksternal tipe `Efs`, `FsxLustre`, `FsxOntap`, `FsxOpenZfs` dan `FileCache` tanpa mengganti armada komputasi dan login.
- Tambahkan dukungan untuk `RHEL9`.
- Tambahkan dukungan untuk `Rocky Linux 9` seperti yang `CustomAmi` dibuat melalui `build-image` proses. Tidak ada resmi publik AWS ParallelCluster `Rocky9` Linux AMI yang tersedia saat ini.
- Hapus `CommunicationParameters` dari daftar penolakan `Slurm` Pengaturan Kustom.
- Tambahkan `DeploymentSettings/DisableSudoAccessForDefault`

`tUser` parameter untuk menonaktifkan akses sudo pengguna default di OS yang didukung.

- Perubahan pada sistem file FSx for Lustre yang ParallelCluster dibuat oleh: Ubah versi server Lustre menjadi 2.15.
- Tambahkan kemungkinan untuk memilih antara Driver Nvidia Sumber Terbuka dan Tertutup saat membangun AMI, melalui atribut node `['cluster']['nvidia']['kernel_open']` buku masak.
- * Tambahkan opsi konfigurasi `clustermgtd ec2_instance_missing_max_count` untuk memungkinkan jumlah percobaan ulang yang dapat dikonfigurasi untuk EC2 akhirnya menggambarkan konsistensi instance dengan instance run.

Perubahan

- Tingkatkan Slurm ke 23.11.4 (dari 23.02.7).
- Tingkatkan driver NVIDIA ke versi 535.154.05.

- Tambahkan dukungan untuk Python 3.11, 3.12 di pcluster CLI dan. aws-parallelcluster-batch-cli
- Bangun antarmuka jaringan menggunakan indeks kartu jaringan dari NetworkCardIndex daftar DescribeInstances respons EC2, alih-alih mengulang rentang. MaximumNetworkCards
- Gagal pembuatan cluster saat menggunakan tipe instance P3, G3, P2 dan G2 karena arsitektur GPU mereka tidak kompatibel dengan Open Source Nvidia Drivers (OpenRM) yang diperkenalkan sebagai bagian dari rilis 3.8.0.
- Tingkatkan dependensi buku masak pihak ketiga: nfs-5.1.2 (dari nfs-5.0.0)
- Tingkatkan penginstal EFA ke 1.30.0.
 - Pengemudi EFA: efa-2.6.0-1
 - Konfigurasi EFA: efa-config-1.15-1
 - Profil EFA: efa-profile-1.6-1
 - Libfabric-AWS: libfabric-aws-1.19.0

- RDMA-inti: `rdma-core-46.0-1`
- Buka MPI: `openmpi40-aws-4.1.6-2` dan `openmpi50-aws-5.0.0-11`
- Tingkatkan NICE DCV ke versi `2023.1-16388`.
 - server: `2023.1.16388-1`
 - xdcv: `2023.1.565-1`
 - gl: `2023.1.1047-1`
 - penampil web: `2023.1.16388-1`

Perbaiki bug

- Perbaiki masalah yang membuat pekerjaan gagal saat dikirimkan sebagai pengguna direktori aktif dari node login. Masalah ini disebabkan oleh konfigurasi integrasi yang tidak lengkap dengan Direktori Aktif eksternal pada node kepala.
- Refactor kebijakan IAM yang ditentukan dalam CloudFormation template `parallelcluster-policies.yaml` untuk mencegah kegagalan penerapan API yang disebabkan oleh kebijakan

yang melebihi batas IAM.

ParallelCluster

- Perbaiki masalah membuat node login gagal bootstrap ketika node kepala membutuhkan waktu lebih lama dari yang diharapkan dalam menulis kunci.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster Versi UI 2024.03.0 dirilis](#)

AWS ParallelCluster Versi UI
2024.03.0 dirilis.

Maret 12, 2024

Fitur-fitur:

- Ditambahkan dukungan untuk AWS ParallelCluster versi 3.9.0
- Menambahkan dukungan untuk Ubuntu 22.04 dan Red Hat Enterprise Linux 9
- Ubuntu 18.04 yang tidak digunakan lagi

Perbaikan bug

- Memperbaiki masalah yang menyebabkan beberapa cluster tidak muncul saat menggunakan banyak cluster

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster Versi UI
2024.02.0 dirilis](#)

AWS ParallelCluster Versi UI
2024.02.0 dirilis

Februari 8, 2024

Perubahan:

- Memperbarui lingkungan runtime Lambda ke Python v3.9

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster Versi UI
2023.12.0 dirilis](#)

AWS ParallelCluster Versi UI
2023.12.0 dirilis.

21 Desember 2023

Fitur-fitur:

- Ditambahkan dukungan untuk penyebaran PCUI dengan jaringan pribadi.
- Menambahkan kemungkinan untuk secara opsional menerapkan Batas Izin ke setiap peran IAM yang dibuat oleh infrastruktur PCUI dan PCAPI
- Menambahkan kemungkinan untuk menerapkan awalan secara opsional ke setiap peran dan kebijakan IAM yang dibuat oleh infrastruktur PCUI dan PCAPI.
- Menambahkan dukungan untuk ParallelCluster versi 3.8.0, tanpa paritas fitur di wizard.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster versi 3.8.0 dirilis](#)

AWS ParallelCluster versi 3.8.0 dirilis.

Desember 19, 2023

Penyempurnaan:

- Tambahkan dukungan untuk Blok Kapasitas EC2 untuk ML.
- Tambahkan dukungan untuk Rocky Linux 8 seperti yang CustomAmi dibuat melalui build-image proses. Tidak ada resmi publik AWS ParallelCluster Rocky8 Linux AMI yang tersedia saat ini.
- Tambahkan Scheduling/ScalingStrategy parameter untuk mengontrol strategi penskalaan cluster yang akan digunakan saat meluncurkan instans EC2 untuk node komputasi Slurm. Nilai yang mungkin adalah all-or-nothing greedy-all-or-nothing best-effort ,, dengan all-or-nothing menjadi default.
- Tambahkan HeadNode/SharedStorageType parameter untuk menggunakan penyimpanan EFS alih-alih ekspor

NFS dari volume root node kepala untuk sumber daya sistem file bersama intra-cluster: ParallelCluster, Intel, Slurm, dan data. /home Peningkatan ini mengurangi beban pada jaringan node kepala.

- Izinkan pemasangan /home sebagai penyimpanan bersama eksternal EFS atau FSx melalui SharedStorage bagian file konfigurasi.
- Tambahkan parameter baru SlurmSettings/MungeKeySecretArn untuk mengizinkan penggunaan kunci MUNGE yang ditentukan pengguna eksternal dari Secrets Manager AWS .
- Tambahkan Monitoring/Alarms/Enabled parameter untuk mengaktifkan CloudWatch Alarm Amazon untuk klaster.
- Tambahkan alarm node kepala untuk memantau pemeriksaan kesehatan EC2, pemanfaatan CPU, dan status keseluruhan node kepala, dan tambahkan ke CloudWatch Dasbor yang dibuat dengan cluster.

- Tambahkan dukungan untuk Asosiasi Repositori Data saat menggunakan `PERSISTENT_2` as `DeploymentType` untuk FSx for Lustre yang dikelola.
- Tambahkan `Scheduling/SlurmSettings/Database/DatabaseName` parameter untuk memungkinkan pengguna menentukan nama kustom untuk database pada server database yang akan digunakan untuk akuntansi Slurm.
- `InstanceType` Buat parameter konfigurasi opsional saat mengonfigurasi sumber `CapacityReservationTarget/CapacityReservationId` daya komputasi.
- Tambahkan kemungkinan untuk menentukan awalan untuk peran dan kebijakan IAM yang dibuat oleh AWS ParallelCluster API.
- Tambahkan kemungkinan untuk menentukan batas izin yang akan diterapkan untuk peran dan kebijakan IAM yang dibuat oleh API. AWS ParallelCluster

Perubahan

- Tingkatkan Slurm ke 23.02.7 (dari 23.02.6).
- Tingkatkan driver NVIDIA ke versi 535.129.03.
- Tingkatkan CUDA Toolkit ke versi 12.2.2.
- Gunakan driver GPU NVIDIA Open Source (OpenRM) sebagai modul kernel NVIDIA untuk Linux alih-alih modul sumber tertutup NVIDIA.
- Hapus dukungan parameter `all_or_nothing_batch` konfigurasi dalam program resume Slurm, demi konfigurasi Scheduling/Scaling Strategy cluster baru.
- Mengubah konvensi penamaan alarm cluster menjadi '[cluster-name] - [component-name] - [metric]' .
- Ubah tipe volume EBS default di wilayah ADC dari gp2 ke gp3, untuk volume root dan tambahan.
- Batas izin opsional untuk AWS ParallelCluster API sekarang diterapkan ke setiap peran IAM yang dibuat oleh infrastruktur API.

- Tingkatkan penginstal EFA ke. 1.29.1
- Pengemudi EFA: efa-2.6.0-1
- Konfigurasi EFA: efa-config-1.15-1
- Profil EFA: efa-profile-1.5-1
- Libfabric-AWS: libfabric-aws-1.19.0-1
- RDMA-inti: rdma-core-46.0-1
- Buka MPI: openmpi40-aws-4.1.6-1
- Tingkatkan GDRCopy ke versi 2.4 di semua OS yang didukung, kecuali untuk Centos 7 di mana versi 2.3.1 digunakan.
- Tingkatkan aws-cfn-bootstrap ke versi 2.0-28.
- Tambahkan dukungan untuk Python 3.10 in. aws-parallelcluster-batch-cli

Perbaiki bug

- Perbaiki konfigurasi penskalaan yang tidak konsisten setelah rollback pembaruan cluster saat memodifikasi daftar jenis instance yang dideklara

sikan dalam Sumber Daya Komputasi.

- Perbaiki pengguna pembuatan kunci SSH saat mengganti pengguna tanpa hak akses root di cluster yang terintegrasi dengan server LDAP eksternal melalui file konfigurasi cluster.
- Perbaiki menonaktifkan mode hemat daya Slurm saat mengatur. `Scaledown Idletime = -1`
- Perbaiki jalur hard-code ke direktori instalasi Slurm dalam `update_slurm_database_password.sh` skrip untuk Slurm Accounting.

[AWS ParallelCluster versi 3.7.2 dirilis](#)

AWS ParallelCluster versi 3.7.2 dirilis.

25 Oktober 2023

Perubahan:

- Tingkatkan Slurm ke 23.02.6.

[AWS ParallelCluster Versi UI 2023.10.0 dirilis](#)

AWS ParallelCluster Versi UI 2023.10.0 dirilis.

20 Oktober 2023

Fitur-fitur:

- Menambahkan dukungan untuk ParallelCluster 3.7.2 dengan paritas fitur di wizard terbatas pada FSx File Cache dan kompatibilitas penjadwalan berbasis memori dengan beberapa jenis instance.

Perbaikan bug:

- Memperbaiki masalah yang menyebabkan kesalahan UI saat PCUI tidak memiliki izin untuk berinteraksi dengan Cost Explorer.

Perbaikan

- Meningkatkan keamanan dengan mengurangi token akses TTL dari 10 menit menjadi 5 menit.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster versi
3.7.1 dirilis](#)

AWS ParallelCluster versi
3.7.1 dirilis.

September 22, 2023

Perubahan:

- Tingkatkan Slurm ke 23.02.5 (dari 23.02.4).
 - Tingkatkan Pmix ke 4.2.6 (dari 3.2.3).
 - Tingkatkan libjwt ke 1.15.3 (dari 1.12.0).
- Tingkatkan penginstal EFA ke 1.26.1, memperbaiki masalah RDMA writedata di P5.
 - Pengemudi EFA: efa-2.5.0-1
 - Konfigurasi EFA: efa-config-1.15-1
 - Profil EFA: efa-profile-1.5-1
 - Libfabric-AWS: libfabric-aws-1.18.2-1
 - ERDMA-inti: rdma-core-46.0-1
 - Buka MPI: openmpi40-aws-4.1.5-4 .

[AWS ParallelCluster versi 3.7.0 dirilis](#)

AWS ParallelCluster versi 3.7.0 dirilis.

Agustus 30, 2023

Penyempurnaan:

- Support konfigurasi prioritas node statis dan dinamis dalam sumber daya komputasi dengan menggunakan file YAMM AWS ParallelCluster konfigurasi.
- Tambahkan dukungan untuk Ubuntu 22. Kunci RSA tidak didukung secara default.
- Tambahkan pengaturan konfigurasi antrian `JobExclusiveAllocation` untuk mengalokasikan node dalam partisi secara eksklusif ke satu pekerjaan pada waktu tertentu.
- Izinkan `aws-parallelcluster-node` paket Override pada waktu pembuatan cluster dan pembaruan cluster. Untuk node kepala, ini berlaku untuk pembaruan cluster. Berguna untuk tujuan pengembangan saja.
- Hindari server NFS dimulai pada node komputasi.

- Tambahkan dukungan untuk node log-in.
- Izinkan penjadwalan berbasis memori ketika beberapa jenis instans ditentukan untuk Sumber Daya Komputasi Slurm.
- Tambahkan dukungan untuk memasang Cache File Amazon yang ada sebagai penyimpanan bersama.

Perubahan:

- Tetapkan node Slurm dinamis prioritas (bobot) 1000 secara default. Dengan melakukan ini, Slurm dapat memprioritaskan node statis idle atas node dinamis idle.
- Buat `aws-parallelcluster-node` daemon hanya menangani partisi yang AWS ParallelCluster dikelola Slurm.
- `EFS-utils` Tingkatkan interval polling pengawas menjadi 10 detik. Perubahan ini berlaku ketika `EncryptionInTransit` disetel ke `true`, yang merupakan satu-satunya kondisi yang menyebabkan pengawas berjalan.

- Tingkatkan penginstal EFA ke. 1.25.1
 - EFA-driver: efa-2.5.0-1 (dari efa-2.1.1g)
 - EFA-config: efa-config-1.15-1 (dari efa-config-1.13-1)
 - Profil EFA: efa-profile-1.5-1 (tidak ada perubahan)
 - libFabric-aws: libfabric-aws-1.18.1-0 (dari libfabric-aws-1.17.1-1)
 - RDMA-core: rdma-core-46.0-1 (dari rdma-core-43.0-1)
 - Buka MPI: openmpi40-aws-4.1.5-4 (dari openmpi40-aws-4.1.5-1)
- Tingkatkan Slurm ke versi 23.02.4.
- Ubah nilai default lmds/lmdsSupport dari v1.0 ke v2.0.
- Ubuntu 18 tidak digunakan lagi.
- Perbarui ukuran volume root default menjadi 40 GB untuk memperhitungkan batasan pada Centos 7.
- Batasi izin pada file /tmp/wait_condition_handle.

txt di dalam node kepala sehingga hanya root yang dapat membacanya.

- Buat file JSON pemetaan partisi-nodelist Slurm untuk digunakan oleh daemon paket node untuk mengenali partisi dan nodelist Slurm yang dikelola PC.
- Tingkatkan driver NVIDIA ke versi 535.54.03.
- Tingkatkan pustaka CUDA ke versi 12.2.0.
- Tingkatkan manajer NVIDIA Fabric ke nvidia-fabricmanager-535.
- Upgrade ARM PL ke versi 23.04.1 hanya untuk Ubuntu 22.04.
- Tingkatkan NICE DCV ke versi. 2023.0-15487
 - Peladen: 2023.0.15487-1
 - xdcv: 2023.0.551-1
 - gl: 2023.0.1039-1
 - penampil web: 2023.0.15487-1

Perbaiki bug:

- Tambahkan validasi ke ScaledownIdleTime nilai, untuk mencegah

pengaturan nilai yang lebih rendah dari -1.

- Perbaiki kegagalan pembuatan cluster dengan Ubuntu Deep Learning AMI pada instance GPU dengan DCV diaktifkan.
- Perbaiki masalah yang menyebabkan kebijakan IAM yang menggantung dibuat saat membuat penyedia sumber daya ParallelCluster CloudFormation khusus dengan CustomLambdaRole
- Memperbaiki masalah yang menyebabkan ketidaksejajaran nama DNS node komputasi pada instance dengan beberapa antarmuka jaringan, saat menggunakan sama dengan `SlurmSettings/Dns/UseEc2Hostnames`
`True`

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[Dokumentasi hanya rilis](#)

AWS ParallelCluster versi 3 panduan pengguna khusus diterbitkan.

Juli 17, 2023

Hanya rilis dokumentasi:

- AWS ParallelCluster versi 3 memiliki panduan pengguna tersendiri.

[AWS ParallelCluster versi
3.6.1 dirilis](#)

AWS ParallelCluster versi
3.6.1 dirilis.

5 Juli 2023

Perubahan:

- Hindari duplikasi node yang dilihat oleh `clustermg` `td` jika node komputasi ditambahkan ke beberapa Slurm partisi.

Perbaikan bug:

- Hapus hard coding nama perangkat volume root (`/dev/sda1` dan `/dev/xvda`) dan ambil dari AMI yang digunakan selama `create-cluster`.
- Perbaiki kegagalan pembuatan cluster saat menggunakan sumber daya CloudFormation khusus dengan `ElasticIp` disetel ke `True`.
- Perbaiki kegagalan pembuatan dan perbarui cluster saat menggunakan sumber daya AWS CloudFormation khusus dengan file konfigurasi besar.
- Perbaiki masalah yang mencegah `pttrace`

perlindungan dinonaktifkan di Ubuntu dan yang tidak mengizinkan Cross Memory Attach (CMA) di libfabric.

- Perbaiki logika fail-over kapasitas yang tidak mencukupi dengan cepat saat menggunakan beberapa jenis instans dan tidak ada instance yang dikembalikan.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster Versi UI
2023.06.0 dirilis](#)

AWS ParallelCluster Versi UI
2023.06.0 dirilis.

Juni 7, 2023

Perubahan:

- Memutakhirkan versi AWS ParallelCluster API default ke 3.6.0.

Perbaikan bug:

- Memperbaiki penyebaran yang rusak untuk Wilayah AWS GovCloud (AS-Barat).
- Panel split sekarang memuat detail cluster dengan benar setelah pembuatan dimulai.

Catatan:

- Fitur Pemantauan Biaya tidak tersedia di AWS GovCloud (US) Regions.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster versi 3.6.0 dirilis](#)

AWS ParallelCluster versi 3.6.0 dirilis.

22 Mei 2023

Dokumentasi:

- Tambahkan dokumentasi untuk [AWS ParallelCluster API pustaka Python](#).

Penyempurnaan:

- Tambahkan dukungan untuk RHEL8.
- Tambahkan [sumber daya AWS CloudFormation khusus](#) untuk membuat dan mengelola cluster dengan CloudFormation.
- Tambahkan dukungan untuk [menyesuaikan Slurm konfigurasi cluster](#) dalam file AWS ParallelCluster YAMM konfigurasi.
- Bangun Slurm dengan dukungan untuk LUA.
- Tingkatkan batas jumlah antrian maksimum per cluster dari 10 menjadi 50. Setiap antrian dapat memiliki hingga 50 sumber daya komputasi. Setiap cluster dapat memiliki hingga 50 sumber daya komputasi.

- Tambahkan dukungan untuk menentukan urutan beberapa [skrip tindakan kustom](#) untuk acara yang dikonfigurasi dalam `OnNodeStart`, `OnNodeConfigured`, dan `OnNodeUpdated` parameter.
- Tambahkan bagian konfigurasi baru `HealthChecks /Gpu`, untuk menerapkan pemeriksaan kesehatan GPU pada node komputasi sebelum pekerjaan dijalankan.
- Tambahkan dukungan untuk `Tags ComputeResources` konfigurasi `SlurmQueues` dan `SlurmQueues /`.
- Tambahkan dukungan untuk [DetailedMonitoring](#) dalam `Monitoring` konfigurasi.
- Tambahkan `mem_used_percent` dan `disk_used_percent` metrik untuk memori node kepala dan pelacakan pemanfaatan disk volume root di AWS ParallelCluster [CloudWatch dashboard](#), dan atur alarm untuk memantau metrik ini.

- Tambahkan dukungan [rotasi log](#) untuk log AWS ParallelCluster terkelola.
- [Lacak kesalahan node komputasi umum dan waktu idle terpanjang node dinamis di Dasbor CloudWatch](#)
- Menerapkan DCV Authenticator Server untuk menggunakan setidaknya a TLS-1.2 protokol saat membuat SSL Socket.
- Instal paket [NVIDIA Data Center GPU Manager \(DCGM\)](#) pada semua sistem operasi yang didukung kecuali dan. aarch64 centos7 alinux2
- Muat modul kernel [nvidia-vm](#) secara default untuk menyediakan fungsionalitas Unified Virtual Memory (UVM) ke driver CUDA.
- Instal [NVIDIA Persistence Daemon](#) sebagai layanan sistem.

Perubahan:

- Tingkatkan Slurm ke versi 23.02.2 (dari versi 22.05.8).
- Tingkatkan munge ke versi 0.5.15 (dari versi 0.5.14).

- Atur Slurm `TreeWidth` ke 30.
- Atur `epilog` konfigurasi Slurm `prolog` dan ke direktori target `/opt/slurm/etc/scripts/prolog.d/` dan `/opt/slurm/etc/scripts/epilog.d/` masing-masing.
- Setel Slurm `BatchStartTimeout` ke maksimum 3 menit untuk menjalankan `Prolog` skrip selama pendaftaran node komputasi.
- Tingkatkan default `CloudWatch log Retention InDays` dari 14 menjadi 180 hari.
- Tingkatkan penginstal EFA ke 1.22.1
 - `Dkms`: 2.8.3-2
 - `EFA-driver`: `efa-2.1.1g` (tidak ada perubahan)
 - `EFA-config`: `efa-config-1.13-1` (tidak ada perubahan)
 - `Profil EFA`: `efa-profile-1.5-1` (tidak ada perubahan)
 - `libFabric-aws`:
`libfabric-aws-1.17.1-1` (dari `libfabric-aws-1.17.0-1`)

- RDMA-core: `rdma-core-43.0-1` (tidak ada perubahan)
- Buka MPI: `openmpi40-aws-4.1.5-1` (tidak ada perubahan)
- Tingkatkan versi klien Lustre ke di 2.12 Amazon Linux 2. Klien Lustre 2.12 telah diinstal pada Ubuntu 20.04, 18.04, dan CentOS ≥ 7.7 .
- Tingkatkan versi klien Lustre ke 2.10.8 CentOS 7.6.
- Tingkatkan driver NVIDIA ke versi 470.182.03 (dari versi 470.141.03).
- Tingkatkan NVIDIA Fabric Manager ke versi 470.182.03 (dari versi 470.141.03).
- Tingkatkan NVIDIA CUDA Toolkit ke versi 11.8.0 (dari versi 11.7.1).
- Tingkatkan sampel NVIDIA CUDA ke versi 11.8.0.
- Tingkatkan Perpustakaan Intel MPI ke Pembaruan Versi 2021 9 (dari Pembaruan Versi 2021 6). Untuk informasi selengkapnya, lihat [Pembaruan Perpustakaan Intel® MPI 2021 9](#).

- Tingkatkan NICE DCV ke versi 2023.0-15022 (dari versi 2022.2-14521).
 - server: 2023.0.15022-1 (dari versi 2022.2-14521-1).
 - xdcv: 2023.0.547-1 (dari versi 2022.2.519-1).
 - gl: 2023.0.1027-1 (dari versi 2022.2.1012-1).
 - web_viewer: 2023.0.15022-1 (dari versi 2022.2.14521-1).
- Tingkatkan aws-cfn-bootstrap ke versi 2.0-24.
- Tingkatkan gambar yang digunakan oleh CodeBuild lingkungan saat membuat gambar kontainer untuk AWS Batch cluster:
 - aws/codebuild/amazonlinux2-x86_64-standard:4.0 (dari aws/codebuild/amazonlinux2-x86_64-standard:3.0).
 - aws/codebuild/amazonlinux2-aarch64-standard:2.0 (dari aws/codebuild/amazonlinux2-aarch64-standard:1.0).

```
onlinux2-aarch64-  
standard:1.0 ).
```

Perbaiki bug:

- Perbaiki Amazon EFS dan validator grup keamanan jaringan Amazon FSx untuk menghindari pelaporan kesalahan palsu.
- Perbaiki penandaan sumber daya yang hilang yang dibuat oleh Image Builder selama `build-image` operasi berlangsung.
- Perbaiki kebijakan pembaruan `MaxCount` untuk selalu melakukan perbandingan numerik pada properti `MaxCount`
- Perbaiki penyelarasan IP pada instance node komputasi dengan beberapa kartu jaringan.
- Perbaiki pengganti `StoragePass` di `slurm_parallelcluster_slurmdbd.conf` saat pembaruan parameter antrian dilakukan dan konfigurasi Slurm akuntansi tidak diperbarui.
- Perbaiki masalah yang menyebabkan grup keamanan menggantung

dibuat saat membuat cluster dengan sistem file EFS yang ada.

- Perbaiki masalah yang menyebabkan `cf-n-hup` daemon gagal saat dimulai ulang.
- Pertimbangkan node dinamis dengan `INVALID_REQUEST` flag sebagai kegagalan bootstrap untuk mode Slurm terlindungi. Node statis yang gagal Slurm pendaftaran sudah diperlakukan sebagai kegagalan bootstrap setelah `node_replacement_timeout`

Untuk detail perubahannya, lihat `CHANGELOG` file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster Versi UI
2023.05.0 dirilis](#)

AWS ParallelCluster Versi UI
2023.05.0 dirilis.

16 Mei 2023

Penyempurnaan:

- Dimulai dengan AWS ParallelCluster versi 3.6.0, tambahkan dukungan untuk RHEL 8.
- Tambahkan pemantauan biaya cluster.
- Dimulai dengan AWS ParallelCluster versi 3.6.0, tingkatkan antrian dan hitung kuota sumber daya.

Perubahan:

- Meningkatkan antarmuka pengguna wizard pembuatan cluster.
- Meningkatkan kecepatan penerapan AWS ParallelCluster UI.
- Meningkatkan antarmuka untuk menambahkan pengguna baru.
- Antrian berada di subnet head node secara default.

Perbaiki bug:

- Beralih ke wilayah yang benar setelah pembuatan cluster selesai.
- Perbaiki tampilan indikator pemuatan di fitur “Edit cluster”.
- Perbaiki pembuatan cluster saat SnapshotId properti EBS dihapus.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster Versi UI
2023.04.0 dirilis](#)

AWS ParallelCluster Versi UI
2023.04.0 dirilis.

17 April 2023

Penyempurnaan:

- Cluster membuat desain ulang wizard.
- Desain ulang halaman log cluster.
- Tambahkan pengaturan nama khusus untuk penyimpanan bersama.
- Tambahkan beberapa pilihan penyimpanan saat menambahkan penyimpanan ke cluster.
- Tambahkan DeletionPolicy dukungan untuk Amazon EFS dan FSx for Lustre.
- Tambahkan ImdsSupport pengaturan dalam konfigurasi cluster.
- Tambahkan dukungan untuk jenis instance C7.
- Ditambahkan tutorial [Mengembalikan ke versi dokumen Manajer AWS Sistem sebelumnya](#).

Perubahan:

- Konfigurasi cluster YAMB berukuran hingga 1MB.

- Pengguna tidak keluar karena otorisasi dengan kredensial sementara Boto3 IAM.
- Opsi multi-threading dinonaktifkan saat instance HPC dipilih.
- Dihapus menonaktifkan rollback pada halaman membuat cluster.
- Pengguna dicegah menggunakan AWS ParallelCluster UI sampai informasi yang diperlukan disediakan.
- Hingga 10 antrian dapat ditambahkan.
- SSM-SessionManager RunShell Dokumen tidak ditimpa selama instalasi AWS ParallelCluster UI.

Perbaikan bug:

- Perbaiki tautan kata sandi reset yang rusak.
- Perbaiki rusak delete stack yang disebabkan oleh EcrPrivateRepository tidak kosong
- Memperbaiki masalah inisialisasi kotak centang Hasilkan Kunci SSH di

bagian Beberapa properti manajemen pengguna.

- Kerusakan tetap disebabkan oleh pekerjaan dengan properti yang tidak ditentukan.
- Pengaturan SCRATCH FSx tetap.
- Tombol instance Start dan Stop tetap, masih diaktifkan setelah diklik sekali.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster-ui](#) paket di GitHub.

[AWS ParallelCluster versi
3.5.1 dirilis](#)

AWS ParallelCluster versi
3.5.1 dirilis.

29 Maret 2023

Penyempurnaan:

- [Tambahkan penguinstal
Ipcluster CLI yang
berdiri sendiri yang dapat
dieksekusi.](#)

Perubahan:

- Tingkatkan penginstal EFA ke. 1.22.0
 - EFA-driver: efa-2.1.1g (dari efa-2.1.1-1)
 - EFA-config: efa-config-1.13-1 (dari efa-config-1.12-1)
 - Profil EFA: efa-profile-1.5-1 (tidak ada perubahan)
 - libFabric-aws: libfabric-aws-1.17.0-1 (dari libfabric-aws-1.16.1amzn3.0-1)
 - RDMA-core: rdma-core-43.0-1 (tidak ada perubahan)
 - Buka MPI: openmpi40-aws-4.1.5-1 (dari openmpi40-aws-4.1.4-3)

Tingkatkan NICE DCV ke versi. 2022.2-14521

- server: 2022.2.14521-1
- xdcv: 2022.2.519-1
- gl: 2022.2.1012-1
- penampil web: 2022.2.14521-1

Perbaiki bug:

- Perbaiki potensi kegagalan peluncuran node yang disebabkan oleh pencocokan pola antara MountDir dan /etc/exports saat menghapus volume Amazon EBS bersama sebagai bagian dari pembaruan klaster.
- Perbaiki untuk mencegah pemotongan file compute_console_output log di setiap clustermgtd iterasi.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi 3.5.0 dirilis](#)

AWS ParallelCluster versi 3.5.0 dirilis.

Februari 20, 2023

Penyempurnaan:

- Akses dan kelola cluster dengan [AWS ParallelCluster UI](#).
- Tambahkan AWS ParallelCluster kebijakan berversi dalam CloudFormation templat yang dapat Anda referensikan di beban kerja Anda.
- Tambahkan pustaka AWS ParallelCluster Python yang dapat Anda gunakan dengan kode Anda sendiri.
- Tambahkan logging output konsol node komputasi ke Amazon CloudWatch pada kegagalan bootstrap node komputasi.
- Tambahkan bidang kegagalan yang berisi kode kegagalan dan alasan untuk `describe-cluster` keluaran saat pembuatan cluster gagal.
- Tambahkan validator untuk mencegah injeksi string berbahaya saat memanggil modul subproses.

- Gagal pembuatan klaster jika status klaster berubah menjadi PROTECTED saat menyediakan node statis.

Perubahan:

- Tingkatkan ke Slurm versi 22.05.8 (dari versi 22.05.7)
- Tingkatkan penginstal EFA ke 1.21.0
 - EFA-driver: efa-2.1.1-1 (dari efa-2.1)
 - EFA-config: efa-config-1.12-1 (dari efa-config-1.11-1)
 - Profil EFA: efa-profile-1.5-1 (tidak ada perubahan)
 - libFabric-aws: libfabric-aws-1.16.1amzn3.0-1 (dari libfabric-aws-1.16.1)
 - RDMA-core: rdma-core-43.0-1 (dari rdma-core-43.0-2)
 - Buka MPI: openmpi40-aws-4.1.4-3 (tidak ada perubahan)
- Buat log Slurm pengontrol lebih bertele-tele dan aktifkan logging tambahan

untuk plugin hemat Slurm daya.

Perbaiki bug:

- Perbaiki pembuatan database cluster dengan memverifikasi bahwa nama cluster tidak lebih dari 40 karakter saat Slurm akuntansi diaktifkan.
- Perbaiki masalah `clustermgtd` yang menyebabkan node komputasi, di-boot ulang Slurm, diganti jika pemeriksaan status instans EC2 gagal.
- Memperbaiki masalah yang mencegah node komputasi, dengan reservasi kapasitas yang dibagikan oleh akun lain, agar tidak diluncurkan karena kebijakan IAM yang salah pada node kepala.

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), [aws-parallelcluster-node](#), dan paketnya. [aws-parallelcluster-ui](#) GitHub

[AWS ParallelCluster versi
3.4.1 dirilis](#)

AWS ParallelCluster versi
3.4.1 dirilis.

13 Januari 2023

Perbaiki bug:

- Perbaiki masalah Slurm penjadwal yang dapat menyebabkan aplikasi pembaruan yang salah ke registri internal node komputasi. Akibatnya jika masalah ini, instans EC2 bisa menjadi tidak tersedia atau dapat didukung oleh jenis instans yang salah.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi 3.4.0 dirilis](#)

AWS ParallelCluster versi 3.4.0 dirilis.

22 Desember 2022

Penyempurnaan:

- Tambahkan dukungan untuk meluncurkan node di beberapa zona ketersediaan untuk meningkatkan ketersediaan kapasitas.
- Tambahkan dukungan untuk menentukan beberapa subnet untuk setiap antrian untuk meningkatkan ketersediaan kapasitas.
- Tambahkan parameter konfigurasi baru di [Iam/ResourceP](#) [refix](#) untuk menentukan awalan untuk jalur dan nama sumber daya IAM yang dibuat oleh AWS ParallelCluster
- Tambahkan bagian konfigurasi [DeploymentSettings](#) baru/ [LambdaFunctionsVpc](#) [Config](#) untuk menentukan konfigurasi Vpc yang digunakan oleh fungsi Lambda AWS ParallelCluster .
- Tambahkan kemampuan untuk menentukan skrip

khusus untuk dijalankan di node kepala selama pembaruan cluster. Skrip dapat ditentukan dengan [HeadNode/CustomActions /OnNodeUpdated](#) saat menggunakan Slurm sebagai penjadwal.

Perubahan:

- Hapus pembuatan target mount Amazon EFS untuk sistem file yang ada.
- Pasang sistem file EFS menggunakan `amazon-efs-utils`. Sistem file EFS dapat dipasang menggunakan enkripsi dalam transit dan pengguna resmi IAM.
- Instal stunnel 5.67 di CentOS7 dan Ubuntu untuk mendukung enkripsi in-transit EFS.
- Tingkatkan penginstal EFA ke 1.20.0 (dari 1.18.0).
 - EFA-driver: `efa-2.1` (dari `efa-1.16.0-1`)
 - EFA-config: `efa-config-1.11-1` (tidak ada perubahan)
 - Profil EFA: `efa-profile-1.5-1` (tidak ada perubahan)

- libFabric-aws:
libfabric-aws-1.16.1 (dari libfabric-aws-1.16.0~amzn4.0-1)
- RDMA-core: rdma-core-43.0-2 dari () rdma-core-41.0-2
- Buka MPI: openmpi40-aws-4.1.4-3 dari () openmpi40-aws-4.1.4-2
- Tingkatkan Slurm ke versi 22.05.7 (dari 22.05.5).
- Upgrade Python ke 3.9.16 dan 3.7.16 (dari 3.9.15 dan 3.7.13).
- Dengan Slurm 22.05.7, node dinamis dalam IDLE +CLOUD+COMPLETING +POWER_DOWN+NOT_RESPONDING status tidak dianggap tidak sehat.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi
3.3.1 dirilis](#)

AWS ParallelCluster versi
3.3.1 dirilis.

Desember 2, 2022

Perubahan:

- AMI AWS ParallelCluster produk resmi sekarang tersedia setelah penghenti an Amazon EC2 pada dua tahun.
- Tingkatkan ukuran memori AWS ParallelCluster API Lambda menjadi 2048 untuk mengurangi penalti start dingin dan menghindari batas waktu.

Perbaikan bug:

- Mencegah penggantian sistem file FSx for Lustre yang dikelola dan hilangnya data pada pembaruan klaster yang mencakup perubahan pada ID subnet armada komputasi.
- [SharedStorage DeletionPolicy](#) berlaku untuk tindakan pembaruan cluster.

Untuk detail perubahannya, lihat CHANGELOG file untuk

paket [aws-parallelcluster](#) di
GitHub

[AWS ParallelCluster
dokumentasi hanya catatan
hpc6id](#)

AWS ParallelCluster
pembaruan khusus dokumenta
si

Desember 2, 2022

- AWS ParallelCluster tidak mendukung jenis instance hpc6id untuk pengaturan/. [HeadNodeInstanceType](#)

[AWS ParallelCluster versi
3.1.5 dirilis](#)

AWS ParallelCluster versi
3.1.5 dirilis.

16 November 2022

Penyempurnaan:

- Perbaiki masalah Slurm yang mencegah penghentian node idle.
- Tingkatkan penginstal EFA ke 1.18.0
 - Pengemudi EFA: `efa-1.16.0-1`
 - EFA-config: `efa-config-1.11-1` (dari `efa-config-1.9-1`)
 - Profil EFA: `efa-profile-1.5-1` (tidak ada perubahan)
 - Libfabric-aws: `libfabric-aws-1.16.0~amzn4.0-1` (dari `libfabric-1.13.2`)
 - RDMA-core: `rdma-core-41.0-2` (dari `rdma-core-37.0`)
 - Buka MPI: `openmpi40-aws-4.1.4-2` (dari `openmpi40-aws-4.1.1-2`)

Perubahan:

- Tambahkan `lambda:ListTags` dan `lambda:Un`

`tagResource` ke tumpukan AWS ParallelCluster API yang `ParallelClusterUserRole` digunakan untuk pembaruan klaster.

- Tingkatkan Perpustakaan Intel MPI ke Pembaruan Versi 2021.6 (dari Pembaruan Versi 2021.4). Untuk informasi selengkapnya, lihat [Intel® MPI Library 2021 Update 6](#).
- Tingkatkan driver NVIDIA ke versi 470.141.03 (dari 470.103.01).
- Tingkatkan NVIDIA Fabric Manager ke versi 470.141.03 (dari 470.103.01).

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi
3.3.0 dirilis](#)

AWS ParallelCluster versi
3.3.0 dirilis.

2 November 2022

Penyempurnaan:

- Tambahkan dukungan untuk konfigurasi alokasi beberapa instans untuk sumber daya komputasi saat digunakan Slurm sebagai penjadwal. Untuk informasi selengkapnya, lihat [Beberapa alokasi tipe instans dengan Slurm](#).
- Tambahkan dukungan untuk menambah dan menghapus [SharedStorage](#) dengan pembaruan cluster, menggunakan konfigurasi yang diperbarui. Untuk informasi selengkapnya, lihat [Penyimpanan bersama](#).
- Tambahkan parameter konfigurasi baru `DeletionPolicy` untuk [Efs](#) dan pengaturan penyimpanan [FsxLustre](#) bersama untuk mendukung retensi penyimpanan.
- Tambahkan dukungan untuk Slurm akuntansi dengan parameter konfigurasi baru [Scheduling /SlurmSettings /Database](#). Untuk

informasi selengkapnya, lihat [Slurmakuntansi dengan AWS ParallelCluster](#).

- Tambahkan dukungan untuk reservasi kapasitas sesuai permintaan dan grup sumber daya reservasi kapasitas. Untuk informasi selengkapnya, lihat [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)](#).
- Tambahkan parameter konfigurasi baru untuk menentukan versi IMDS yang akan didukung di cluster atau membangun infrastruktur gambar di cluster, [Imds/ImdsSupport](#), dan build, [Imds/ImdsSupport](#), konfigurasi.
- Tambahkan dukungan untuk [Networking /Placement Group](#) di [ComputeResources](#) bagian [SlurmQueues](#).
- Tambahkan dukungan untuk instance dengan beberapa antarmuka jaringan yang dibatasi hanya satu ENI per perangkat.
- Tingkatkan validasi jaringan untuk sistem file Amazon

EFS eksternal dengan memeriksa blok CIDR di grup keamanan terlampir.

- Tambahkan validator untuk memeriksa apakah jenis instance yang dikonfigurasi mendukung grup penempatan.
- Konfigurasi NFS menjadi min (256, maks (8, num_core* 4)) untuk memastikan stabilitas dan kinerja yang lebih baik.
- Pindahkan instalasi NFS pada waktu pembuatan untuk mengurangi waktu konfigurasi.
- Aktifkan enkripsi sisi server untuk topik EcrlmageB uilder SNS yang dibuat saat menerapkan AWS ParallelCluster API dan digunakan untuk memberi tahu tentang peristiwa docker image build.

Perubahan:

- Ubah perilaku [SlurmQueues /Networking /PlacementGroup /Enabled](#).
Sekarang membuat grup penempatan terkelola yang unik untuk setiap sumber daya komputasi, bukan

grup penempatan terkelola tunggal untuk semua sumber daya komputasi.

- Tambahkan dukungan untuk [SlurmQueues /Networking /Placement Group /Name](#) sebagai metode penamaan yang disukai.
- Pindahkan tag node kepala dari Launch Template ke definisi instance untuk menghindari penggantian node kepala pada pembaruan tag.
- Nonaktifkan multithreading melalui skrip yang dijalankan oleh `cloud-init` dan bukan melalui `CpuOptions` set di Template Peluncuran.
- Tingkatkan Python ke versi 3.9 dan NodeJS ke versi 16 di infrastruktur API, wadah API Docker, dan sumber daya Lambda cluster.
- Hapus dukungan untuk Python 3.6 in. `aws-parallelcluster-batch-c`
`li`
- Tingkatkan Slurm ke versi 22.05.5 (dari 21.08.8-2).

- Tingkatkan driver NVIDIA ke versi 470.141.03 (dari 470.129.06).
- Tingkatkan NVIDIA Fabric Manager ke versi 470.141.03 (dari 470.129.06).
- Tingkatkan NVIDIA CUDA Toolkit ke versi 11.7.1 ().
from 11.4.4
- Upgrade Python yang digunakan dalam AWS ParallelCluster virtualenvs dari ke. 3.7.13 3.9.15
- Tingkatkan penginstal EFA ke versi 1.18.0.
 - EFA-driver: efa-1.16.0-1 (tidak ada perubahan)
 - Konfigurasi EFA: () efa-config-1.11-1 from efa-config-1.10-1
 - Profil EFA: efa-profile-1.5-1 (tidak ada perubahan)
 - Libfabric-aws: libfabric-aws-1.16.0~amzn4.0-1 (dari). libfabric-aws-1.16.0~amzn2.0-1
 - RDMA-core: rdma-core-41.0-2 (dari) rdma-core-37.0

- Buka MPI: `openmpi40-aws-4.1.4-2` (dari `openmpi40-aws-4.1.1-2`)
- Tingkatkan NICE DCV ke versi `2022.1-13300` (dari `2022.0-12760`).
- Aktifkan penindasan `SingleSubnetValidator forQueues`.
- Jangan mengganti DRAIN node saat node dalam `COMPLETING` keadaan karena Epilog mungkin masih berjalan.

Perbaiki bug:

- Perbaiki validasi parameter filter dalam AWS ParallelCluster `ListClusterLogStreams` perintah untuk gagal ketika filter yang salah dilewatkan.
- Perbaiki validasi [SharedStorage EfsSettings](#) parameter/gagal validasi ketika `FileSystemId` ditentukan bersama dengan parameter [SharedStorage /EfsSettings](#) lainnya. Sebelumnya, `FileSystemId` tidak termasuk.
- Perbaiki pembaruan cluster saat mengubah urutan

[SharedStorage](#) bersama dengan perubahan lain dalam konfigurasi.

- Perbaiki `UpdateParallelClusterLambdaRole` di AWS ParallelCluster API untuk mengunggah log ke CloudWatch.
- Perbaiki Cinc tidak menggunakan bundel sertifikat CA lokal saat menginstal paket sebelum buku masak apa pun dijalankan.
- Perbaiki hang dalam memutakhirkan ubuntu dengan `pcluster build-image` kapan `Build:UpdateOsPackages:Enabled:true` diatur.
- Perbaiki parsing konfigurasi cluster YAMM dengan gagal pada kunci duplikat.

Untuk detail perubahannya, lihat `CHANGELOG` file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster dokumentasi hanya referensi API yang ditambahkan.](#)

AWS ParallelCluster pembaruan khusus dokumentasi

27 Oktober 2022

- Ditambahkan versi 3 [AWS ParallelClusterReferensi API](#) ke dokumentasi.

[AWS ParallelCluster versi
3.2.1 dirilis](#)

AWS ParallelCluster versi
3.2.1 dirilis.

3 Oktober 2022

Penyempurnaan:

- Tingkatkan logika untuk mengaitkan tabel perutean host ke kartu jaringan yang berbeda untuk mendukung instans EC2 dengan lebih baik dengan beberapa NIC.

Perubahan:

- Tingkatkan driver NVIDIA ke versi 470.141.03.
- Tingkatkan NVIDIA Fabric Manager ke versi 470.141.03.
- Nonaktifkan tugas `cron` pekerjaan `man-db` dan `mlocate`, yang mungkin berdampak negatif pada kinerja node.
- Tingkatkan Perpustakaan Intel MPI ke 2021.6.0.602.
- Upgrade Python dari 3.7.10 ke 3.7.13 sebagai tanggapan atas risiko keamanan ini.

Perbaiki bug:

- Hindari kegagalan `DescribeCluster` saat konfigurasi cluster tidak tersedia.

Untuk detail perubahannya, lihat `CHANGELOG` file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi 3.2.0 dirilis](#)

AWS ParallelCluster versi 3.2.0 dirilis.

27 Juli 2022

Penyempurnaan:

- Tambahkan dukungan untuk [penjadwalan berbasis memori](#) di Slurm.
 - Konfigurasi memori nyata node komputasi dalam konfigurasi cluster Slurm.
- Tambahkan parameter konfigurasi baru [Scheduling /SlurmSettings /EnableMemoryBasedScheduling](#) untuk mengaktifkan penjadwalan berbasis memori di Slurm.
- Tambahkan parameter konfigurasi baru [Scheduling /SlurmQueues /ComputeResources /SchedulableMemory](#) untuk mengganti nilai default memori yang dilihat oleh penjadwal pada node komputasi.

- Tingkatkan fleksibilitas pada pembaruan konfigurasi cluster untuk menghindari penghentian dan dimulainya seluruh cluster bila memungkinkan. Tambahkan parameter konfigurasi baru [Scheduling /SlurmSettings /QueueUpdateStrategy](#) untuk mengatur strategi yang disukai untuk digunakan saat node komputasi memerlukan pembaruan dan penggantian konfigurasi.
- Tingkatkan mekanisme fail-over atas sumber daya komputasi yang tersedia saat mengalami masalah kapasitas yang tidak mencukupi dengan instans EC2. [Nonaktifkan node komputasi dengan jumlah waktu yang dapat dikonfigurasi](#) ketika peluncuran node gagal karena kapasitas yang tidak mencukupi.
- Tambahkan dukungan untuk memasang sistem [FSx for OpenZFS](#) file [FSx for ONTAP](#) dan yang ada.
- Tambahkan dukungan untuk memasang beberapa instance [Sistem File Elastis Amazon yang ada, FSx for](#)

[Lustre, FSx untuk ONTAP,](#)
[dan FSx untuk](#) sistem file
OpenZFS.

- Tambahkan dukungan untuk [FSx for Lustre Persisten t_2](#) jenis penyebaran saat membuat sistem file baru.
- Minta pengguna untuk mengaktifkan EFA untuk jenis instans yang didukung saat menggunakan [pcluster configure](#) wizard.
- Tambahkan dukungan untuk me-reboot node komputasi menggunakan Slurm.
- Tingkatkan penanganan status daya Slurm untuk juga memperhitungkan mematikan node secara manual.
- Instal NVIDIA GDRCopy 2.3 ke dalam AMI produk untuk mengaktifkan salinan memori GPU latensi rendah.

Perubahan:

- Tingkatkan penginstal EFA ke versi 1.17.2.
 - Pengemudi EFA: efa-1.16.0-1
 - Konfigurasi EFA: efa-config-1.10-1

- Profil EFA: efa-profile-1.5-1
- Libfabric: libfabric-aws-1.16.0~amzn2.0-1
- Inti RDMA: rdma-core-41.0-2
- Buka MPI: openmpi40-aws-4.1.4-2
- Tingkatkan NICE DCV ke versi 2022.0-12760.
- Tingkatkan driver NVIDIA ke versi 470.129.06.
- Tingkatkan NVIDIA Fabric Manager ke versi 470.129.06.
- Ubah tipe volume EBS default dari gp2 ke gp3 baik di root maupun volume tambahan.
- Perubahan pada sistem file FSx for Lustre yang dibuat oleh: AWS ParallelCluster
 - Ubah jenis penerapan default menjadi Scratch_2 .
 - Ubah versi server Lustre menjadi. 2.12
- Tidak memerlukan [Placement Group /Enabled](#) untuk disetel ke true saat melewati Placement Group /yang adaId.

- Tidak mengizinkan pengaturan `PlacementGroup /Id` ketika `PlacementGroup /secara eksplisit Enabled` disetel ke `false`
- Tambahkan `parallelcluster:cluster-name` tag ke semua sumber daya yang dibuat oleh AWS ParallelCluster.
- Tambahkan `lambda:ListTags` dan `lambda:UntagResource` `ParallelClusterUserRole` gunakan tumpukan AWS ParallelCluster API untuk pembaruan klaster.
- Batasi akses IPv6 IMDS ke pengguna admin `root` dan cluster saja, ketika parameter konfigurasi `HeadNode/Imds/SecureDDI` diaktifkan.
- Dengan AMI khusus, gunakan ukuran volume root AMI alih-alih ParallelCluster default 35 GiB. Nilai dapat diubah dalam file konfigurasi cluster.
- Penonaktifan otomatis armada komputasi ketika parameter konfigurasi `Scheduling //SlurmQueues ComputeRes`

`sources /SpotPrice`

lebih rendah dari harga

pemenuhan permintaan

Spot minimum yang

diperlukan.

- Tampilkan `requested_value` dan `current_value` nilai dalam set perubahan saat menambahkan atau menghapus bagian selama pembaruan.
- Nonaktifkan `aws-ubuntu-eni-helper` layanan, tersedia di Deep Learning AMI, untuk menghindari konflik `configure_nw_interface.sh` saat mengonfigurasi instance dengan beberapa kartu jaringan.
- Hapus dukungan untuk Python 3.6.
- Setel MTU ke 9001 untuk semua antarmuka jaringan saat mengonfigurasi instance dengan beberapa kartu jaringan.
- Hapus trailing dot saat mengonfigurasi node komputasi FQDN.
- Kelola node statis di `POWERING_DOWN` .
- Tidak menggantikan node dinamis `POWER_DOWN`

karena pekerjaan mungkin masih berjalan.

- Mulai ulang `clustermgtd` dan `slurmctld` daemon pada waktu pembaruan cluster hanya ketika `Scheduling` parameter diperbarui dalam konfigurasi cluster.
- Perbarui `slurmctld` dan `slurmd` `systemd` layanan file.
- Batasi akses IPv6 ke IMDS hanya untuk pengguna `root` dan `cluster admin`, ketika parameter konfigurasi `HeadNode//Imdsdiaktifkan.Secured`.
- Setel konfigurasi Slurm `AuthInfo=cred_expire=70` untuk mengurangi waktu pekerjaan yang diminta harus menunggu sebelum memulai lagi ketika node tidak tersedia.
- Tingkatkan dependensi buku masak pihak ketiga:
 - `apt-7.4.2` (dari `apt-7.4.0`)
 - `baris-4.5.2` (dari `baris-4.0.1`)
 - `openssh-2.10.3` (dari `openssh-2.9.1`)
 - `pyenv-3.5.1` (dari `pyenv-3.4.2`)

- selinux-6.0.4 (dari selinux-3.1.1)
- yum-7.4.0 (dari yum-6.1.1)
- yum-epel-4.5.0 (dari yum-epel-4.1.2)

Perbaiki bug:

- Perbaiki perilaku default untuk melewati langkah AWS ParallelCluster validasi dan pengujian saat membuat AMI kustom.
- Perbaiki kebocoran pegangan file. `computemgtd`
- Perbaiki kondisi balapan yang secara sporadis menyebabkan instans yang diluncurkan segera dihentikan karena belum tersedia dalam respons `EC2.DescribeInstances`
- Perbaiki dukungan untuk `DisableSimultaneousMultithreading` parameter pada tipe instans dengan prosesor Arm.
- Perbaiki kegagalan pembaruan tumpukan AWS ParallelCluster API saat memutakhirkan dari versi sebelumnya. Tambahkan pola sumber

daya yang digunakan untuk ListImage PipelineImages Action diEcrImageDeletionLambdaRole .

- Perbaiki AWS ParallelCluster API yang menambahkan izin yang hilang yang diperlukan untuk mengimpor atau mengekspor dari Amazon S3 saat membuat sistem file FSx for Lustre.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster pembaruan khusus dokumentasi tahun ini hingga saat ini](#)

AWS ParallelCluster pembaruan khusus dokumentasi.

6 Juli 2022

Bagian baru:

- [Praktik terbaik: peringatan anggaranV3](#)
- [Praktik terbaik: memindahkan kluster ke versiAWS ParallelCluster minor atau patch baruV3](#)
- [Bekerja dengan Amazon S3V3](#)
- [Mengerjakan Instans ITSpotV3](#)
- [Slurmmodus cluster dilindungiV3](#)
- [AWS ParallelClustersumber daya dan penandaanV3](#)
- [CloudWatchDasbor AmazonV3](#)
- [Integrasi dengan Amazon CloudWatch LogsV3](#)
- [Elastic Fabric AdapterV3](#)
- [AWS ParallelCluster Kustomisasi AMIV3](#)
- [Meluncurkan instans dengan OCR \(Cadangan Kapasitas Sesuai Permintaan\)V3](#)
- [Penambalan AMI dan penggantian instans EC2V3](#)

- [Bagaimana AWS ParallelCluster berhasil](#)
- [Mengonfigurasi enkripsi penyimpanan bersama dengan kunci AWS KMS](#)
- [Menjalankan pekerjaan dalam kluster mode antrian ganda](#)
- [Menggunakan API AWS ParallelCluster ini](#)

Pembaruan bagian:

- [Praktik terbaik: kinerja jaringan](#): Menambahkan praktik terbaik untuk menggunakan Adaptor Kain Elastis.
- [AWS Identity and Access Management izin di AWS ParallelCluster](#): Berbagai pembaruan dan ditambahkan [Kebijakan AWS ParallelCluster pengguna](#) saat menggunakan [Amazon FSx for Lustre](#).
- [Memecahkan Masalah AWS ParallelCluster](#): Berbagai pembaruan.

[AWS ParallelCluster versi
3.1.4 dirilis](#)

AWS ParallelCluster versi
3.1.4 dirilis.

Mei 16, 2022

Penyempurnaan:

- Tambahkan validasi untuk [Directory Service](#) / [PasswordSecretArn](#) gagal jika rahasianya tidak ada.

Tambahkan dukungan untuk mengaktifkan Slurm otentikasi JWT.

Perubahan:

- Tingkatkan Slurm ke versi 21.08.8-2.
- Bangun Slurm dengan dukungan JWT.
- Tidak memerlukan [Placement Group](#) / [Enabled](#) untuk disetel ke true saat melewati [Placement Group](#) / yang ada [Id](#).
- Tambahkan `lambda:TagResource` untuk `ParallelClusterUserRole` digunakan oleh tumpukan ParallelCluster API untuk pembuatan cluster dan pembuatan gambar.

Perbaiki bug:

- Perbaiki kemampuan untuk mengekspor log cluster saat menggunakan `export-cluster-logs` perintah dengan `--filters` opsi.
- Perbaiki titik masuk AWS Batch Docker untuk menggunakan direktori `/home` bersama untuk mengoordinasikan eksekusi pekerjaan Multi-node-paralel .
- Setel ulang alamat node saat mengatur slurm node statis yang tidak sehat ke bawah untuk menghindari perawatan node statis gagal dengan kapasitas yang tidak mencukupi sebagai node kegagalan bootstrap.

Untuk detail perubahannya, lihat `CHANGELOG` file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi
3.1.3 dirilis](#)

AWS ParallelCluster versi
3.1.3 dirilis.

20 April 2022

Penyempurnaan:

- Jalankan pembuatan kunci SSH bersama dengan pembuatan direktori HOME, misalnya, selama login SSH, saat beralih ke pengguna lain dan saat menjalankan perintah sebagai pengguna lain.
- Tambahkan dukungan untuk FQDN dan LDAP Distinguished Names dalam parameter konfigurasi/. [Directory Service DomainName](#) Validator baru sekarang memeriksa kedua sintaks.
- `update_directory_service_password.sh`
Skrip baru yang digunakan pada node kepala mendukung pembaruan manual kata sandi Direktori Aktif dalam konfigurasi SSSD. Kata sandi diambil oleh AWS Secrets Manager as dari konfigurasi cluster.
- Tambahkan dukungan untuk menerapkan infrastruktur API di lingkungan tanpa VPC default.

Perubahan:

- Nonaktifkan C-States yang lebih dalam di AMI resmi x86_64 dan AMI yang dibuat melalui `build-image` perintah, untuk menjamin kinerja tinggi dan latensi rendah.
- Pembaruan paket OS dan perbaikan keamanan.
- Ubah gambar dasar Amazon Linux 2 untuk menggunakan AMI dengan Kernel 5.10.

Perbaikan bug:

- Perbaiki tumpukan `build-image DELETE_FAILED` setelah image dibangun berhasil, karena kebijakan EC2 Image Builder yang baru.
- Perbaiki parameter [Directory Service DomainAdd](#) konfigurasi/konversi ke properti `ldap_uri` SSSD ketika berisi beberapa alamat domain.

Untuk detail perubahan, lihat `CHANGELOG` file untuk [aws-parallelcluster](#), dan paket aktif.

[aws-parallelcluster-cookbook](#)

GitHub

[AWS ParallelCluster versi 3.1.2 dirilis](#)

AWS ParallelCluster versi 3.1.2 dirilis.

2 Maret 2022

Perubahan:

- Tingkatkan Slurm ke versi 21.08.6 (dari 21.08.5).

Perbaiki bug:

- Perbaiki pembaruan `/etc/hosts` file pada node komputasi saat cluster digunakan di subnet tanpa akses internet.
- Perbaiki node komputasi bootstrap untuk menunggu inisialisasi drive sementara sebelum bergabung dengan cluster.

Untuk detail perubahannya, lihat `CHANGELOG` file untuk paket [aws-parallelcluster](#) aktif.
GitHub

[AWS ParallelCluster versi
3.1.1 dirilis](#)

AWS ParallelCluster versi
3.1.1 dirilis.

Februari 10, 2022

- Tambahkan dukungan untuk beberapa lingkungan cluster pengguna dengan [mengintegrasikan dengan domain Active Directory \(AD\)](#) yang dikelola. AWS Directory Service
- Tambahkan dukungan untuk [UseEc2Hostnames](#) dalam file konfigurasi cluster. Ketika disetel ke true, gunakan nama host default EC2 (misalnya ip-1-2-3-4) untuk node komputasi.
- Tambahkan dukungan untuk pembuatan cluster di [subnet tanpa akses internet](#).
- Tambahkan dukungan untuk beberapa jenis instans komputasi per antrian.
- Tambahkan dukungan untuk penjadwalan GPU dengan Slurm pada instans ARM dengan kartu NVIDIA.
- Tambahkan flag disingkat untuk `cluster-name` (-n), `region` (-r), `image-id` (-i) dan `cluster-configuration /image-configuration` (-c) ke CLI AWS ParallelCluster .

- Tambahkan dukungan untuk NEW_CHANGED_DELETE opsi untuk parameter FSx for Lustre [AutoImportPolicy](#) .
- Tambahkan parallelcluster:compute-resource-name tag ke LaunchTemplates sumber daya EC2 yang digunakan oleh node komputasi.
- Tingkatkan grup keamanan yang dibuat di dalam klaster untuk memungkinkan koneksi masuk dari grup keamanan khusus saat SecurityGroups parameter ditentukan untuk beberapa node kepala dan/atau antrian.
- Instal driver NVIDIA dan perpustakaan CUDA untuk ARM.

Perubahan:

- Tingkatkan Slurm ke versi 21.08.5 (dari20.11.8).
- Tingkatkan plugin Slurm ke versi 21.08 (dari20.11).
- Tingkatkan NICE DCV ke versi 2021.3-11591 (dari2021.1-10851).

- Tingkatkan driver NVIDIA ke versi 470.103.01 (dari 470.57.02).
- Tingkatkan manajer NVIDIA Fabric ke versi 470.103.01 (dari 470.57.02).
- Tingkatkan CUDA ke versi 11.4.4 (dari 11.4.0).
- [Intel MPI](#) diperbarui ke Pembaruan Versi 2021.4 (diperbarui dari Pembaruan Versi 2019.8). Untuk informasi selengkapnya, lihat [Intel® MPI Library 2021 Update 4](#).
- Tingkatkan pMix ke versi 3.2.3 (dari 3.1.5).
- Hapus dumping node komputasi yang gagal ke `/home/logs/compute`. File log node komputasi tersedia di dalam CloudWatch dan di log konsol EC2.
- Aktifkan potensi untuk menekan `SlurmQueues` dan `ComputeResources` memperpanjang validator.
- Nonaktifkan pembaruan paket pada waktu peluncuran instans di Amazon Linux 2.
- Nonaktifkan metadata gambar ImageBuilder yang disempurnakan EC2 saat

membuat AWS ParallelCluster gambar khusus.

- Secara eksplisit mengatur sumber `cloud-init` data menjadi EC2. Ini menghemat waktu boot untuk platform Ubuntu dan CentOS.
- Gunakan nama sumber daya komputasi daripada jenis instance dalam nama template peluncuran armada komputasi.
- Arahkan `stderr` dan `stdout` ke file log CLI untuk mencegah teks yang tidak diinginkan dalam output CLI `pcluster`.
- Pindahkan resep konfigurasi/instal ke buku masak terpisah yang dipanggil dari yang utama. Entrypoint yang ada dipertahankan dan kompatibel ke belakang.
- Unduh dependensi platform Intel HPC selama waktu pembuatan AMI untuk menghindari kontak internet selama waktu pembuatan cluster.
- Jangan hapus - dari nama sumber daya komputasi saat mengonfigurasi node Slurm.

- Jangan mengkonfigurasi GPU di Slurm ketika driver NVIDIA tidak diinstal.
- Perbaiki `ecs:ListContainerInstances` izin di `BatchUserRole` .
- Perbaiki ekspor log cluster ketika tidak ada awalan yang ditentukan, sebelumnya diekspor ke awalan. `None`
- Perbaiki rollback yang tidak dilakukan jika terjadi kegagalan pembaruan cluster.
- Perbaiki `ecs:ListContainerInstances` izin di `BatchUserRole` .
- Perbaiki `RootVolume` skema untuk `HeadNode` dengan memunculkan kesalahan jika tidak didukung `KmsKeyId` ditentukan.
- Perbaiki metrik Amazon FSx yang hilang untuk ditampilkan di Dasbor. `CloudWatch`
- Perbaiki `EfaSecurityGroupValidator` .
Sebelumnya, itu berpotensi menghasilkan kegagalan palsu ketika grup keamanan khusus disediakan dan EFA diaktifkan.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#), dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi 3.0.3 dirilis](#)

AWS ParallelCluster versi 3.0.3 dirilis.

Januari 17, 2022

- Nonaktifkan `log4j-cve-2021-44228-hotpatch` agent (`Log4jHotPatch`) di Amazon Linux 2 untuk menghindari potensi penurunan kinerja. Untuk informasi selengkapnya, lihat [Pengumuman Hotpatch Amazon Linux untuk Apache Log4j](#).

Untuk detail perubahan, lihat CHANGELOG file untuk [aws-parallelcluster](#) dan paket di [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster versi 3.0.2 dirilis](#)

AWS ParallelCluster versi 3.0.2 dirilis.

November 5, 2021

Tingkatkan [Elastic Fabric Adapter](#) penginstal ke 1.14.1

- Konfigurasi EFA: efa-config-1.9-1 (dari efa-config-1.9)
- Profil EFA: efa-profile-1.5-1 (dari efa-profile-1.5)
- Modul Kernel EFA: efa-1.14.2 (dari efa-1.13.0)
- Inti RDMA: rdma-core-37.0 (dari rdma-core-35)
- Libfabric: libfabric-1.13.2 (dari libfabric-1.13.0)
- Buka MPI: openmpi40-aws-4.1.1-2 (tidak ada perubahan)

GPUDirect RDMA selalu diaktifkan jika didukung oleh jenis instans. Opsi [GdrSupport](#) konfigurasi tidak berpengaruh.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#) dan paket

aktif. [aws-parallelcluster-node](#)

GitHub

[AWS ParallelCluster versi 3.0.1 dirilis](#)

AWS ParallelCluster versi 3.0.1 dirilis.

27 Oktober 2021

Alat migrasi konfigurasi cluster

- Pelanggan sekarang dapat memigrasikan konfigurasi kluster mereka dari format AWS ParallelCluster versi 2 ke format versi 3 berbasis YAML AWS ParallelCluster . Untuk informasi selengkapnya, lihat [pcluster3-config-converter](#) .

Head node dapat dihentikan

- [Setelah menghentikan armada komputasi, node kepala dapat dihentikan dan kemudian dimulai ulang menggunakan konsol Amazon EC2 atau perintah `stop-instance`.](#) AWS CLI

Wilayah AWS Membaca default dari ~/.aws/config file

- Untuk [pcluster](#) perintah, jika tidak Wilayah AWS ditentukan dalam file konfigurasi, di lingkungan, atau pada baris perintah, default yang Wilayah AWS

ditentukan dalam `region` pengaturan di `[default]` bagian `~/.aws/config` file digunakan.

Untuk detail perubahannya, lihat `CHANGELOG` file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#) dan paket aktif. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster versi
3.0.0 dirilis](#)

AWS ParallelCluster versi
3.0.0 dirilis.

10 September 2021

Support untuk manajemen
klaster melalui Amazon API
Gateway

- Pelanggan sekarang dapat mengelola dan menerapkan cluster melalui titik akhir HTTP dengan Amazon API Gateway. Ini membuka kemungkinan baru untuk alur kerja skrip atau berbasis peristiwa.

Antarmuka baris AWS ParallelCluster perintah (CLI) juga telah didesain ulang untuk kompatibilitas dengan API ini dan menyertakan opsi keluaran JSON baru. Fungsionalitas baru ini memungkinkan pelanggan untuk menerapkan kemampuan blok bangunan serupa menggunakan CLI juga.

Peningkatan kreasi AMI
kustom

- Pelanggan sekarang memiliki akses ke proses yang lebih kuat untuk

membuat dan mengelola AMI kustom menggunakan EC2 Image Builder. AMI kustom sekarang dapat dikelola melalui file AWS ParallelCluster konfigurasi terpisah, dan dapat dibuat menggunakan [pcluster build-image](#) perintah di antarmuka baris AWS ParallelCluster perintah.

Untuk detail perubahannya, lihat CHANGELOG file untuk [aws-parallelcluster](#), [aws-parallelcluster-cookbook](#) dan paket aktif. [aws-parallelcluster-node](#) GitHub

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.