AWS ParallelCluster Benutzerhandbuch (v2)

AWS ParallelCluster

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS ParallelCluster: AWS ParallelCluster Benutzerhandbuch (v2)

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS ParallelCluster ... 1
- Preisgestaltung ... 1

Einrichten AWS ParallelCluster ... 2
- Installation AWS ParallelCluster ... 2
  - Installation AWS ParallelCluster in einer virtuellen Umgebung (empfohlen) ... 2
  - Installation AWS ParallelCluster in einer nicht-virtuellen Umgebung mit Pip ... 3
  - Nach der Installation zu ergreifende Schritte ... 3
  - Detaillierte Anweisungen für jede Umgebung ... 4
  - Virtuelle Umgebung ... 4
  - Linux ... 6
  - macOS ... 11
  - Windows ... 13
- Konfiguration AWS ParallelCluster ... 16
- Bewährte Methoden ... 24
  - Bewährte Methoden: Auswahl des Master-Instance-Typs ... 24
  - Bewährte Methoden: Netzwerkleistung ... 25
  - Bewährte Methoden: Budgetwarnungen ... 25
  - Bewährte Methoden: Umstellung eines Clusters auf eine neue AWS ParallelCluster Minor- oder Patch-Version ... 26
- Wechseln Sie von CfnCluster zu AWS ParallelCluster ... 27
- Unterstützte Regionen ... 29

Benutzen AWS ParallelCluster ... 31
- Netzwerkkonfigurationen ... 31
  - AWS ParallelCluster in einem einzigen öffentlichen Subnetz ... 32
  - AWS ParallelCluster unter Verwendung von zwei Subnetzen ... 33
  - AWS ParallelCluster in einem einzigen privaten Subnetz, verbunden mit AWS Direct Connect ... 34
  - AWS ParallelCluster mit `awsbatch` Scheduler ... 35
- Benutzerdefinierte Bootstrap-Aktionen ... 37
  - Konfiguration ... 38
  - Argumente ... 38
  - Beispiel ... 38
- Arbeiten mit Amazon S3 ... 40
  - Beispiele ... 40

Arbeiten mit Spot-Instances ... 41
Szenario 1: Spot-Instance ohne ausgeführte Aufgaben wird unterbrochen ... 42
Szenario 2: Spot-Instance mit Einzelknotenaufgaben wird unterbrochen ... 42
Szenario 3: Spot-Instance, auf der Aufgaben mit mehreren Knoten ausgeführt werden, wird unterbrochen ... 43
AWS Identity and Access Management Rollen in AWS ParallelCluster ... 45
Standardeinstellungen für die Clustererstellung ... 45
Verwenden einer vorhandenen IAM-Rolle für Amazon EC2 ... 46
AWS ParallelCluster Beispiel für Instanz- und Benutzerrichtlinien ... 46
Scheduler werden unterstützt von AWS ParallelCluster ... 88
Son of Grid Engine ... 88
Slurm Workload Manager ... 89
Torque Resource Manager ... 102
AWS Batch ... 102
Tagging ... 110
CloudWatch Amazon-Dashboard ... 113
Integration mit Amazon CloudWatch Logs ... 115
Elastic Fabric Adapter ... 117
Intel Select Solutions ... 118
Intel MPI aktivieren ... 120
Intel HPC-Plattform-Spezifikation ... 121
ARM-Leistungsbibliotheken ... 122
Stellen Sie über Amazon DCV eine Connect zum Hauptknoten her ... 124
Amazon DCV HTTPS-Zertifikat ... 125
Lizenzierung von Amazon DCV ... 125
Verwenden von `pcluster update` ... 126
AMI-Patching und EC2 Instanzersatz ... 129
Aktualisierung oder Austausch der Headnode-Instanz ... 129
Einschränkungen des Instance-Speichers ... 130
Behelfslösungen für Einschränkungen beim Instanzspeicher ... 131
Stoppen und starten Sie den Hauptknoten eines Clusters ... 132
AWS ParallelCluster CLI-Befehle ... 134
`pcluster` ... 134
Argumente ... 134
Unterbefehle: ... 134
`pcluster configure` ... 135

pcluster create ... 136
pcluster createami ... 138
pcluster dcv ... 142
pcluster delete ... 144
pcluster instances ... 146
pcluster list ... 147
pcluster ssh ... 148
pcluster start ... 149
pcluster status ... 150
pcluster stop ... 151
pcluster update ... 152
pcluster version ... 155
pcluster-config ... 155
Benannte Argumente ... 155
Konfiguration ... 158
Layout ... 159
[global] Abschnitt ... 159
cluster_template ... 159
update_check ... 160
sanity_check ... 160
[aws] Abschnitt ... 160
[aliases] Abschnitt ... 161
[cluster] Abschnitt ... 162
additional_cfn_template ... 164
additional_iam_policies ... 164
base_os ... 165
cluster_resource_bucket ... 167
cluster_type ... 168
compute_instance_type ... 169
compute_root_volume_size ... 169
custom_ami ... 170
cw_log_settings ... 171
dashboard_settings ... 171
dcv_settings ... 172
desired_vcpus ... 172
disable_cluster_dns ... 173

disable_hyperthreading 173
ebs_settings 174
ec2_iam_role 175
efs_settings 175
enable_efa 176
enable_efa_gdr 176
enable_intel_hpc_platform 177
encrypted_ephemeral 178
ephemeral_dir 178
extra_json 178
fsx_settings 179
iam_lambda_role 179
initial_queue_size 180
key_name 181
maintain_initial_size 181
master_instance_type 182
master_root_volume_size 182
max_queue_size 183
max_vcpus 183
min_vcpus 184
placement 184
placement_group 185
post_install 186
post_install_args 186
pre_install 187
pre_install_args 187
proxy_server 187
queue_settings 188
raid_settings 189
s3_read_resource 189
s3_read_write_resource 189
scaling_settings 190
scheduler 190
shared_dir 191
spot_bid_percentage 192
spot_price 192

tags ... 193
template_url ... 194
vpc_settings ... 194
[compute_resource] Abschnitt ... 195
initial_count ... 195
instance_type ... 196
max_count ... 196
min_count ... 197
spot_price ... 197
[cw_log] Abschnitt ... 198
enable ... 198
retention_days ... 198
[dashboard] Abschnitt ... 199
enable ... 199
[dcv] Abschnitt ... 200
access_from ... 201
enable ... 201
port ... 202
[ebs] Abschnitt ... 202
shared_dir ... 203
ebs_kms_key_id ... 203
ebs_snapshot_id ... 204
ebs_volume_id ... 204
encrypted ... 204
volume_iops ... 205
volume_size ... 206
volume_throughput ... 207
volume_type ... 207
[efs] Abschnitt ... 208
efs_fs_id ... 209
efs_kms_key_id ... 210
encrypted ... 210
performance_mode ... 211
provisioned_throughput ... 211
shared_dir ... 212
throughput_mode ... 212

[fsx] Abschnitt 213
auto_import_policy 215
automatic_backup_retention_days 216
copy_tags_to_backups 216
daily_automatic_backup_start_time 217
data_compression_type 218
deployment_type 218
drive_cache_type 219
export_path 220
fsx_backup_id 220
fsx_fs_id 221
fsx_kms_key_id 221
import_path 222
imported_file_chunk_size 222
per_unit_storage_throughput 223
shared_dir 223
storage_capacity 224
storage_type 225
weekly_maintenance_start_time 227
[queue] Abschnitt 227
compute_resource_settings 228
compute_type 228
disable_hyperthreading 229
enable_efa 229
enable_efa_gdr 230
placement_group 230
[raid] Abschnitt 231
shared_dir 232
ebs_kms_key_id 232
encrypted 233
num_of_raid_volumes 233
raid_type 233
volume_iops 234
volume_size 235
volume_throughput 236
volume_type 236

[scaling] Abschnitt ... 237
scaledown_idletime ... 238
[vpc] Abschnitt ... 238
additional_sg ... 239
compute_subnet_cidr ... 239
compute_subnet_id ... 239
master_subnet_id ... 239
ssh_from ... 240
use_public_ips ... 240
vpc_id ... 240
vpc_security_group_id ... 241
Beispiele ... 40
Slurm Beispiel ... 242
SGE und Torque Beispiel ... 243
AWS Batch Beispiel ... 244
Wie AWS ParallelCluster funktioniert ... 246
AWS ParallelCluster Prozesse ... 246
SGE and Torque integration processes ... 247
Slurm integration processes ... 253
AWS Dienste, die genutzt werden von AWS ParallelCluster ... 253
AWS Auto Scaling ... 254
AWS Batch ... 255
AWS CloudFormation ... 255
Amazon CloudWatch ... 256
CloudWatch Amazon-Protokolle ... 256
AWS CodeBuild ... 256
Amazon-DynamoDB ... 257
Amazon Elastic Block Store ... 257
Amazon Elastic Compute Cloud ... 257
Amazon Elastic Container Registry ... 257
Amazon EFS ... 258
Amazon FSx für Lustre ... 258
AWS Identity and Access Management ... 258
AWS Lambda ... 259
Amazon DCV ... 259
Amazon Route 53 ... 259

Amazon Simple Notification Service ... 259
Amazon Simple Queue Service ... 260
Amazon Simple Storage Service ... 260
Amazon VPC ... 261
AWS ParallelCluster Auto Scaling ... 261
Hochskalieren ... 262
Herunterskalieren ... 263
Statischer Cluster ... 263
Tutorials ... 264
Du führst deinen ersten Job am AWS ParallelCluster ... 264
Überprüfen der Installation ... 264
Erstellen Sie Ihren ersten Cluster ... 265
Loggen Sie sich in Ihren Headnode ein ... 265
Führen Sie Ihren ersten Job aus mit SGE ... 266
Ein benutzerdefiniertes AWS ParallelCluster AMI erstellen ... 267
So passen Sie das AWS ParallelCluster AMI an ... 268
Ändern eines -AMIs ... 269
Erstellen Sie ein benutzerdefiniertes AWS ParallelCluster AMI ... 271
Verwenden eines benutzerdefinierten AMIs zur Laufzeit ... 273
Einen MPI-Job mit AWS ParallelCluster einem Scheduler ausführen `awsbatch` ... 273
Erstellen des Clusters ... 274
Loggen Sie sich in Ihren Hauptknoten ein ... 265
Führen Sie Ihren ersten Job aus mit AWS Batch ... 276
Ausführen eines MPI-Auftrags in einer parallelen Umgebung mit mehreren Knoten ... 278
Festplattenverschlüsselung mit einem benutzerdefinierten KMS-Schlüssel ... 282
Erstellen der -Rolle ... 282
Erteilen Sie Ihre Schlüsselberechtigungen ... 283
Erstellen des Clusters ... 274
Tutorial zum Modus mit mehreren Warteschlangen ... 284
Ihre Jobs im Modus AWS ParallelCluster mit mehreren Warteschlangen ausführen ... 284
Entwicklung ... 297
Ein benutzerdefiniertes AWS ParallelCluster Kochbuch einrichten ... 297
Schritte ... 298
Ein benutzerdefiniertes AWS ParallelCluster Knotenpaket einrichten ... 299
Schritte ... 299
Fehlerbehebung ... 301

Protokolle abrufen und aufbewahren ... 301
Behebung von Problemen bei der Stack-Bereitstellung ... 302
Behebung von Problemen in Clustern mit mehreren Warteschlangenmodi ... 303
Wichtige Protokolle ... 303
Behebung von Problemen mit der Knoteninitialisierung ... 304
Behebung unerwarteter Knotenersetzungen und -beendigungen ... 306
Probleminstanzen und Knoten ersetzen, beenden oder herunterfahren ... 308
Behebung anderer bekannter Knoten- und Jobprobleme ... 308
Behebung von Problemen in Clustern im Single-Queue-Modus ... 308
Wichtige Protokolle ... 309
Fehlerbehebung bei fehlgeschlagenen Start- und Verbindungsvorgängen ... 311
Behebung von Skalierungsproblemen ... 311
Behebung anderer Probleme im Zusammenhang mit Clustern ... 311
Probleme bei der Platzierung von Gruppen und beim Starten von Instances ... 312
Verzeichnisse, die nicht ersetzt werden können ... 312
Behebung von Problemen in Amazon DCV ... 313
Protokolle für Amazon DCV ... 314
Speicher vom Typ Amazon DCV-Instance ... 314
Probleme mit Ubuntu Amazon DCV ... 314
Behebung von Problemen in Clustern mit AWS Batch Integration ... 314
Probleme mit dem Hauptknoten ... 315
AWS Batch Probleme bei der Einreichung parallel Jobs mit mehreren Knoten ... 315
Probleme mit der Datenverarbeitung ... 315
Fehlschläge Job ... 315
Fehlerbehebung, wenn eine Ressource nicht erstellt werden kann ... 315
Behebung von Problemen mit der Größe der IAM-Richtlinie ... 317
Zusätzliche Unterstützung ... 317
AWS ParallelCluster Unterstützungspolitik ... 318
Sicherheit ... 319
Sicherheitsinformationen für Dienste, die genutzt werden von AWS ParallelCluster ... 320
Datenschutz ... 320
Datenverschlüsselung ... 321
Weitere Informationen finden Sie auch unter ... 323
Identitäts- und Zugriffsverwaltung ... 323
Compliance-Validierung ... 324
Erzwingen von TLS 1.2 ... 325

Ermitteln Ihrer derzeit unterstützten Protokolle 325
Kompilieren von OpenSSL und Python 327
Versionshinweise und Dokumentenverlauf 329
..... ccclxxiv

Was ist AWS ParallelCluster

AWS ParallelCluster ist ein AWS unterstütztes Open-Source-Cluster-Management-Tool, das Sie bei der Bereitstellung und Verwaltung von HPC-Clustern (High Performance Computing) in der AWS Cloud unterstützt. Es richtet automatisch die erforderlichen Rechenressourcen, den Scheduler und das gemeinsam genutzte Dateisystem ein. Sie können es mit und verwenden AWS ParallelCluster AWS Batch Slurm Scheduler.

Mit AWS ParallelCluster können Sie schnell HPC-Rechenumgebungen für Machbarkeitsstudien und Produktionsumgebungen erstellen und bereitstellen. Darüber hinaus können Sie auch einen umfassenden Workflow erstellen und implementieren AWS ParallelCluster, z. B. ein Genomik-Portal, das einen gesamten DNA-Sequenzierungs-Workflow automatisiert.

Preisgestaltung

Wenn Sie die AWS ParallelCluster Befehlszeilenschnittstelle (CLI) oder API verwenden, zahlen Sie nur für die AWS Ressourcen, die beim Erstellen oder Aktualisieren von AWS ParallelCluster Images und Clustern erstellt werden. Weitere Informationen finden Sie unter AWS Dienste, die genutzt werden von AWS ParallelCluster.

Einrichten AWS ParallelCluster

Themen

- Installation AWS ParallelCluster
- Konfiguration AWS ParallelCluster
- Bewährte Methoden
- Wechseln Sie von CfnCluster zu AWS ParallelCluster
- Unterstützte Regionen

Installation AWS ParallelCluster

AWS ParallelCluster wird als Python-Paket vertrieben und mit `pip` dem Python-Paketmanager installiert. Weitere Informationen zur Installation von Python-Paketen finden Sie unter Pakete installieren im Python Packaging User Guide.

Möglichkeiten zur Installation AWS ParallelCluster:

- Verwenden einer virtuellen Umgebung (empfohlen)
- Verwenden von `pip`

Die Versionsnummer der neuesten CLI finden Sie auf der Releases-Seite unter GitHub.

In diesem Handbuch wird bei den Beispielbefehlen davon ausgegangen, dass Sie Python v3 installiert haben. Die `pip`-Beispielbefehle verwenden die `pip3`-Version.

Installation AWS ParallelCluster in einer virtuellen Umgebung (empfohlen)

Wir empfehlen die Installation AWS ParallelCluster in einer virtuellen Umgebung. Wenn Sie bei der Installation AWS ParallelCluster mit auf Probleme stoßen`pip3`, können Sie die Installation AWS ParallelCluster in einer virtuellen Umgebung durchführen, um das Tool und seine Abhängigkeiten zu isolieren. Oder Sie können eine andere Python-Version verwenden, die Sie normalerweise nicht nutzen.

Installation AWS ParallelCluster in einer nicht-virtuellen Umgebung mit Pip

Die primäre Verteilungsmethode für AWS ParallelCluster Linux, Windows und MacOS ist `pip` ein Paketmanager für Python. Er bietet eine einfache Möglichkeit zum Installieren, Aktualisieren und Entfernen von Python-Paketen und ihren Abhängigkeiten.

Aktuelle AWS ParallelCluster Version

AWS ParallelCluster wird regelmäßig aktualisiert. Informationen darüber, ob Sie die neueste Version haben, finden Sie auf der Releases-Seite unter GitHub.

Wenn Sie bereits über `pip` eine unterstützte Version von Python verfügen, können Sie diese mit AWS ParallelCluster dem folgenden Befehl installieren. Wenn Sie Python-Version 3+ installiert haben, empfehlen wir Ihnen, den **`pip3`**-Befehl zu verwenden.

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

Nach der Installation zu ergreifende Schritte

Nach der Installation AWS ParallelCluster müssen Sie möglicherweise den Pfad der ausführbaren Datei zu Ihrer `PATH` Variablen hinzufügen. Plattformspezifische Anweisungen finden Sie in den folgenden Themen:

- Linux – Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu
- macOS – Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu
- Windows – Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu

Sie können überprüfen, ob die AWS ParallelCluster Installation korrekt ist, indem Sie Folgendes ausführen`pcluster version`.

```
$ pcluster version
2.11.9
```

AWS ParallelCluster wird regelmäßig aktualisiert. Um auf die neueste Version von zu aktualisieren AWS ParallelCluster, führen Sie den Installationsbefehl erneut aus. Einzelheiten zur neuesten Version von AWS ParallelCluster finden Sie in den AWS ParallelCluster Versionshinweisen.

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

Verwenden Sie zur Deinstallation AWS ParallelClusterpip uninstall.

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

Wenn Sie nicht über Python und pip verfügen, befolgen Sie die Anleitung für Ihre Umgebung.

Detaillierte Anweisungen für jede Umgebung

- AWS ParallelCluster In einer virtuellen Umgebung installieren (empfohlen)
- Unter AWS ParallelCluster Linux installieren
- AWS ParallelCluster Auf macOS installieren
- Unter Windows installieren AWS ParallelCluster

AWS ParallelCluster In einer virtuellen Umgebung installieren (empfohlen)

Wir empfehlen die Installation AWS ParallelCluster in einer virtuellen Umgebung, um Versionskonflikte mit anderen pip Paketen zu vermeiden.

Voraussetzungen

- Überprüfen Sie, ob pip und Python installiert sind. Wir empfehlenpip3, und Python 3 Version 3.8. Verwenden Sie bei Einsatz von Python 2 pip anstelle von pip3 und virtualenv anstelle von venv.

Zur Installation AWS ParallelCluster in einer virtuellen Umgebung

1. Wenn virtualenv nicht installiert ist, installieren Sie virtualenv mit pip3. Wenn python3 -m virtualenv help Hilfeinformationen anzeigt, fahren Sie mit Schritt 2 fort.

   Linux, macOS, or Unix

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

   Führen Sie exit aus, um das aktuelle Terminalfenster zu verlassen, und öffnen Sie ein neues Terminalfenster, um Änderungen an der Umgebung zu übernehmen.

Windows

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

Führen Sie aus `exit`, um die aktuelle Eingabeaufforderung zu verlassen, und öffnen Sie eine neue Eingabeaufforderung, um Änderungen an der Umgebung zu übernehmen.

2. Erstellen Sie eine virtuelle Umgebung und benennen Sie sie.

   Linux, macOS, or Unix

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

   Alternativ können Sie mit der `-p`-Option eine bestimmte Version von Python angeben.

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

   Windows

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

3. Aktivieren Sie die neue virtuelle Umgebung.

   Linux, macOS, or Unix

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

   Windows

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

4. AWS ParallelCluster In Ihrer virtuellen Umgebung installieren.

   Linux, macOS, or Unix

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

Windows

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

5. Stellen Sie sicher, dass AWS ParallelCluster es korrekt installiert ist.

   Linux, macOS, or Unix

   ```
   $ pcluster version
   2.11.9
   ```

   Windows

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

Sie können den Befehl `deactivate` verwenden, um die virtuelle Umgebung zu beenden. Jedes Mal, wenn Sie eine Sitzung starten, müssen Sie die Umgebung erneut aktivieren.

Um auf die neueste Version von zu aktualisieren AWS ParallelCluster, führen Sie den Installationsbefehl erneut aus.

Linux, macOS, or Unix

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

Windows

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

Unter AWS ParallelCluster Linux installieren

Sie können AWS ParallelCluster und seine Abhängigkeiten auf den meisten Linux-Distributionen installierenpip, indem Sie einen Paketmanager für Python verwenden. Bestimmen Sie zuerst, ob Python und `pip` installiert sind:

1. Um festzustellen, ob Ihre Version von Linux Python und `pip` enthält, führen Sie `pip --version` aus.

```
$ pip --version
```

Wenn Sie `pip` installiert haben, fahren Sie mit dem Thema Installation AWS ParallelCluster mit Pip fort. Andernfalls machen Sie mit Schritt 2 weiter.

2. Um festzustellen, ob Python installiert ist, führen Sie `python --version` aus.

```
$ python --version
```

Wenn Sie Python 3 Version 3.6+ oder Python 2 Version 2.7 installiert haben, fahren Sie mit dem Thema Installation AWS ParallelCluster mit pip fort. Andernfalls installieren Sie Python und kehren dann zu diesem Verfahren zurück, um `pip` zu installieren.

3. Installieren Sie `pip` mithilfe des Skripts der Python Packaging Authority.
4. Laden Sie das Installationsskript mit dem Befehl `curl` herunter.

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

5. Führen Sie das Skript mit Python aus, um die aktuelle Version von `pip` und andere erforderliche Supportpakete herunterzuladen.

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

or

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

Wenn Sie den Schalter `--user` einschließen, installiert das Skript `pip` im Pfad `~/.local/bin`.

6. Gehen Sie wie folgt vor, um zu überprüfen, ob der Ordner, der enthält, Teil Ihrer `PATH` Variablen `pip` ist:

   a. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `basename $SHELL` aus.

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

   - Bash – `.bash_profile`, `.profile` oder `.bash_login`

- Zsh – `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` oder `.login`.

b. Fügen Sie einen Exportbefehl an das Ende Ihres Profilskripts hinzu und orientieren Sie sich dabei an folgendem Beispiel.

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

Der Exportbefehl fügt den Pfad, der in diesem Beispiel `~/.local/bin` ist, am Anfang der bestehenden Variablen `PATH` ein.

c. Laden Sie das Profil erneut in Ihre aktuelle Sitzung, damit die Änderungen wirksam werden.

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

7. Stellen Sie sicher, dass `pip` korrekt installiert ist.

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

Sections

- Installieren Sie AWS ParallelCluster mit pip
- Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu
- Installieren von Python auf Linux

Installieren Sie AWS ParallelCluster mit **`pip`**

`pip`Zum Installieren verwenden AWS ParallelCluster.

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

Wenn Sie den Schalter `--user` verwenden, installiert `pip` AWS ParallelCluster in `~/.local/bin`.

Stellen Sie sicher, dass es korrekt AWS ParallelCluster installiert wurde.

```
$ pcluster version
2.11.9
```

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

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

Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu

Nach der Installation mit `pip` müssen Sie möglicherweise die ausführbare `pcluster`-Datei zur `PATH`-Umgebungsvariablen des Betriebssystems hinzufügen.

Führen Sie den folgenden Befehl aus, um den Ordner zu überprüfen AWS ParallelCluster, in dem die `pip` Installation installiert ist.

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

Wenn Sie den `--user` Schalter bei der Installation weggelassen haben AWS ParallelCluster, befindet sich die ausführbare Datei möglicherweise im `bin` Ordner Ihrer Python-Installation. Wenn Sie nicht wissen, wo Python installiert ist, führen Sie diesen Befehl aus.

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

Die Ausgabe ist möglicherweise der Pfad zu einem Symlink, nicht zu der tatsächlichen ausführbaren Datei. Um zu sehen, worauf der Symlink verweist, führen Sie `ls -al` aus.

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

Wenn dies der gleiche Ordner ist, den Sie in Schritt 3 in Installation AWS ParallelCluster zum Pfad hinzugefügt haben, sind Sie fertig mit der Installation. Andernfalls müssen Sie die Schritte 3a — 3c erneut ausführen und diesen zusätzlichen Ordner zum Pfad hinzufügen.

Installieren von Python auf Linux

Wenn Ihre Distribution nicht mit Python oder mit einer früheren Version geliefert wurde, installieren Sie Python vor der Installation von `pip` und AWS ParallelCluster.

So installieren Sie Python 3 unter Linux

1. Überprüfen Sie, ob Python bereits installiert ist.

```
$ python3 --version
```

or

```
$ python --version
```

> **Note**
>
> Wenn Ihre Linux-Verteilung Python bereits enthielt, müssen Sie möglicherweise das Python-Entwicklerpaket installieren. Das Entwicklerpaket enthält die für die Kompilierung von Erweiterungen und die Installation von AWS ParallelCluster benötigten Header und Bibliotheken. Verwenden Sie Ihren Paket-Manager, um das Entwicklerpaket zu installieren. Es heißt in der Regel `python-dev` oder `python-devel`.

2. Wenn kein Python 2.7 oder höher installiert ist, installieren Sie Python mit dem Paket-Manager Ihrer Verteilung. Der Befehl und der Paketname variieren:

   - Auf Debian-Derivaten wie Ubuntu verwenden Sie `apt`.

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

   - Auf Red Hat und seinen Derivaten verwenden Sie `yum`.

     ```
     $ sudo yum install python3
     ```

   - Verwenden Sie auf SUSE und Derivaten `zypper`.

     ```
     $ sudo zypper install python3
     ```

3. Öffnen Sie eine Eingabeaufforderung oder Shell, und führen Sie den folgenden Befehl aus, um zu überprüfen, ob Python korrekt installiert ist.

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

AWS ParallelCluster Auf macOS installieren

Sections

- Voraussetzungen
- Installation AWS ParallelCluster auf macOS mit pip
- Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu

Voraussetzungen

- Python 3 Version 3.7+ oder Python 2 Version 2.7

Überprüfen Sie Ihre Python-Installation.

```
$ python --version
```

Wenn auf Ihrem Computer noch kein Python installiert ist oder Sie eine andere Version von Python installieren möchten, befolgen Sie die Anleitung im Abschnitt Unter AWS ParallelCluster Linux installieren.

Installation AWS ParallelCluster auf macOS mit pip

Sie können es auch `pip` direkt zur Installation AWS ParallelCluster verwenden. Falls nicht über `pip` verfügen, befolgen Sie die Anleitungen im Installations-Hauptthema. Führen Sie `pip3 --version` aus, um zu überprüfen, ob Ihre Version von macOS bereits Python und `pip3` enthält.

```
$ pip3 --version
```

Zur Installation AWS ParallelCluster auf macOS

1. Laden Sie die neueste Version von Python von der Downloads-Seite von Python.org herunter und installieren Sie sie.
2. Laden Sie das von der Python Packaging Authority bereitgestellte `pip3`-Installationsskript herunter und führen Sie es aus.

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

3. Verwenden Sie für die Installation Ihr neu installiertes `pip3` AWS ParallelCluster. Wir empfehlen Ihnen, bei Verwendung von Python-Version 3+ den `pip3`-Befehl zu verwenden.

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

4. Stellen Sie sicher, dass AWS ParallelCluster es korrekt installiert ist.

```
$ pcluster version
2.11.9
```

   Wenn das Programm nicht gefunden wird, fügen Sie es dem Befehlszeilenpfad hinzu.

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

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

Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu

Nach der Installation mit `pip` müssen Sie möglicherweise das Programm `pcluster` zur PATH-Umgebungsvariablen des Betriebssystems hinzufügen. Der Speicherort des Programms hängt davon ab, wo Python installiert ist.

Example AWS ParallelCluster Installationsort — macOS mit Python 3.6 und **`pip`** (Benutzermodus)

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

Geben Sie anstelle der Version in dem Beispiel oben Ihre Python-Version an.

Wenn Sie nicht wissen, wo Python installiert ist, führen Sie `which python` aus

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

Die Ausgabe ist möglicherweise der Pfad zu einem Symlink, nicht der zum tatsächlichen Programm. Führen Sie `ls -al` aus, um festzustellen, wohin dieser verweist.

```
$ ls -al /usr/local/bin/python3
```

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

`pip` installiert Programme in demselben Ordner, der auch die Python-Anwendung enthält. Fügen Sie diesen Ordner Ihrer `PATH`-Variablen hinzu.

Um deine **`PATH`** Variable zu ändern (Linux, macOS oder Unix)

1. Suchen Sie das Profilskript für die Shell in Ihrem Benutzerordner. Wenn Sie nicht sicher sind, welche Shell Sie haben, führen Sie `echo $SHELL` aus.

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

   - Bash – `.bash_profile`, `.profile` oder `.bash_login`
   - Zsh – `.zshrc`
   - Tcsh —`.tcshrc`,`.cshrc`, oder `.login`
2. Fügen Sie dem Profilskript einen Exportbefehl hinzu.

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

   Dieser Befehl fügt der aktuellen `PATH`-Variablen einen Pfad, in diesem Beispiel `~/.local/bin`, hinzu.
3. Laden Sie das Profil in der aktuellen Sitzung.

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

Unter Windows installieren AWS ParallelCluster

Sie können AWS ParallelCluster unter Windows installieren`pip`, indem Sie, einen Paketmanager für Python, verwenden. Wenn Sie bereits über `pip` verfügen, befolgen Sie die Anweisungen im Hauptthema zur Installation.

Sections

- Installation AWS ParallelCluster mit Python und pip unter Windows
- Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu

Installation AWS ParallelCluster mit Python und **`pip`** unter Windows

Die Python Software Foundation bietet Installationsprogramme für Windows, die `pip` beinhalten.

So installieren Sie Python und **`pip`** (Windows)

1. Laden Sie die Installationsdatei für Python Windows x86-64 von der Downloads-Seite von Python.org herunter.
2. Führen Sie das Installationsprogramm aus.
3. Wählen Sie Python 3 zu PATH hinzufügen aus.
4. Wählen Sie Install Now (Jetzt installieren) aus.

Das Installationsprogramm installiert Python in Ihrem Benutzerordner und fügt die Programmordner Ihrem Benutzerpfad hinzu.

Zur Installation AWS ParallelCluster mit **`pip3`** (Windows)

Bei Verwendung von Python-Version 3+ empfehlen wir Ihnen, den `pip3`-Befehl zu verwenden.

1. Öffnen Sie dieEingabeaufforderung über das Start-Menü.
2. Verwenden Sie die folgenden Befehle, um zu prüfen, ob Python und `pip` korrekt installiert sind.

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

3. Installation AWS ParallelCluster mit`pip`.

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

4. Stellen Sie sicher, dass AWS ParallelCluster es korrekt installiert ist.

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

Führen Sie das Installationsprogramm erneut aus, um auf die neueste Version zu aktualisieren.

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

Fügen Sie die AWS ParallelCluster ausführbare Datei zu Ihrem Befehlszeilenpfad hinzu

Fügen Sie das `pcluster` Programm nach der Installation AWS ParallelCluster mit `pip` der `PATH` Umgebungsvariablen Ihres Betriebssystems hinzu.

Mit dem folgenden Befehl finden Sie heraus, wo das `pcluster`-Programm installiert ist.

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

Wenn dieser Befehl keine Ergebnisse zurückgibt, müssen Sie den Pfad manuell hinzufügen. Verwenden Sie die Befehlszeile oder Windows Explorer, um herauszufinden, wo es auf Ihrem Computer installiert ist. Typische Pfade sind beispielsweise:

- Python 3 und **pip3** – `C:\Python38\Scripts\`
- Python 3 und Option **pip3** --user — `%APPDATA%\Python\Python38\Scripts`

> **Note**
>
> Ordnernamen mit Versionsnummern, die sich ändern können. Die vorherigen Beispiele zeigen Python38. Ersetzen Sie nach Bedarf mit der Versionsnummer, die Sie verwenden.

Ändern der PATH-Variablen (Windows)

1. Betätigen Sie die Windows-Taste und geben Sie **`environment variables`** ein.
2. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
3. Wählen Sie PATH (PFAD) und Edit (Bearbeiten) aus.
4. Fügen Sie den Pfad dem Feld Variablenwert hinzu. Beispiel: ***`C:\new\path`***
5. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
6. Schließen Sie alle laufenden Eingabeaufforderungen und öffnen Sie das Eingabeaufforderungsfenster erneut.

Konfiguration AWS ParallelCluster

Führen Sie nach der Installation AWS ParallelCluster die folgenden Konfigurationsschritte aus.

Stellen Sie sicher, dass Ihr AWS Konto über eine Rolle verfügt, die die für die Ausführung der `pcluster` CLI erforderlichen Berechtigungen umfasst. Weitere Informationen finden Sie unter AWS ParallelCluster Beispiel für Instanz- und Benutzerrichtlinien.

Richten Sie Ihre AWS Anmeldeinformationen ein. Weitere Informationen finden Sie unter Konfigurieren der AWS CLI im AWS CLI -Benutzerhandbuch.

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

Der AWS-Region Ort, an dem der Cluster gestartet wird, muss mindestens ein EC2 Amazon-Schlüsselpaar haben. Weitere Informationen finden Sie unter EC2 Amazon-Schlüsselpaare im EC2 Amazon-Benutzerhandbuch.

```
$ pcluster configure
```

Der Konfigurationsassistent fordert Sie zur Eingabe aller Informationen auf, die zum Erstellen Ihres Clusters benötigt werden. Die Einzelheiten der Reihenfolge unterscheiden sich bei der Verwendung AWS Batch als Scheduler von der Verwendung Slurm. Weitere Hinweise zu einer Clusterkonfiguration finden Sie unterKonfiguration.

> **Note**
>
> Ab Version 2.11.5 wird die AWS ParallelCluster Verwendung von nicht unterstützt SGE or Torque Scheduler. Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.

Slurm

Wählen Sie aus der Liste der gültigen AWS-Region Kennungen den Ort aus, AWS-Region an dem Ihr Cluster ausgeführt werden soll.

> **Note**
>
> Die AWS-Regionen angezeigte Liste basiert auf der Partition Ihres Kontos und enthält nur die Partitionen, AWS-Regionen die für Ihr Konto aktiviert sind. Weitere Informationen zur Aktivierung AWS-Regionen für Ihr Konto finden Sie unter Verwaltung AWS-Regionen in der Allgemeine AWS-Referenz. Das gezeigte Beispiel stammt aus der AWS globalen Partition. Wenn sich Ihr Konto in der AWS GovCloud (US) Partition befindet, werden nur AWS-Regionen in dieser Partition aufgeführt (`gov-us-east-1`und`gov-us-west-1`). Ebenso, wenn sich Ihr Konto in der AWS China Partition befindet, `cn-northwest-1` werden nur `cn-north-1` und angezeigt. Eine vollständige Liste der AWS-Regionen unterstützten AWS ParallelCluster Programme finden Sie unterUnterstützte Regionen.

```
Allowed values for the AWS-Region 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
AWS-Region ID [ap-northeast-1]:
```

Wählen Sie den Scheduler aus, der mit dem Cluster verwendet werden soll.

```
Allowed values for Scheduler:
1. slurm
```

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

Wählen Sie das Betriebssystem aus.

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

> **Note**
>
> Support für `alinux2` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

Die minimale und maximale Größe des Clusters für Datenverarbeitungsknoten wird eingegeben. Dies wird anhand der Instances-Anzahl gemessen.

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

Die Instanztypen Head und Compute Nodes werden eingegeben. Bei Instance-Typen sind Ihre Konto-Instance-Limits groß genug, um Ihre Anforderungen zu erfüllen. Weitere Informationen finden Sie unter Limits für On-Demand-Instances im EC2 Amazon-Benutzerhandbuch.

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

Das key pair wird aus den bei Amazon registrierten Schlüsselpaaren EC2 im ausgewählten Bereich ausgewählt AWS-Region.

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

Nachdem die vorherigen Schritte abgeschlossen sind, entscheiden Sie, ob Sie eine vorhandene VPC verwenden oder eine VPC für AWS ParallelCluster Sie erstellen lassen möchten. Wenn

Sie keine ordnungsgemäß konfigurierte VPC haben, AWS ParallelCluster können Sie eine neue erstellen. Es verwendet entweder sowohl den Head- als auch den Compute-Knoten im selben öffentlichen Subnetz oder nur den Head-Knoten in einem öffentlichen Subnetz mit allen Knoten in einem privaten Subnetz. Es ist möglich, Ihr Limit für die Anzahl von VPCs in a zu erreichen. AWS-Region Das Standardlimit beträgt jeweils fünf VPCs AWS-Region. Weitere Informationen zu diesem Limit und dazu, wie Sie eine Erhöhung beantragen können, finden Sie unter VPC und Subnetze im Amazon VPC-Benutzerhandbuch.

Wenn Sie eine VPC AWS ParallelCluster erstellen lassen, müssen Sie entscheiden, ob sich alle Knoten in einem öffentlichen Subnetz befinden sollen.

> **Important**
>
> VPCs erstellt von VPC Flow Logs standardmäßig AWS ParallelCluster nicht aktivieren. VPC Flow Logs ermöglichen es Ihnen, Informationen über den IP-Verkehr zu und von Netzwerkschnittstellen in Ihrem VPCs zu erfassen. Weitere Informationen finden Sie unter VPC-Flow-Protokolle im Amazon-VPC-Benutzerhandbuch.

> **Note**
>
> Wenn Sie möchten`1. Master in a public subnet and compute fleet in a private subnet`, AWS ParallelCluster wird ein NAT-Gateway erstellt, das zusätzliche Kosten verursacht, auch wenn Sie Ressourcen mit einem kostenlosen Kontingent angeben.

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

Wenn Sie keine neue VPC erstellen, müssen Sie eine vorhandene VPC auswählen.

Wenn Sie die VPC AWS ParallelCluster erstellen möchten, notieren Sie sich die VPC-ID, damit Sie sie später AWS CLI löschen können.

```
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
```

Nachdem die VPC ausgewählt wurde, müssen Sie entscheiden, ob Sie vorhandene Subnetze verwenden oder neue erstellen möchten.

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

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

AWS Batch

Wählen Sie aus der Liste der gültigen AWS-Region Kennungen den Ort aus, AWS-Region an dem Ihr Cluster ausgeführt werden soll.

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

```
AWS-Region ID [ap-northeast-1]:
```

Wählen Sie den Scheduler aus, der mit dem Cluster verwendet werden soll.

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

Wenn `awsbatch` als Scheduler ausgewählt wird, wird `alinux2` als Betriebssystem verwendet.

Die minimale und maximale Größe des Clusters für Datenverarbeitungsknoten wird eingegeben. Dies wird in v CPUs gemessen.

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

Der Instanztyp des Kopfknotens ist eingegeben. Bei Verwendung des `awsbatch`-Schedulers nutzen die Datenverarbeitungsknoten den Instance-Typ „`optimal`“.

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

Das EC2 Amazon-Schlüsselpaar wird aus den bei Amazon registrierten Schlüsselpaaren EC2 im ausgewählten Bereich ausgewählt AWS-Region.

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

Entscheiden Sie, ob Sie vorhandene verwenden VPCs oder die AWS ParallelCluster Erstellung VPCs für Sie übernehmen lassen möchten. Wenn Sie keine ordnungsgemäß konfigurierte VPC haben, AWS ParallelCluster können Sie eine neue erstellen. Es verwendet entweder sowohl den Head- als auch den Compute-Knoten im selben öffentlichen Subnetz oder nur den Head-Knoten in einem öffentlichen Subnetz mit allen Knoten in einem privaten Subnetz. Es ist möglich, Ihr Limit für die Anzahl von VPCs in a zu erreichen. AWS-Region Die Standardzahl von VPCs ist fünf. Weitere Informationen zu diesem Limit und dazu, wie Sie eine Erhöhung beantragen können, finden Sie unter VPC und Subnetze im Amazon VPC-Benutzerhandbuch.

> **Important**
>
> VPCs erstellt von VPC Flow Logs standardmäßig AWS ParallelCluster nicht aktivieren. VPC Flow Logs ermöglichen es Ihnen, Informationen über den IP-Verkehr zu und von Netzwerkschnittstellen in Ihrem VPCs zu erfassen. Weitere Informationen finden Sie unter VPC-Flow-Protokolle im Amazon-VPC-Benutzerhandbuch.

Wenn Sie eine VPC AWS ParallelCluster erstellen lassen, entscheiden Sie, ob sich alle Knoten in einem öffentlichen Subnetz befinden sollen.

> **Note**
>
> Wenn Sie möchten`1. Master in a public subnet and compute fleet in a private subnet`, AWS ParallelCluster wird ein NAT-Gateway erstellt, das zusätzliche Kosten verursacht, auch wenn Sie Ressourcen im kostenlosen Kontingent angeben.

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

Wenn Sie keine neue VPC erstellen, müssen Sie eine vorhandene VPC auswählen.

Wenn Sie die VPC AWS ParallelCluster erstellen möchten, notieren Sie sich die VPC-ID, damit Sie sie später AWS CLI löschen können.

```
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
```

Nachdem die VPC ausgewählt wurde, entscheiden Sie, ob Sie vorhandene Subnetze verwenden oder neue erstellen möchten.

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

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

Wenn Sie die vorherigen Schritte abgeschlossen haben, wird ein einfacher Cluster in einer VPC gestartet. Die VPC verwendet ein vorhandenes Subnetz, das öffentliche IP-Adressen unterstützt. Die Routing-Tabelle für das Subnetz lautet. `0.0.0.0/0 => igw-xxxxxx` Beachten Sie die folgenden Bedingungen:

- Die VPC muss `DNS Resolution = yes` und `DNS Hostnames = yes` lauten.
- Die VPC sollte auch über DHCP-Optionen mit den richtigen `domain-name` für verfügen. AWS-Region Der standardmäßige DHCP-Optionssatz spezifiziert bereits die erforderlichen AmazonProvidedDNS. Wenn Sie mehr als einen Domain-Namenserver angeben, finden Sie weitere Informationen unter DHCP-Optionssätze im Amazon VPC-Benutzerhandbuch. Wenn Sie private Subnetze verwenden, verwenden Sie ein NAT-Gateway oder einen internen Proxy, um den Webzugriff für Rechenknoten zu aktivieren. Weitere Informationen finden Sie unter Netzwerkkonfigurationen.

Wenn alle diese Einstellungen gültige Werte enthalten, können Sie den Cluster starten, indem Sie den Befehl „create“ ausführen.

```
$ pcluster create mycluster
```

Sobald der Cluster den Status „CREATE_COMPLETE“ erreicht hat, können Sie mit den normalen SSH-Client-Einstellungen eine Verbindung herstellen. Weitere Informationen zur Verbindung mit EC2 Amazon-Instances finden Sie im EC2 Benutzerhandbuch im EC2 Amazon-Benutzerhandbuch.

Führen Sie den folgenden Befehl aus, um den Cluster zu löschen.

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

Um die Netzwerkressourcen in der VPC zu löschen, können Sie den CloudFormation Netzwerkstapel löschen. Der Stack-Name beginnt mit“parallelclusternetworking-"und enthält die Erstellungszeit im Format „YYYYMMDDHHMMSS“. Sie können die Stapel mit dem Befehl list-stacks auflisten.

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

Der Stapel kann mit dem Befehl delete-stack gelöscht werden.

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

Die VPC, die für Sie `pcluster configure` erstellt, wird nicht im CloudFormation Netzwerk-Stack erstellt. Sie können diese VPC manuell in der Konsole oder mithilfe von löschen. AWS CLI

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

Bewährte Methoden

Bewährte Methoden: Auswahl des Master-Instance-Typs

Obwohl der Master-Knoten keinen Job ausführt, sind seine Funktionen und seine Größe entscheidend für die Gesamtleistung des Clusters.

Bei der Auswahl des Instanztyps, den Sie für Ihren Master-Knoten verwenden möchten, sollten Sie die folgenden Punkte berücksichtigen:

- Clustergröße: Der Master-Knoten orchestriert die Skalierungslogik des Clusters und ist dafür verantwortlich, dem Scheduler neue Knoten zuzuordnen. Wenn Sie den Cluster mit einer beträchtlichen Anzahl von Knoten nach oben oder unten skalieren müssen, sollten Sie dem Master-Knoten zusätzliche Rechenkapazität zur Verfügung stellen.
- Gemeinsam genutzte Dateisysteme: Wenn Sie gemeinsam genutzte Dateisysteme verwenden, um Artefakte zwischen Rechenknoten und dem Master-Knoten gemeinsam zu nutzen, sollten Sie berücksichtigen, dass der Master der Knoten ist, der den NFS-Server verfügbar macht. Aus diesem

Grund sollten Sie einen Instance-Typ mit ausreichender Netzwerkbandbreite und ausreichend dedizierter Amazon EBS-Bandbreite für Ihre Workflows wählen.

Bewährte Methoden: Netzwerkleistung

Es gibt drei Hinweise, die das gesamte Spektrum der Möglichkeiten zur Verbesserung der Netzwerkkommunikation abdecken.

- Platzierungsgruppe: Eine Cluster-Platzierungsgruppe ist eine logische Gruppierung von Instances innerhalb einer einzigen Availability Zone. Weitere Informationen zu Platzierungsgruppen finden Sie unter Platzierungsgruppen im EC2 Amazon-Benutzerhandbuch. Sie können den Cluster so konfigurieren, dass er Ihre eigene Platzierungsgruppe verwendet, `placement_group = your-placement-group-name` oder Sie können eine Platzierungsgruppe mit der `"compute"` Strategie mit AWS ParallelCluster erstellen lassen`placement_group = DYNAMIC`. Weitere Informationen finden Sie unter `placement_group` Für den Modus mit mehreren Warteschlangen und `placement_group` für den Modus mit einer einzelnen Warteschlange.
- Enhanced Networking: Erwägen Sie, einen Instance-Typ zu wählen, der Enhanced Networking unterstützt. Weitere Informationen finden Sie unter Enhanced Networking on Linux im EC2 Amazon-Benutzerhandbuch.
- Elastic Fabric Adapter: Um ein hohes Maß an skalierbarer Kommunikation zwischen Instanzen zu unterstützen, sollten Sie die Wahl von EFA-Netzwerkschnittstellen für Ihr Netzwerk in Betracht ziehen. Die von EFA entwickelte Hardware zur Umgehung von Betriebssystemen (OS) verbessert die Kommunikation zwischen den Instanzen mit der On-Demand-Elastizität und Flexibilität der Cloud. AWS Um ein einzelnes zu konfigurieren Slurm Cluster-Warteschlange zur Verwendung von EFA, festgelegt`enable_efa = true`. Weitere Hinweise zur Verwendung von EFA mit finden Sie AWS ParallelCluster unter Elastic Fabric Adapter und. `enable_efa` Weitere Informationen zu EFA finden Sie unter Elastic Fabric Adapter im EC2 Amazon-Benutzerhandbuch für Linux-Instances.
- Instance-Bandbreite: Die Bandbreite skaliert mit der Instance-Größe. Bitte wählen Sie den Instance-Typ, der Ihren Anforderungen besser entspricht. Weitere Informationen finden Sie unter Amazon EBS-optimierte Instances und Amazon EBS-Volumetypen im EC2 Amazon-Benutzerhandbuch.

Bewährte Methoden: Budgetwarnungen

Um die AWS ParallelCluster Ressourcenkosten zu verwalten, empfehlen wir Ihnen, mithilfe von AWS Budgets Aktionen Benachrichtigungen über Budgets und definierte Budgetschwellenwerte

für ausgewählte AWS Ressourcen zu erstellen. Weitere Informationen finden Sie im AWS Budgets Benutzerhandbuch unter Konfiguration einer Budgetaktion. Sie können Amazon auch verwenden CloudWatch , um einen Abrechnungsalarm zu erstellen. Weitere Informationen finden Sie unter Einen Abrechnungsalarm erstellen, um Ihre geschätzten AWS Gebühren zu überwachen.

Bewährte Methoden: Umstellung eines Clusters auf eine neue AWS ParallelCluster Minor- oder Patch-Version

Derzeit ist jede AWS ParallelCluster Nebenversion zusammen mit ihrer `pcluster` CLI eigenständig. Um einen Cluster auf eine neue Minor- oder Patch-Version zu verschieben, müssen Sie den Cluster mithilfe der CLI der neuen Version neu erstellen.

Um den Prozess der Migration eines Clusters auf eine neue Nebenversion zu optimieren oder Ihre gemeinsam genutzten Speicherdaten aus anderen Gründen zu speichern, empfehlen wir Ihnen, die folgenden bewährten Methoden zu verwenden.

- Speichern Sie persönliche Daten in externen Volumes wie Amazon EFS und FSx für Lustre. Auf diese Weise können Sie die Daten problemlos von einem Cluster in einen anderen verschieben.
- Erstellen Sie gemeinsam genutzte Speichersysteme der unten aufgeführten Typen mit dem AWS CLI oder AWS Management Console:
  - `[ebs]` Abschnitt
  - `[efs]` Abschnitt
  - `[fsx]` Abschnitt

  Fügen Sie sie der neuen Cluster-Konfiguration als bestehende Dateisysteme hinzu. Auf diese Weise bleiben sie erhalten, wenn Sie den Cluster löschen, und können an einen neuen Cluster angehängt werden. Für gemeinsam genutzte Speichersysteme fallen im Allgemeinen Gebühren an, unabhängig davon, ob sie an einen Cluster angeschlossen oder von diesem getrennt sind.

  Wir empfehlen die Verwendung von Amazon EFS- oder Amazon FSx for Lustre-Dateisystemen, da diese gleichzeitig an mehrere Cluster angehängt werden können und Sie sie an den neuen Cluster anhängen können, bevor Sie den alten Cluster löschen. Weitere Informationen finden Sie unter Mounten von Amazon EFS-Dateisystemen im Amazon EFS-Benutzerhandbuch und Zugreifen auf FSx Lustre-Dateisysteme im Amazon FSx for Lustre-Benutzerhandbuch.
- Verwenden Sie anstelle eines benutzerdefinierten AMI benutzerdefinierte Bootstrap-Aktionen, um Ihre Instances anzupassen. Dadurch wird der Erstellungsprozess optimiert, da nicht für jede neue Version ein neues benutzerdefiniertes AMI erstellt werden muss.

- Empfohlene Reihenfolge.
  1. Aktualisieren Sie die Clusterkonfiguration, um vorhandene Dateisystemdefinitionen zu verwenden.
  2. Überprüfen Sie die `pcluster` Version und aktualisieren Sie sie bei Bedarf.
  3. Erstellen und testen Sie den neuen Cluster.
     - Stellen Sie sicher, dass Ihre Daten im neuen Cluster verfügbar sind.
     - Stellen Sie sicher, dass Ihre Anwendung im neuen Cluster funktioniert.
  4. Wenn Ihr neuer Cluster vollständig getestet und betriebsbereit ist und Sie sicher sind, dass Sie den alten Cluster nicht verwenden werden, löschen Sie ihn.

Wechseln Sie von CfnCluster zu AWS ParallelCluster

AWS ParallelCluster ist eine erweiterte Version von CfnCluster.

Wenn Sie es derzeit verwenden CfnCluster, empfehlen wir, es AWS ParallelCluster stattdessen zu verwenden und damit neue Cluster zu erstellen. Sie können es zwar weiterhin verwenden CfnCluster, es wird jedoch nicht mehr weiterentwickelt und es werden keine neuen Features oder Funktionen hinzugefügt.

Die Hauptunterschiede zwischen CfnCluster und AWS ParallelCluster werden in den folgenden Abschnitten beschrieben.

AWS ParallelCluster CLI verwaltet eine andere Gruppe von Clustern

Mit der `cfncluster` CLI erstellte Cluster können nicht mit der `pcluster` CLI verwaltet werden. Die folgenden Befehle funktionieren nicht auf Clustern, die erstellt wurden von CfnCluster:

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

Um Cluster zu verwalten, mit denen Sie erstellt haben CfnCluster, müssen Sie die `cfncluster` CLI verwenden.

Wenn Sie ein CfnCluster Paket zur Verwaltung Ihrer alten Cluster benötigen, empfehlen wir Ihnen, es in einer virtuellen Python-Umgebung zu installieren und zu verwenden.

AWS ParallelCluster und CfnCluster verwenden Sie verschiedene benutzerdefinierte IAM-Richtlinien

Benutzerdefinierte IAM-Richtlinien, die zuvor für die CfnCluster Clustererstellung verwendet wurden, können nicht mit verwendet werden. AWS ParallelCluster Wenn Sie benutzerdefinierte Richtlinien für benötigen AWS ParallelCluster, müssen Sie neue erstellen. Sehen Sie sich den AWS ParallelCluster Leitfaden an.

AWS ParallelCluster und CfnCluster verwende verschiedene Konfigurationsdateien

Die AWS ParallelCluster Konfigurationsdatei befindet sich im `~/.parallelcluster` Ordner. Die CfnCluster-Konfigurationsdatei befindet sich im Ordner `~/.cfncluster`.

Wenn Sie eine vorhandene CfnCluster Konfigurationsdatei mit verwenden möchten AWS ParallelCluster, müssen Sie die folgenden Aktionen ausführen:

1. die Konfigurationsdatei von `~/.cfncluster/config` nach `~/.parallelcluster/config` verschieben.
2. Wenn Sie die `extra_json`-Konfigurationsparameter verwenden, ändern Sie sie wie in der folgenden Abbildung dargestellt.

CfnCluster Einstellung:

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

AWS ParallelCluster Einstellung:

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

AWS ParallelCluster In ist Ganglia standardmäßig deaktiviert

In AWS ParallelCluster sind Ganglien standardmäßig deaktiviert. Gehen Sie wie folgt vor, um Ganglien zu aktivieren:

1. Legen Sie den Parameter `extra_json` wie folgt fest:

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

2. Ändern Sie die Hauptsicherheitsgruppe, um Verbindungen zu Port 80 zuzulassen.

Die `parallelcluster-<CLUSTER_NAME>-MasterSecurityGroup-<xxx>`-Sicherheitsgruppe muss geändert werden, indem eine neue Sicherheitsgruppenregel hinzugefügt wird, um eine eingehende Verbindung mit Port 80 von Ihrer öffentlichen IP-Adresse aus zuzulassen. Weitere Informationen finden Sie unter Regeln zu einer Sicherheitsgruppe hinzufügen im EC2 Amazon-Benutzerhandbuch.

Unterstützte Regionen

AWS ParallelCluster Version 2.x ist in den folgenden AWS-Regionen Versionen verfügbar:

Name der Region	Region
USA Ost (Ohio)	us-east-2
USA Ost (Nord-Virginia)	us-east-1
USA West (Nordkalifornien)	us-west-1
USA West (Oregon)	us-west-2
Afrika (Kapstadt)	af-south-1
Asien-Pazifik (Hongkong)	ap-east-1
Asien-Pazifik (Mumbai)	ap-south-1
Asien-Pazifik (Seoul)	ap-northeast-2
Asien-Pazifik (Singapur)	ap-southeast-1
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1
Kanada (Zentral)	ca-central-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1

Name der Region	Region
Europe (Frankfurt)	eu-central-1
Europa (Irland)	eu-west-1
Europa (London)	eu-west-2
Europa (Mailand)	eu-south-1
Europa (Paris)	eu-west-3
Europa (Stockholm)	eu-north-1
Naher Osten (Bahrain)	me-south-1
Südamerika (São Paulo)	sa-east-1
AWS GovCloud (US-Ost)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1

Benutzen AWS ParallelCluster

Themen

- Netzwerkkonfigurationen
- Benutzerdefinierte Bootstrap-Aktionen
- Arbeiten mit Amazon S3
- Arbeiten mit Spot-Instances
- AWS Identity and Access Management Rollen in AWS ParallelCluster
- Scheduler werden unterstützt von AWS ParallelCluster
- AWS ParallelCluster Ressourcen und Tagging
- CloudWatch Amazon-Dashboard
- Integration mit Amazon CloudWatch Logs
- Elastic Fabric Adapter
- Intel Select Solutions
- Intel MPI aktivieren
- Intel HPC-Plattform-Spezifikation
- ARM-Leistungsbibliotheken
- Stellen Sie über Amazon DCV eine Connect zum Hauptknoten her
- Verwenden von pcluster update
- AMI-Patching und EC2 Instanzersatz

Netzwerkkonfigurationen

AWS ParallelCluster verwendet Amazon Virtual Private Cloud (VPC) für Netzwerke. VPC bietet eine flexible und konfigurierbare Netzwerkplattform, auf der Sie Cluster bereitstellen können.

Die VPC muss über `DNS Resolution = yes` DHCP-Optionen mit dem richtigen Domainnamen für die Region verfügen. `DNS Hostnames = yes` Der standardmäßige DHCP-Optionssatz spezifiziert bereits das erforderliche DNS. AmazonProvided Wenn Sie mehr als einen Domain-Namenserver angeben, finden Sie weitere Informationen unter DHCP-Optionssätze im Amazon VPC-Benutzerhandbuch.

AWS ParallelCluster unterstützt die folgenden Konfigurationen auf hoher Ebene:

- Ein Subnetz für Kopf- und Rechenknoten.
- Zwei Subnetze, mit dem Hauptknoten in einem öffentlichen Subnetz und Rechenknoten in einem privaten Subnetz. Bei den Subnetzen kann es sich entweder um neue oder um bestehende Subnetze handeln.

Alle diese Konfigurationen können mit oder ohne öffentliche IP-Adressierung betrieben werden. AWS ParallelCluster kann auch so eingesetzt werden, dass ein HTTP-Proxy für alle AWS Anfragen verwendet wird. Die Kombinationen dieser Konfigurationen bedeuten, dass viele Bereitstellungsszenarien möglich sind. Sie können beispielsweise ein einzelnes öffentliches Subnetz mit vollständigem Zugriff über das Internet konfigurieren. Oder Sie können ein vollständig privates Netzwerk mithilfe AWS Direct Connect eines HTTP-Proxys für den gesamten Datenverkehr konfigurieren.

Illustrationen für einige dieser Szenarien finden Sie in den folgenden Architekturdiagrammen:

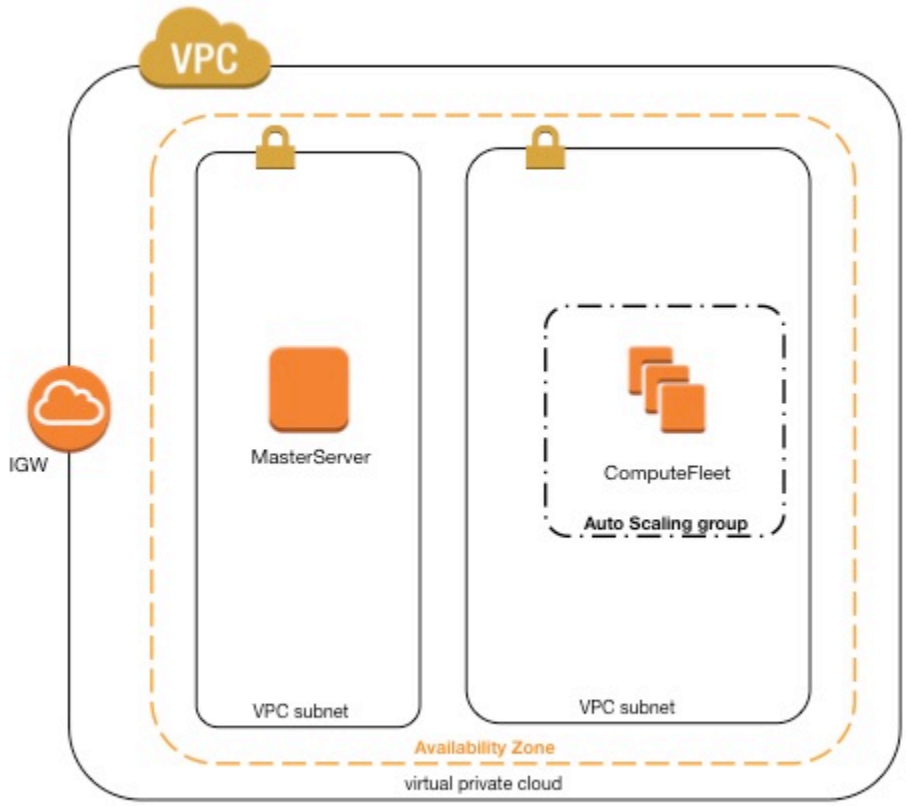
AWS ParallelCluster in einem einzigen öffentlichen Subnetz

Die Konfiguration für diese Architektur erfordert die folgenden Einstellungen:

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

Die `use_public_ips`-Einstellung kann nicht auf `false` gesetzt werden, da das Internet-Gateway voraussetzt, dass alle Instances über eine global eindeutige IP-Adresse verfügen. Weitere Informationen finden Sie unter Aktivieren des Internetzugangs im Amazon VPC-Benutzerhandbuch.

AWS ParallelCluster unter Verwendung von zwei Subnetzen



Die Konfiguration zum Erstellen eines neuen privaten Subnetzes für Datenverarbeitungs-Instances erfordert die folgenden Einstellungen:

Beachten Sie, dass alle Werte nur als Beispiele angegeben werden.

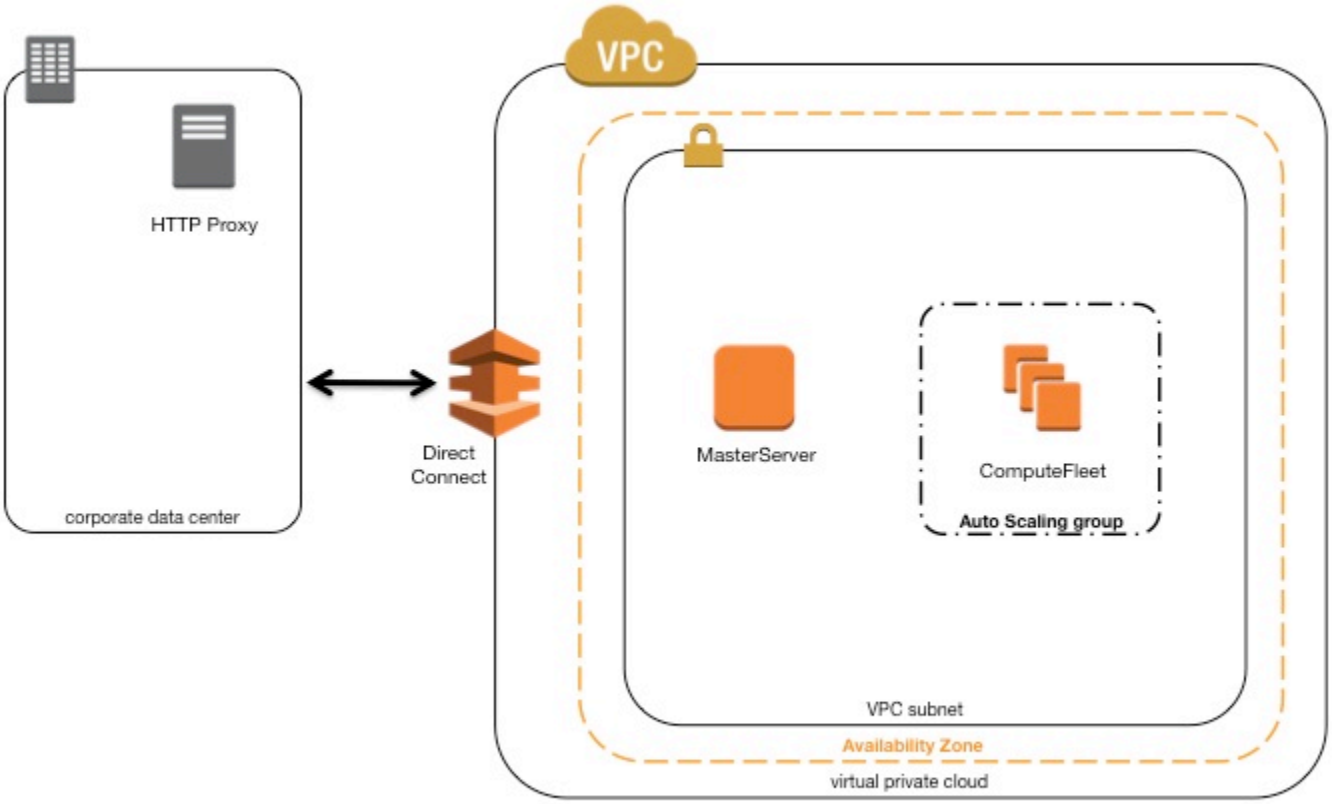
```
[vpc public-private-new]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-<public>
compute_subnet_cidr = 10.0.1.0/24
```

Die Konfiguration zur Verwendung eines vorhandenen privaten Netzwerks erfordert die folgenden Einstellungen:

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

Beide Konfigurationen erfordern ein NAT-Gateway oder einen internen Proxy, um den Webzugriff für Recheninstanzen zu ermöglichen.

AWS ParallelCluster in einem einzigen privaten Subnetz, verbunden mit AWS Direct Connect



Die Konfiguration für diese Architektur erfordert die folgenden Einstellungen:

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

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

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

Wenn `use_public_ips` auf `false` festgelegt ist, muss die VPC für die Verwendung des Proxy für den gesamten Datenverkehr korrekt eingerichtet werden. Webzugriff ist sowohl für Haupt- als auch für Rechenknoten erforderlich.

AWS ParallelCluster mit **`awsbatch`** Scheduler

Wenn Sie `awsbatch` als Scheduler-Typ verwenden, AWS ParallelCluster wird eine AWS Batch verwaltete Rechenumgebung erstellt. Die AWS Batch Umgebung kümmert sich um die Verwaltung von Amazon Elastic Container Service (Amazon ECS) Container-Instances, die in der gestartet werden`compute_subnet`. AWS Batch Damit Amazon ECS-Container-Instances ordnungsgemäß funktionieren, benötigen sie externen Netzwerkzugriff, um mit dem Amazon ECS-Serviceendpunkt zu kommunizieren. Daraus ergeben sich die folgenden Szenarien:

- Der `compute_subnet` verwendet ein NAT-Gateway für den Zugriff auf das Internet. (Wir empfehlen diesen Ansatz.)
- Im `compute_subnet` gestartete Instances verfügen über öffentliche IP-Adressen und können eine Internetverbindung über ein Internet-Gateway herstellen.

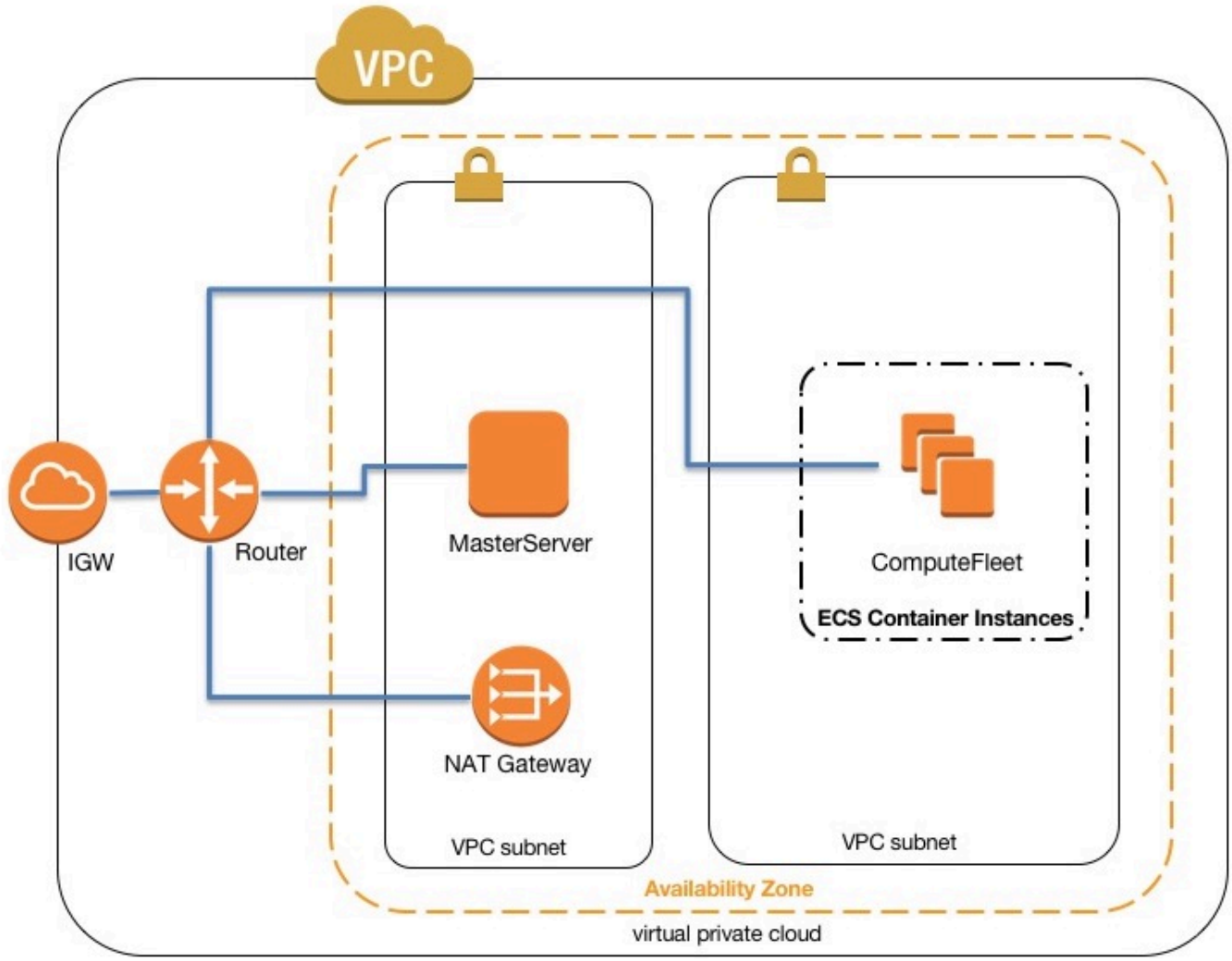
Wenn Sie außerdem an parallel Jobs mit mehreren Knoten interessiert sind (aus den AWS Batch Dokumenten):

AWS Batch parallel Jobs mit mehreren Knoten verwenden den Amazon `awsvpc` ECS-Netzwerkmodus, der Ihren parallel Job-Containern mit mehreren Knoten dieselben Netzwerkeigenschaften wie EC2 Amazon-Instances verleiht. Jeder Container eines parallelen Auftrags mit mehreren Knoten erhält seine eigene Elastic Network-Schnittstelle, eine primäre private IP-Adresse und einen internen DNS-Hostnamen. Die Netzwerkschnittstelle wird im selben Amazon VPC-Subnetz wie ihre Host-Rechenressource erstellt. Alle Sicherheitsgruppen, die auf Ihre Datenverarbeitungsressourcen angewendet werden, werden auch darauf angewendet.

Bei Verwendung von Amazon ECS Task Networking bietet der `awsvpc` Netzwerkmodus keine elastischen Netzwerkschnittstellen mit öffentlichen IP-Adressen für Aufgaben, die den EC2 Amazon-

Starttyp verwenden. Um auf das Internet zugreifen zu können, müssen Aufgaben, die den EC2 Amazon-Starttyp verwenden, in einem privaten Subnetz gestartet werden, das für die Verwendung eines NAT-Gateways konfiguriert ist.

Sie müssen ein NAT-Gateway konfigurieren, damit der Cluster parallel Jobs mit mehreren Knoten ausführen kann.



Weitere Informationen finden Sie unter den folgenden Themen:

- AWS Batch verwaltete Computerumgebungen
- AWS Batch parallel Jobs mit mehreren Knoten
- Amazon ECS Task Networking mit dem `awsvpc` Netzwerkmodus

Benutzerdefinierte Bootstrap-Aktionen

AWS ParallelCluster kann beliebigen Code entweder vor (vor der Installation) oder nach (nach der Installation) der Haupt-Bootstrap-Aktion ausführen, wenn der Cluster erstellt wird. In den meisten Fällen wird dieser Code in Amazon Simple Storage Service (Amazon S3) gespeichert und der Zugriff erfolgt über eine HTTPS-Verbindung. Der Code wird als Root ausgeführt und kann in jeder Skriptsprache vorliegen, die vom Cluster-Betriebssystem unterstützt wird. Oft ist der Code in Bash oder Python.

Aktionen vor der Installation werden aufgerufen, bevor eine Bootstrap-Aktion für die Clusterbereitstellung gestartet wird, z. B. die Konfiguration von NAT, Amazon Elastic Block Store (Amazon EBS) oder des Schedulers. Einige Aktionen vor der Installation umfassen das Ändern des Speichers, das Hinzufügen zusätzlicher Benutzer und das Hinzufügen von Paketen.

Aktionen nach der Installation werden aufgerufen, nachdem die Cluster-Bootstrap-Prozesse abgeschlossen sind. Aktionen nach der Installation sind die letzten Aktionen, die ausgeführt werden, bevor eine Instanz als vollständig konfiguriert und abgeschlossen betrachtet wird. Einige Aktionen nach der Installation umfassen das Ändern von Scheduler-Einstellungen, das Ändern des Speichers und das Ändern von Paketen.

Sie können Argumente an Skripts übergeben, indem Sie sie bei der Konfiguration angeben. Dazu übergeben Sie sie in doppelten Anführungszeichen an die Aktionen vor oder nach der Installation.

Wenn eine Aktion vor oder nach der Installation fehlschlägt, schlägt auch der Instanz-Bootstrap fehl. Der Erfolg wird mit einem Exit-Code von Null (0) signalisiert. Jeder andere Exit-Code weist darauf hin, dass der Instanz-Bootstrap fehlgeschlagen ist.

Sie können zwischen Running Head- und Compute-Knoten unterscheiden. Suchen Sie nach der `/etc/parallelcluster/cfnconfig` Datei und werten Sie die `cfn_node_type` Umgebungsvariablen aus, die jeweils den Wert `MasterServer` "" und `ComputeFleet` "" für den Kopf- und den Rechenknoten haben.

```
#!/bin/bash

. "/etc/parallelcluster/cfnconfig"

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

```
    ComputeFleet)
        echo "I am a compute node" >> /tmp/compute.txt
    ;;
    *)
    ;;
esac
```

Konfiguration

Die folgenden Konfigurationseinstellungen werden oft verwendet, Vor-/Nachinstallationsaktionen und Argumente zu definieren.

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

Argumente

Die ersten beiden Argumente (`$0` und `$1`) sind für den Skriptnamen und die URL reserviert.

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

Beispiel

Die folgenden Schritte erstellen ein einfaches Nachinstallationsskript, mit dem die R-Pakete in einem Cluster installiert werden.

1. Erstellen Sie ein Skript.

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

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

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

2. Laden Sie das Skript mit den richtigen Berechtigungen auf Amazon S3 hoch. Wenn öffentliche Leseberechtigungen für Sie nicht geeignet sind, verwenden Sie einen der `s3_read_write_resource` Parameter `s3_read_resource` oder, um den Zugriff zu gewähren. Weitere Informationen finden Sie unter Arbeiten mit Amazon S3.

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

> **Important**
>
> Wenn das Skript unter Windows bearbeitet wurde, müssen die Zeilenenden von CRLF in LF geändert werden, bevor das Skript auf Amazon S3 hochgeladen wird.

3. Aktualisieren Sie die AWS ParallelCluster Konfiguration, sodass sie die neue Aktion nach der Installation enthält.

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

Wenn der Bucket nicht öffentlich lesbar ist, verwenden Sie ihn `s3` als URL-Protokoll.

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

4. Starten Sie den Cluster.

```
$ pcluster create mycluster
```

5. Überprüfen Sie die Ausgabe.

```
$ less /var/log/cfn-init.log
2019-04-11 10:43:54,588 [DEBUG] Command runpostinstall output: post-install script
 has 4 arguments
arg: s3://bucket-name/test.sh
arg: R
arg: curl
arg: wget
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

Arbeiten mit Amazon S3

Um Cluster-Ressourcen Zugriff auf Amazon S3 S3-Buckets zu gewähren, geben Sie den Bucket ARNs in den `s3_read_write_resource` Parametern `s3_read_resource` und in der AWS ParallelCluster Konfiguration an. Weitere Informationen zur Steuerung des Zugriffs mit finden Sie AWS ParallelCluster unterAWS Identity and Access Management Rollen in AWS ParallelCluster.

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

Beide Parameter akzeptieren entweder einen `*` oder einen gültigen Amazon S3 S3-ARN. Informationen zur Angabe von Amazon S3 ARNs finden Sie im Amazon S3 S3-ARN-Format in der Allgemeine AWS-Referenz.

Beispiele

Das folgende Beispiel gibt Ihnen Lesezugriff auf jedes Objekt im Amazon S3 S3-Bucket my_corporate_bucket.

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

Das folgende Beispiel bietet Ihnen Lesezugriff auf den Bucket, aber nicht auf Elemente aus dem Bucket.

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

Dieses letzte Beispiel gibt Ihnen Lesezugriff auf den Bucket und auf die im Bucket gespeicherten Elemente.

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

Arbeiten mit Spot-Instances

AWS ParallelCluster verwendet Spot-Instances, wenn in der Cluster-Konfiguration auf `cluster_type` = Spot gesetzt wurde. Spot-Instances sind kostengünstiger als On-Demand-Instances, sie können jedoch unterbrochen werden. Die Auswirkung der Unterbrechung ist je nach verwendetem Scheduler unterschiedlich. Es kann hilfreich sein, die Unterbrechungsbenachrichtigungen für Spot-Instances zu nutzen, die eine zweiminütige Warnung enthalten, bevor Amazon Ihre Spot-Instance stoppen oder beenden EC2 muss. Weitere Informationen finden Sie unter Spot-Instance-Unterbrechungen im EC2 Amazon-Benutzerhandbuch. In den folgenden Abschnitten werden drei Szenarien beschrieben, in denen Spot-Instances unterbrochen werden können.

> **Note**
>
> Die Verwendung von Spot-Instances setzt voraus, dass die mit dem `AWSServiceRoleForEC2Spot` Service verknüpfte Rolle in Ihrem Konto vorhanden ist. Führen Sie den folgenden Befehl aus AWS CLI, um diese Rolle in Ihrem Konto mithilfe von zu erstellen:
>
> ```
> aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
> ```
>
> Weitere Informationen finden Sie unter Service-verknüpfte Rolle für Spot-Instance-Anfragen im EC2 Amazon-Benutzerhandbuch.

Szenario 1: Spot-Instance ohne ausgeführte Aufgaben wird unterbrochen

Wenn diese Unterbrechung auftritt, wird AWS ParallelCluster versucht, die Instance zu ersetzen, falls die Scheduler-Warteschlange ausstehende Jobs enthält, für die zusätzliche Instances erforderlich sind, oder wenn die Anzahl der aktiven Instances unter der `initial_queue_size` Einstellung liegt. Wenn keine neuen Instanzen bereitgestellt werden AWS ParallelCluster können, wird eine Anfrage für neue Instanzen regelmäßig wiederholt.

Szenario 2: Spot-Instance mit Einzelknotenaufgaben wird unterbrochen

Das Verhalten dieser Unterbrechung hängt vom verwendeten Scheduler ab.

Slurm

Der Job schlägt mit dem Statuscode von `NODE_FAIL` fehl und der Job wird in eine Warteschlange gestellt (sofern dies nicht `--no-requeue` beim Absenden des Jobs angegeben wurde). Wenn es sich bei dem Knoten um einen statischen Knoten handelt, wird er ersetzt. Wenn der Knoten ein dynamischer Knoten ist, wird der Knoten beendet und zurückgesetzt. Weitere Informationen über `sbatch` und einschließlich des `--no-requeue` Parameters finden Sie sbatchin der Slurm-Dokumentation.

> **Note**
>
> Dieses Verhalten hat sich in AWS ParallelCluster Version 2.9.0 geändert. Frühere Versionen haben den Job mit dem Statuscode von beendet `NODE_FAIL` und der Knoten wurde aus der Scheduler-Warteschlange entfernt.

SGE

> **Note**
>
> Dies gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster -Schedulern nicht unterstützt. SGE Torque

Die Aufgabe wird beendet. Wenn für die Aufgabe die Markierung für erneute Ausführung aktiviert ist, (entweder mit `qsub -r yes` oder `qalter -r yes`) oder bei der Warteschlange die `rerun-`

Konfiguration auf „TRUE“ festgelegt ist, wird die Aufgabe neu geplant. Die Datenverarbeitungs-Instance wird aus der Scheduler-Warteschlange entfernt. Dieses Verhalten stammt von den folgenden SGE-Konfigurationsparametern:

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

Torque

> **Note**
>
> Dies gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster -Schedulern nicht unterstützt. SGE Torque

Die Aufgabe wird aus dem System und der Knoten aus dem Scheduler entfernt. Der Job wird nicht erneut ausgeführt. Wenn mehrere Jobs auf der Instanz ausgeführt werden, während sie unterbrochen wird, kann es bei Torque beim Entfernen des Knotens zu einem Timeout kommen. In der `sqswatcher` Protokolldatei wird möglicherweise ein Fehler angezeigt. Dies hat keinen Einfluss auf die Skalierungslogik, und bei nachfolgenden Wiederholungsversuchen wird eine ordnungsgemäße Bereinigung durchgeführt.

Szenario 3: Spot-Instance, auf der Aufgaben mit mehreren Knoten ausgeführt werden, wird unterbrochen

Das Verhalten dieser Unterbrechung hängt vom verwendeten Scheduler ab.

Slurm

Der Job schlägt mit dem Statuscode von fehl`NODE_FAIL`, und der Job wird in die Warteschlange gestellt (es sei denn, dies `--no-requeue` wurde bei der Übermittlung des Jobs angegeben). Wenn es sich bei dem Knoten um einen statischen Knoten handelt, wird er ersetzt. Wenn der Knoten ein dynamischer Knoten ist, wird der Knoten beendet und zurückgesetzt. Andere Knoten, auf denen die beendeten Jobs ausgeführt wurden, wurden möglicherweise anderen ausstehenden Jobs zugewiesen oder nach Ablauf der konfigurierten `scaledown_idletime` Zeit herunterskaliert.

> **Note**
>
> Dieses Verhalten hat sich in AWS ParallelCluster Version 2.9.0 geändert. Frühere Versionen haben den Job mit dem Statuscode von beendet `NODE_FAIL` und der Knoten wurde aus der Scheduler-Warteschlange entfernt. Andere Knoten, auf denen die beendeten Jobs ausgeführt wurden, wurden möglicherweise nach Ablauf der konfigurierten `scaledown_idletime` Zeit herunterskaliert.

SGE

> **Note**
>
> Dies gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster -Schedulern nicht unterstützt. SGE Torque

Der Job wurde nicht beendet und läuft weiterhin auf den verbleibenden Knoten. Der Datenverarbeitungsknoten wird aus der Scheduler-Warteschlange entfernt, aber in der Hosts-Liste als verwaister und nicht verfügbarer Knoten angezeigt.

Der Benutzer muss die Aufgabe löschen, wenn dies geschieht (`qdel <jobid>`). Der Knoten wird weiterhin in der Hostliste (`qhost`) angezeigt, obwohl dies keine Auswirkungen hat AWS ParallelCluster. Um den Host aus der Liste zu entfernen, führen Sie nach dem Ersetzen der Instanz den folgenden Befehl aus.

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

Torque

> **Note**
>
> Dies gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster -Schedulern nicht unterstützt. SGE Torque

Die Aufgabe wird aus dem System und der Knoten aus dem Scheduler entfernt. Der Job wird nicht erneut ausgeführt. Wenn mehrere Jobs auf der Instanz ausgeführt werden, während sie unterbrochen wird, kann es bei Torque beim Entfernen des Knotens zu einem Timeout kommen. In der `sqswatcher` Protokolldatei wird möglicherweise ein Fehler angezeigt. Dies hat keinen Einfluss auf die Skalierungslogik, und bei nachfolgenden Wiederholungsversuchen wird eine ordnungsgemäße Bereinigung durchgeführt.

Weitere Informationen zu Spot-Instances finden Sie unter Spot-Instances im EC2 Amazon-Benutzerhandbuch.

AWS Identity and Access Management Rollen in AWS ParallelCluster

AWS ParallelCluster verwendet AWS Identity and Access Management (IAM-) Rollen für Amazon EC2 , um Instances den Zugriff auf AWS Dienste für die Bereitstellung und den Betrieb eines Clusters zu ermöglichen. Standardmäßig wird die IAM-Rolle für Amazon EC2 erstellt, wenn der Cluster erstellt wird. Das bedeutet, dass der Benutzer, der den Cluster erstellt, die entsprechenden Berechtigungen einrichten muss, wie in den folgenden Abschnitten beschrieben.

AWS ParallelCluster verwendet mehrere AWS Dienste, um einen Cluster bereitzustellen und zu betreiben. Die vollständige Liste finden Sie im AWS ParallelCluster Abschnitt Verwendete AWS Dienste.

Sie können die Änderungen an den Beispielrichtlinien in der AWS ParallelCluster Dokumentation unter nachverfolgen GitHub.

Themen

- Standardeinstellungen für die Clustererstellung
- Verwenden einer vorhandenen IAM-Rolle für Amazon EC2
- AWS ParallelCluster Beispiel für Instanz- und Benutzerrichtlinien

Standardeinstellungen für die Clustererstellung

Wenn Sie die Standardeinstellungen für die Clustererstellung verwenden, EC2 wird vom Cluster eine Standard-IAM-Rolle für Amazon erstellt. Der Benutzer, der den Cluster erstellt, muss über die richtigen Berechtigungen verfügen, um alle Ressourcen zu erstellen, die zum Starten des Clusters

erforderlich sind. Dazu gehört die Erstellung einer IAM-Rolle für Amazon EC2. In der Regel muss der Benutzer über die Berechtigungen einer AdministratorAccessverwalteten Richtlinie verfügen, wenn er die Standardeinstellungen verwendet. Informationen zu verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

Verwenden einer vorhandenen IAM-Rolle für Amazon EC2

Anstelle der Standardeinstellungen können Sie `ec2_iam_role`beim Erstellen eines Clusters eine vorhandene verwenden. Sie müssen jedoch die IAM-Richtlinie und -Rolle definieren, bevor Sie versuchen, den Cluster zu starten. In der Regel wählen Sie eine bestehende IAM-Rolle für Amazon EC2 , um die Berechtigungen zu minimieren, die Benutzern beim Starten von Clustern gewährt werden. AWS ParallelCluster Beispiel für Instanz- und BenutzerrichtlinienDazu gehören die Mindestberechtigungen, die von erforderlich sind, AWS ParallelCluster und ihre Funktionen. Sie müssen sowohl Richtlinien als auch Rollen als individuelle Richtlinien in IAM erstellen und die Rollen und Richtlinien dann den entsprechenden Ressourcen zuordnen. Einige der Rollenrichtlinien können umfangreich werden und zu Kontingentfehlern führen. Weitere Informationen finden Sie unter Behebung von Problemen mit der Größe der IAM-Richtlinie. Ersetzen Sie in den Richtlinien die Zeichenfolgen `<REGION><AWS ACCOUNT ID>`, und ähnliche Zeichenfolgen durch die entsprechenden Werte.

Wenn Sie beabsichtigen, den Standardeinstellungen für Clusterknoten zusätzliche Richtlinien hinzuzufügen, empfehlen wir Ihnen, die zusätzlichen benutzerdefinierten IAM-Richtlinien zusammen mit der `additional_iam_policies`Einstellung zu übergeben, anstatt die `ec2_iam_role`Einstellungen zu verwenden.

AWS ParallelCluster Beispiel für Instanz- und Benutzerrichtlinien

Die folgenden Beispielrichtlinien beinhalten Amazon Resource Names (ARNs) für die Ressourcen. Wenn Sie in den Partitionen AWS GovCloud (US) oder AWS China arbeiten, ARNs muss die geändert werden. Insbesondere müssen sie von „arn:aws“ auf „arn:aws-us-gov“ für die AWS GovCloud (US) Partition oder „arn:aws-cn“ für die China-Partition geändert werden. AWS Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in AWS GovCloud (US) Regionen im AWS GovCloud (US) Benutzerhandbuch und ARNs für AWS Dienste in China unter Erste Schritte mit AWS Diensten in China.

Diese Richtlinien beinhalten die derzeit erforderlichen Mindestberechtigungen AWS ParallelCluster, Funktionen und Ressourcen. Einige der Rollenrichtlinien können umfangreich werden und zu Kontingentfehlern führen. Weitere Informationen finden Sie unter Behebung von Problemen mit der Größe der IAM-Richtlinie.

Themen

- ParallelClusterInstancePolicymit SGESlurm, oder Torque
- ParallelClusterInstancePolicy mit awsbatch
- ParallelClusterUserPolicy mit Slurm
- ParallelClusterUserPolicymit oder SGETorque
- ParallelClusterUserPolicy mit awsbatch
- ParallelClusterLambdaPolicymit, oder SGESlurmTorque
- ParallelClusterLambdaPolicy mit awsbatch
- ParallelClusterUserPolicyfür Benutzer

`ParallelClusterInstancePolicy`mit SGESlurm, oder Torque

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von SGE Torque OR-Schedulern AWS ParallelCluster nicht unterstützt. Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.

Themen

- ParallelClusterInstancePolicy mit Slurm
- ParallelClusterInstancePolicymit oder SGETorque

`ParallelClusterInstancePolicy` mit Slurm

Im folgenden Beispiel wird die `ParallelClusterInstancePolicy` Verwendung Slurm als Scheduler festgelegt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:AttachVolume",
```

```
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstances",
                "ec2:DescribeRegions",
                "ec2:TerminateInstances",
                "ec2:DescribeLaunchTemplates",
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "EC2"
        },
        {
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:subnet/<COMPUTE SUBNET ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:network-interface/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*",
                "arn:aws:ec2:<REGION>::image/<IMAGE ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:key-pair/<KEY NAME>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:security-group/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:launch-template/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:placement-group/*"
            ],
            "Effect": "Allow",
            "Sid": "EC2RunInstances"
        },
        {
            "Action": [
                "dynamodb:ListTables"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBList"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
```

```
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
            ],
            "Resource": [
                "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:GetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:DeleteItem",
                "dynamodb:DescribeTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBTable"
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
            ],
            "Effect": "Allow",
            "Sid": "S3GetObj"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "IAMPassRole",
```

```
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::dcv-license.<REGION>/*"
            ],
            "Effect": "Allow",
            "Sid": "DcvLicense"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "GetClusterConfig"
        },
        {
            "Action": [
                "fsx:DescribeFileSystems"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
```

```
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "CWLogs"
        },
        {
            "Action": [
                "route53:ChangeResourceRecordSets"
            ],
            "Resource": [
                "arn:aws:route53:::hostedzone/*"
            ],
            "Effect": "Allow",
            "Sid": "Route53"
        }
    ]
}
```

`ParallelClusterInstancePolicy`mit oder SGETorque

Im folgenden Beispiel wird der Befehl `ParallelClusterInstancePolicy` using SGE or Torque als Scheduler festgelegt.

> **Note**
>
> Diese Richtlinie gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster -Schedulern nicht unterstützt. SGE Torque

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:AttachVolume",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstances",
```

```
                "ec2:DescribeRegions",
                "ec2:TerminateInstances",
                "ec2:DescribeLaunchTemplates",
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "EC2"
        },
        {
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:subnet/<COMPUTE SUBNET ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:network-interface/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*",
                "arn:aws:ec2:<REGION>::image/<IMAGE ID>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:key-pair/<KEY NAME>",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:security-group/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:launch-template/*",
                "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:placement-group/*"
            ],
            "Effect": "Allow",
            "Sid": "EC2RunInstances"
        },
        {
            "Action": [
                "dynamodb:ListTables"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBList"
        },
        {
            "Action": [
                "sqs:SendMessage",
                "sqs:ReceiveMessage",
                "sqs:ChangeMessageVisibility",
                "sqs:DeleteMessage",
                "sqs:GetQueueUrl"
```

```
            ],
            "Resource": [
                "arn:aws:sqs:<REGION>:<AWS ACCOUNT ID>:parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "SQSQueue"
        },
        {
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DescribeTags",
                "autoscaling:SetInstanceHealth"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "Autoscaling"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
            ],
            "Resource": [
                "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/
parallelcluster-*/*"
            ],
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:GetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:DeleteItem",
                "dynamodb:DescribeTable"
            ],
```

```
            "Resource": [
                "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "DynamoDBTable"
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
            ],
            "Effect": "Allow",
            "Sid": "S3GetObj"
        },
        {
            "Action": [
                "sqs:ListQueues"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "SQSList"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "IAMPassRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        },
        {
```

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

```
            ],
            "Resource": [
                "arn:aws:route53:::hostedzone/*"
            ],
            "Effect": "Allow",
            "Sid": "Route53"
        }
    ]
}
```

`ParallelClusterInstancePolicy` mit `awsbatch`

Im folgenden Beispiel wird die `ParallelClusterInstancePolicy` Verwendung `awsbatch` als Scheduler festgelegt. Sie müssen dieselben Richtlinien einbeziehen, die denen zugewiesen sind, `BatchUserRole` die im AWS Batch AWS CloudFormation verschachtelten Stack definiert sind. Der `BatchUserRole`-ARN wird als Stack-Ausgabe bereitgestellt. In diesem Beispiel ist „*<RESOURCES S3 BUCKET>*“ der Wert der `cluster_resource_bucket` Einstellung; wenn nicht angegeben, `cluster_resource_bucket` ist „*<RESOURCES S3 BUCKET>*“ „parallelcluster-*“. Das folgende Beispiel gibt einen Überblick über die erforderlichen Berechtigungen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "batch:RegisterJobDefinition",
                "logs:GetLogEvents"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "batch:SubmitJob",
                "cloudformation:DescribeStacks",
                "ecs:ListContainerInstances",
                "ecs:DescribeContainerInstances",
                "logs:FilterLogEvents",
                "s3:PutObject",
                "s3:Get*",
```

```
                "s3:DeleteObject",
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-
definition/<AWS_BATCH_STACK - JOB_DEFINITION_SERIAL_NAME>:1",
                "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-
definition/<AWS_BATCH_STACK - JOB_DEFINITION_MNP_NAME>*",
                "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/<AWS_BATCH_STACK -
 JOB_QUEUE_NAME>",
                "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/<STACK NAME>/
*",
                "arn:aws:s3:::<RESOURCES S3 BUCKET>/batch/*",
                "arn:aws:iam::<AWS ACCOUNT ID>:role/<AWS_BATCH_STACK - JOB_ROLE>",
                "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/<ECS COMPUTE
 ENVIRONMENT>",
                "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/*",
                "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-
stream:*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "batch:DescribeJobQueues",
                "batch:TerminateJob",
                "batch:DescribeJobs",
                "batch:CancelJob",
                "batch:DescribeJobDefinitions",
                "batch:ListJobs",
                "batch:DescribeComputeEnvironments"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
```

```
        {
            "Action": [
                "ec2:DescribeInstances",
                "ec2:AttachVolume",
                "ec2:DescribeVolumes",
                "ec2:DescribeInstanceAttribute"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2"
        },
        {
            "Action": [
                "cloudformation:DescribeStackResource",
                "cloudformation:SignalResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudFormation"
        },
        {
            "Action": [
                "fsx:DescribeFileSystems"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:TagResource",
                "logs:UntagResource",
                "logs:CreateLogStream"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "CWLogs"
        }
    ]
```

```
}
```

`ParallelClusterUserPolicy` mit Slurm

Im folgenden Beispiel wird die `ParallelClusterUserPolicy` mit Slurm als Scheduler festgelegt. In diesem Beispiel ist „*<RESOURCES S3 BUCKET>*“ der Wert der `cluster_resource_bucket` Einstellung; wenn nicht angegeben, `cluster_resource_bucket` ist „*<RESOURCES S3 BUCKET>*“ „parallelcluster-*“.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Rolle verwenden `ec2_iam_role = <role_name>`, müssen Sie die IAM-Ressource so ändern, dass sie den Namen dieser Rolle wie folgt enthält:
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"`
> auf:
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/<role_name>"`

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

```
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Describe"
        },
        {
            "Action": [
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DescribeNatGateways",
                "ec2:CreateNatGateway",
                "ec2:DescribeInternetGateways",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DescribeRouteTables",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:CreateSubnet",
                "ec2:ModifySubnetAttribute"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "NetworkingEasyConfig"
        },
        {
            "Action": [
                "ec2:CreateVolume",
                "ec2:RunInstances",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AttachNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:ModifyVolumeAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteVolume",
                "ec2:TerminateInstances",
                "ec2:DeleteSecurityGroup",
                "ec2:DisassociateAddress",
                "ec2:RevokeSecurityGroupIngress",
```

```
                "ec2:RevokeSecurityGroupEgress",
                "ec2:ReleaseAddress",
                "ec2:CreatePlacementGroup",
                "ec2:DeletePlacementGroup"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Modify"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "ScalingModify"
        },
        {
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:ListTagsOfResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBDescribe"
        },
        {
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:TagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBModify"
```

```
        },
        {
            "Action": [
                "route53:ChangeResourceRecordSets",
                "route53:ChangeTagsForResource",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetChange",
                "route53:GetHostedZone",
                "route53:ListResourceRecordSets",
                "route53:ListQueryLoggingConfigs"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Route53HostedZones"
        },
        {
            "Action": [
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:ListStacks",
                "cloudformation:GetTemplate"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudFormationDescribe"
        },
        {
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:UpdateStack"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CloudFormationModify"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3ResourcesBucket"
        },
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::<REGION>-aws-parallelcluster*"
            ],
            "Effect": "Allow",
            "Sid": "S3ParallelClusterReadOnly"
        },
        {
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3Delete"
        },
        {
            "Action": [
                "iam:PassRole",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
                "iam:TagRole",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": [
                "arn:aws:iam::<AWS ACCOUNT ID>:role/<PARALLELCLUSTER EC2 ROLE NAME>",
                "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "IAMModify"
        },
```

```
        {
          "Condition": {
              "StringEquals": {
                  "iam:AWSServiceName": [
                      "fsx.amazonaws.com",
                      "s3.data-source.lustre.fsx.amazonaws.com"
                  ]
              }
          },
          "Action": [
              "iam:CreateServiceLinkedRole"
          ],
          "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/*",
          "Effect": "Allow",
          "Sid": "IAMServiceLinkedRole"
        },
        {
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile"
            ],
            "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
            "Effect": "Allow",
            "Sid": "IAMCreateInstanceProfile"
        },
        {
            "Action": [
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:GetRolePolicy",
                "iam:GetPolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
                "iam:PutRolePolicy",
                "iam:DeleteRolePolicy"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "IAMInstanceProfile"
        },
        {
            "Action": [
                "elasticfilesystem:DescribeMountTargets",
                "elasticfilesystem:DescribeMountTargetSecurityGroups",
```

```
                "ec2:DescribeNetworkInterfaceAttribute"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFSDescribe"
        },
        {
            "Action": [
                "ssm:GetParametersByPath"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SSMDescribe"
        },
        {
            "Action": [
                "fsx:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Action": [
                "elasticfilesystem:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFS"
        },
        {
            "Action": [
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup",
                "logs:TagResource",
                "logs:UntagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudWatchLogs"
        },
        {
```

```
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:AddPermission",
                "lambda:RemovePermission",
                "lambda:TagResource",
                "lambda:ListTags",
                "lambda:UntagResource"
            ],
            "Resource": [
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "Lambda"
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
              "cloudwatch:PutDashboard",
              "cloudwatch:ListDashboards",
              "cloudwatch:DeleteDashboards",
              "cloudwatch:GetDashboard"
            ],
            "Resource": "*"
        }
    ]
}
```

`ParallelClusterUserPolicy`mit oder SGETorque

> **Note**
>
> Dieser Abschnitt gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster -Schedulern nicht unterstützt. SGE Torque

Im folgenden Beispiel wird der`ParallelClusterUserPolicy`, mit SGE oder Torque als Scheduler festgelegt. In diesem Beispiel ist „*<RESOURCES S3 BUCKET>*“ der Wert der `cluster_resource_bucket` Einstellung; wenn nicht angegeben, `cluster_resource_bucket` ist „*<RESOURCES S3 BUCKET>*“ „parallelcluster-*“.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Rolle verwenden `ec2_iam_role = <role_name>`, müssen Sie die IAM-Ressource so ändern, dass sie den Namen dieser Rolle wie folgt enthält:
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster-*"`
> auf:
> `"Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/<role_name>"`

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

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

```
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Modify"
        },
        {
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AutoScalingDescribe"
        },
        {
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:PutScalingPolicy",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeletePolicy",
                "autoscaling:DisableMetricsCollection",
                "autoscaling:EnableMetricsCollection"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "AutoScalingModify"
        },
        {
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:ListTagsOfResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBDescribe"
```

```
        },
        {
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:TagResource"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "DynamoDBModify"
        },
        {
            "Action": [
                "sqs:GetQueueAttributes"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SQSDescribe"
        },
        {
            "Action": [
                "sqs:CreateQueue",
                "sqs:SetQueueAttributes",
                "sqs:DeleteQueue",
                "sqs:TagQueue"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SQSModify"
        },
        {
            "Action": [
                "sns:ListTopics",
                "sns:GetTopicAttributes"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SNSDescribe"
        },
        {
            "Action": [
```

```
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sns:DeleteTopic"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "SNSModify"
        },
        {
            "Action": [
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
                "cloudformation:DescribeStacks",
                "cloudformation:ListStacks",
                "cloudformation:GetTemplate"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "CloudFormationDescribe"
        },
        {
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:UpdateStack"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "CloudFormationModify"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3ResourcesBucket"
        },
        {
            "Action": [
```

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

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

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

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

`ParallelClusterUserPolicy` mit `awsbatch`

Im folgenden Beispiel wird die `ParallelClusterUserPolicy` Verwendung `awsbatch` als Scheduler festgelegt. In diesem Beispiel ist „*<RESOURCES S3 BUCKET>*“ der Wert der `cluster_resource_bucket` Einstellung; wenn nicht angegeben, `cluster_resource_bucket` ist „*<RESOURCES S3 BUCKET>*“ „parallelcluster-*“.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeKeyPairs",
                "ec2:DescribeRegions",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
```

```
                "ec2:DescribePlacementGroups",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeSnapshots",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeAddresses",
                "ec2:CreateTags",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Describe"
        },
        {
            "Action": [
                "ec2:CreateLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:ModifyLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2LaunchTemplate"
        },
        {
            "Action": [
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DescribeNatGateways",
                "ec2:CreateNatGateway",
                "ec2:DescribeInternetGateways",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DescribeRouteTables",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
```

```
                "ec2:CreateSubnet",
                "ec2:ModifySubnetAttribute"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "NetworkingEasyConfig"
        },
        {
            "Action": [
                "ec2:CreateVolume",
                "ec2:RunInstances",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AttachNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateSecurityGroup",
                "ec2:ModifyVolumeAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteVolume",
                "ec2:TerminateInstances",
                "ec2:DeleteSecurityGroup",
                "ec2:DisassociateAddress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:ReleaseAddress",
                "ec2:CreatePlacementGroup",
                "ec2:DeletePlacementGroup"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EC2Modify"
        },
        {
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:GetItem",
                "dynamodb:PutItem",
                "dynamodb:Query",
                "dynamodb:TagResource"
```

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

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

```
            "Sid": "IAMRole"
        },
        {
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*",
            "Effect": "Allow",
            "Sid": "IAMInstanceProfile"
        },
        {
            "Action": [
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:GetRolePolicy",
                "iam:PutRolePolicy",
                "iam:DeleteRolePolicy",
                "iam:GetPolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "IAM"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3ResourcesBucket"
        },
        {
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
            ],
            "Effect": "Allow",
            "Sid": "S3ParallelClusterReadOnly"
        },
        {
            "Action": [
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::<RESOURCES S3 BUCKET>"
            ],
            "Effect": "Allow",
            "Sid": "S3Delete"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:InvokeFunction",
                "lambda:AddPermission",
                "lambda:RemovePermission",
                "lambda:TagResource",
                "lambda:ListTags",
                "lambda:UntagResource"
            ],
            "Resource": [
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:parallelcluster-*",
                "arn:aws:lambda:<REGION>:<AWS ACCOUNT ID>:function:pcluster-*"
            ],
            "Effect": "Allow",
            "Sid": "Lambda"
        },
        {
            "Action": [
                "logs:*"
            ],
            "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:*",
            "Effect": "Allow",
            "Sid": "Logs"
```

```
        },
        {
            "Action": [
                "codebuild:*"
            ],
            "Resource": "arn:aws:codebuild:<REGION>:<AWS ACCOUNT ID>:project/
parallelcluster-*",
            "Effect": "Allow",
            "Sid": "CodeBuild"
        },
        {
            "Action": [
                "ecr:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "ECR"
        },
        {
            "Action": [
                "batch:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Batch"
        },
        {
            "Action": [
                "events:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "AmazonCloudWatchEvents"
        },
        {
            "Action": [
                "ecs:DescribeContainerInstances",
                "ecs:ListContainerInstances"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "ECS"
        },
        {
```

```
            "Action": [
                "elasticfilesystem:CreateFileSystem",
                "elasticfilesystem:CreateMountTarget",
                "elasticfilesystem:DeleteFileSystem",
                "elasticfilesystem:DeleteMountTarget",
                "elasticfilesystem:DescribeFileSystems",
                "elasticfilesystem:DescribeMountTargets"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "EFS"
        },
        {
            "Action": [
                "fsx:*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "FSx"
        },
        {
            "Sid": "CloudWatch",
            "Effect": "Allow",
            "Action": [
              "cloudwatch:PutDashboard",
              "cloudwatch:ListDashboards",
              "cloudwatch:DeleteDashboards",
              "cloudwatch:GetDashboard"
            ],
            "Resource": "*"
        }
    ]
}
```

`ParallelClusterLambdaPolicy`mit, oder SGESlurmTorque

Im folgenden Beispiel wird die `ParallelClusterLambdaPolicy` mit SGE, Slurm oder Torque als Scheduler festgelegt.

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von SGE Torque OR-Schedulern AWS ParallelCluster nicht unterstützt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*",
      "Effect": "Allow",
      "Sid": "CloudWatchLogsPolicy"
    },
    {
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Effect": "Allow",
      "Sid": "S3BucketPolicy"
    },
    {
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "DescribeInstances"
    },
    {
      "Action": [
```

```
        "ec2:TerminateInstances"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "FleetTerminatePolicy"
    },
    {
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem"
      ],
      "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
      "Effect": "Allow",
      "Sid": "DynamoDBTable"
    },
    {
      "Action": [
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": [
        "arn:aws:route53:::hostedzone/*"
      ],
      "Effect": "Allow",
      "Sid": "Route53DeletePolicy"
    }
  ]
}
```

`ParallelClusterLambdaPolicy` mit `awsbatch`

Im folgenden Beispiel wird die `ParallelClusterLambdaPolicy` Verwendung `awsbatch` als Scheduler festgelegt.

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

```
      "Resource": "arn:aws:logs:*:*:*",
      "Sid": "CloudWatchLogsPolicy"
    },
    {
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:ListImages"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "ECRPolicy"
    },
    {
      "Action": [
        "codebuild:BatchGetBuilds",
        "codebuild:StartBuild"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "CodeBuildPolicy"
    },
    {
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Sid": "S3BucketPolicy"
    }
  ]
}
```

`ParallelClusterUserPolicy`für Benutzer

Im folgenden Beispiel wird der `ParallelClusterUserPolicy` Wert für Benutzer festgelegt, die keine Cluster erstellen oder aktualisieren müssen. Die folgenden Befehle werden unterstützt.

- `pcluster dcv`
- `pcluster instances`

- pcluster list
- pcluster ssh
- pcluster start
- pcluster status
- pcluster stop
- pcluster version

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
             "Sid": "MinimumModify",
             "Action": [
                "autoscaling:UpdateAutoScalingGroup",
                "batch:UpdateComputeEnvironment",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResources",
                "cloudformation:GetTemplate",
                "dynamodb:GetItem",
                "dynamodb:PutItem"
             ],
             "Effect": "Allow",
             "Resource": [
                 "arn:aws:autoscaling:<REGION>:<AWS ACCOUNT
 ID>:autoScalingGroup:*:autoScalingGroupName/parallelcluster-*",
                 "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:compute-environment/*",
                 "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/<CLUSTERNAME>/
*",
                 "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/<CLUSTERNAME>"
             ]
        },
        {
           "Sid": "Describe",
           "Action": [
                "cloudformation:DescribeStacks",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus"
           ],
           "Effect": "Allow",
           "Resource": "*"
```

```
        }
    ]
}
```

Scheduler werden unterstützt von AWS ParallelCluster

AWS ParallelCluster unterstützt mehrere Scheduler, die mithilfe der `scheduler` Einstellung eingestellt werden.

> **Note**
>
> Beginnend mit Version 2.11.5 unterstützt die Verwendung von SGE OR-Schedulern AWS ParallelCluster nicht. Torque Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.

Themen

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

Son of Grid Engine (`sge`)

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von oder AWS ParallelCluster Schedulern nicht unterstützt. SGE Torque Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.

AWS ParallelCluster Versionen 2.11.4 und früher verwenden 8.1.9. Son of Grid Engine

Slurm Workload Manager (`slurm`)

AWS ParallelCluster Version 2.11.9 verwendet 20.11.9. Slurm Weitere Informationen zu Slurm finden Sie unter https://slurm.schedmd.com/. Downloads finden Sie unter https://github.com/SchedMD/slurm/tags. Sie finden den Quellcode unter https://github.com/SchedMD/slurm.

> **Important**
>
> AWS ParallelCluster wird mit Slurm Konfigurationsparametern getestet, die standardmäßig bereitgestellt werden. Alle Änderungen, die Sie an diesen Slurm Konfigurationsparametern vornehmen, erfolgen auf eigenes Risiko. Sie werden nur nach bestem Wissen und Gewissen unterstützt.

AWS ParallelCluster Version (en)	Unterstützte Slurm-Version
2.11.7, 2.11.8, 2.11.9	20,11,9
2.11.4 bis 2.11.6	20.11.8
2.11.0 bis 2.11.3	20.11.7
2.10.4	20,02,7
2.9.0 bis 2.10.3	20.02.4
2.6 bis 2.8.1	19.05.5
2,5.0, 2,5.1	19.05.3-2
2.3.1 bis 2.4.1	18.08.6-2
vor 2.3.1	16.05.3-1

Modus mit mehreren Warteschlangen

AWS ParallelCluster Version 2.9.0 führte den Modus mit mehreren Warteschlangen ein. Der Modus mit mehreren Warteschlangen wird unterstützt, wenn `scheduler` er auf eingestellt `slurm` und die `queue_settings` Einstellung definiert ist. Dieser Modus ermöglicht die Koexistenz verschiedener

Instanztypen in den Rechenknoten. Die Rechenressourcen, die die verschiedenen Instanztypen enthalten, können nach Bedarf nach oben oder unten skaliert werden. Im Warteschlangenmodus werden bis zu fünf (5) Warteschlangen unterstützt, und jeder `[queue]`Abschnitt kann sich auf bis zu drei (3) `[compute_resource]`Abschnitte beziehen. Jeder dieser `[queue]`Abschnitte ist eine Partition inSlurm Workload Manager. Weitere Informationen erhalten Sie unter SlurmLeitfaden für den Modus mit mehreren Warteschlangen und Tutorial zum Modus mit mehreren Warteschlangen.

Jeder `[compute_resource]`Abschnitt in einer Warteschlange muss einen anderen Instanztyp haben, und jeder dieser Abschnitte `[compute_resource]` ist weiter in statische und dynamische Knoten unterteilt. Statische Knoten `[compute_resource]` werden jeweils von 1 bis zum Wert von nummeriert`min_count`. Dynamische Knoten `[compute_resource]` werden jeweils von eins (1) bis (`max_count`-`min_count`) nummeriert. Wenn beispielsweise 2 und 10 `min_count` `max_count` ist, `[compute_resource]` werden die dynamischen Knoten für diesen Wert von eins (1) bis acht (8) nummeriert. Der Wert kann jederzeit zwischen Null (0) und der maximalen Anzahl dynamischer Knoten in a liegen`[compute_resource]`.

Die Instances, die in die Compute-Flotte gestartet werden, werden dynamisch zugewiesen. Um dies zu verwalten, werden Hostnamen für jeden Knoten generiert. Das Format des Hostnamens lautet wie folgt:

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

- `$QUEUE`ist der Name der Warteschlange. Wenn der Abschnitt beispielsweise beginnt, `[queue` *`queue-name`*`]` dann "`$QUEUE`" ist "*`queue-name`*“.
- `$STATDYN`ist `st` für statische Knoten oder `dy` für dynamische Knoten.
- `$INSTANCE_TYPE`ist der Instanztyp für`[compute_resource]`, aus der `instance_type` Einstellung.
- `$NODENUM`ist die Nummer des Knotens. `$NODENUM`liegt zwischen eins (1) und dem Wert von `min_count` für statische Knoten und zwischen eins (1) und (`max_count`-`min_count`) für dynamische Knoten.

Sowohl Hostnamen als auch vollqualifizierte Domainnamen (FQDN) werden mithilfe von Amazon Route 53-Hosting-Zonen erstellt. Der FQDN ist`$HOSTNAME.$CLUSTERNAME.pcluster`, wobei der Name des `[cluster]`Abschnitts `$CLUSTERNAME` steht, der für den Cluster verwendet wird.

Verwenden Sie den `pcluster-config convert` Befehl, um Ihre Konfiguration in einen Warteschlangenmodus zu konvertieren. Es schreibt eine aktualisierte Konfiguration mit einem

einzigen [queue]Abschnitt namens[queue compute]. Diese Warteschlange enthält einen einzelnen [compute_resource]Abschnitt, der benannt ist[compute_resource default]. Das [queue compute] und [compute_resource default] hat Einstellungen, die aus dem angegebenen [cluster]Abschnitt migriert wurden.

SlurmLeitfaden für den Modus mit mehreren Warteschlangen

AWS ParallelCluster Version 2.9.0 führte den Modus mit mehreren Warteschlangen und eine neue Skalierungsarchitektur für Slurm Workload Manager (Slurm) ein.

Die folgenden Abschnitte bieten einen allgemeinen Überblick über die Verwendung eines Slurm Clusters mit der neu eingeführten Skalierungsarchitektur.

Übersicht

Die neue Skalierungsarchitektur basiert auf dem Cloud Slurm Scheduling Guide und dem Energiespar-Plugin. Weitere Informationen zum Energiespar-Plugin finden Sie im SlurmPower Saving Guide. In der neuen Architektur sind Ressourcen, die potenziell für einen Cluster verfügbar gemacht werden können, in der Slurm Konfiguration in der Regel als Cloud-Knoten vordefiniert.

Lebenszyklus eines Cloud-Knotens

Während ihres gesamten Lebenszyklus treten Cloud-Knoten in mehrere, wenn nicht sogar alle der folgenden Zustände ein:POWER_SAVING, POWER_UP (pow_up), ALLOCATED (alloc) und POWER_DOWN (pow_dn). In einigen Fällen kann ein Cloud-Knoten in den OFFLINE Status wechseln. In der folgenden Liste werden verschiedene Aspekte dieser Zustände im Lebenszyklus eines Cloud-Knotens beschrieben.

- Ein Knoten in einem POWER_SAVING Status wird mit einem ~ Suffix (zum Beispielidle~) in sinfo angezeigt. In diesem Status gibt es keine EC2 Instanz, die den Knoten unterstützt. SlurmKann dem Knoten jedoch weiterhin Jobs zuweisen.
- Ein Knoten, der in einen POWER_UP Status übergeht, wird mit einem # Suffix (zum Beispielidle#) in angezeigt. sinfo
- Wenn Slurm einem Knoten in einem Status eine Aufgabe zugewiesen wird, wechselt der Knoten automatisch in einen POWER_SAVING Status. POWER_UP Andernfalls können Knoten mithilfe des Befehls manuell in den POWER_UP Status versetzt werden. scontrol update nodename=*nodename* state=power_up In dieser Phase ResumeProgram wird der aufgerufen und EC2 Instances werden gestartet und konfiguriert, um einen POWER_UP Knoten zu unterstützen.

- Ein Knoten, der derzeit zur Verwendung verfügbar ist, wird ohne Suffix (z. B.`idle`) in angezeigt. `sinfo` Nachdem der Knoten eingerichtet wurde und dem Cluster beigetreten ist, steht er für die Ausführung von Jobs zur Verfügung. In dieser Phase ist der Knoten ordnungsgemäß konfiguriert und einsatzbereit. In der Regel empfehlen wir, dass die Anzahl der EC2 Instanzen der Anzahl der verfügbaren Knoten entspricht. In den meisten Fällen sind statische Knoten immer verfügbar, nachdem der Cluster erstellt wurde.
- Ein Knoten, der in einen `POWER_DOWN` Status übergeht, wird mit einem `%` Suffix (z. B.`idle%`) in angezeigt. `sinfo` Dynamische Knoten wechseln automatisch danach in den `POWER_DOWN` Status. `scaledown_idletime` Im Gegensatz dazu werden statische Knoten in den meisten Fällen nicht ausgeschaltet. Knoten können jedoch mithilfe des `scontrol update nodename=`*`nodename`* `state=powering_down` Befehls manuell in den `POWER_DOWN` Status versetzt werden. In diesem Zustand wird die einem Knoten zugeordnete Instanz beendet und der Knoten wird in den `POWER_SAVING` Zustand zurückgesetzt, in dem er future verwendet werden kann`scaledown_idletime`. Die `scaledown-idletime` Einstellung wird in der Slurm Konfiguration als `SuspendTimeout` Einstellung gespeichert.
- Ein Knoten, der offline ist, wird mit einem `*` Suffix (z. B.`down*`) in `sinfo` angezeigt. Ein Knoten geht offline, wenn der Slurm Controller den Knoten nicht kontaktieren kann oder wenn die statischen Knoten deaktiviert sind und die unterstützenden Instanzen beendet werden.

Betrachten Sie nun die im folgenden `sinfo` Beispiel gezeigten Knotenzustände.

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

Für die `efa-st-c5n18xlarge-1` Knoten `spot-st-t2large-[1-2]` und sind bereits Backing-Instances eingerichtet, sodass sie verwendet werden können. Die `ondemand-dy-c52xlarge-[1-2]` Knoten befinden sich im `POWER_UP` Status und sollten innerhalb weniger Minuten verfügbar sein. Der `gpu-dy-g38xlarge-1` Knoten befindet sich im `POWER_DOWN` Status und wechselt danach in den `POWER_SAVING` Status `scaledown_idletime` (standardmäßig 120 Sekunden).

Alle anderen Knoten befinden sich im `POWER_SAVING` Status, ohne dass sie von EC2 Instanzen unterstützt werden.

Mit einem verfügbaren Knoten arbeiten

Ein verfügbarer Knoten wird von einer EC2 Instanz unterstützt. Standardmäßig kann der Knotenname verwendet werden, um per SSH direkt auf die Instanz zuzugreifen (zum Beispiel`ssh efa-st-c5n18xlarge-1`). Die private IP-Adresse der Instanz kann mit dem `scontrol show nodes` *`nodename`* Befehl abgerufen und das `NodeAddr` Feld markiert werden. Bei Knoten, die nicht verfügbar sind, sollte das `NodeAddr` Feld nicht auf eine laufende EC2 Instanz verweisen. Vielmehr sollte es mit dem Knotennamen identisch sein.

Status und Einreichung von Job

In den meisten Fällen werden übermittelte Jobs sofort Knoten im System zugewiesen oder als ausstehend eingestuft, wenn alle Knoten zugewiesen sind.

Wenn die für einen Job zugewiesenen Knoten Knoten in einem `POWER_SAVING` Status enthalten, beginnt der Job mit einem `CF` oder`CONFIGURING`. Zu diesem Zeitpunkt wartet der Job darauf, dass die Knoten im `POWER_SAVING` Status in den `POWER_UP` Status wechseln und verfügbar sind.

Nachdem alle für einen Job zugewiesenen Knoten verfügbar sind, wechselt der Job in den Status `RUNNING` (`R`).

Standardmäßig werden alle Jobs an die Standardwarteschlange (auch Partition inSlurm) übergeben. Dies wird durch ein `*` Suffix nach dem Warteschlangennamen gekennzeichnet. Sie können mit der Option zum Einreichen von `-p` Jobs eine Warteschlange auswählen.

Alle Knoten sind mit den folgenden Funktionen konfiguriert, die in Befehlen zur Auftragsübermittlung verwendet werden können:

- Ein Instanztyp (zum Beispiel`c5.xlarge`)
- Ein Knotentyp (Dies ist entweder `dynamic` oder`static`.)

Sie können alle für einen bestimmten Knoten verfügbaren Funktionen anzeigen, indem Sie den `scontrol show nodes` *`nodename`* Befehl verwenden und die `AvailableFeatures` Liste überprüfen.

Ein weiterer Gesichtspunkt sind Arbeitsplätze. Betrachten Sie zunächst den Ausgangsstatus des Clusters, den Sie anzeigen können, indem Sie den `sinfo` Befehl ausführen.

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

Beachten Sie, dass `spot` dies die Standardwarteschlange ist. Sie wird durch das `*` Suffix angezeigt.

Sendet einen Job an einen statischen Knoten in der Standardwarteschlange (`spot`).

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

Sendet einen Job an einen dynamischen Knoten in der `EFA` Warteschlange.

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

Sendet einen Job an acht (8) `c5.2xlarge` Knoten und zwei (2) `t2.xlarge` Knoten an die `ondemand` Warteschlange.

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

Senden Sie einen Job an einen GPU-Knoten in der `gpu` Warteschlange.

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

Betrachten Sie nun den Status der Jobs mithilfe des `squeue` Befehls.

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

Die Jobs 7 und 8 (in den `efa` Warteschlangen `spot` und) werden bereits ausgeführt (`R`). Die Jobs 12 und 13 werden noch konfiguriert (`CF`) und warten wahrscheinlich darauf, dass die Instanzen verfügbar werden.

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

Status und Funktionen des Knotens

In den meisten Fällen werden Knotenstatus AWS ParallelCluster gemäß den spezifischen Prozessen im Cloud-Knoten-Lebenszyklus, die weiter oben in diesem Thema beschrieben wurden, vollständig verwaltet.

Ersetzt oder beendet jedoch AWS ParallelCluster auch fehlerhafte Knoten in `DRAINED` Zuständen `DOWN` und Knoten, die über fehlerhafte Backing-Instances verfügen. Weitere Informationen finden Sie unter `clustermgtd`.

Status der Partition

AWS ParallelCluster unterstützt die folgenden Partitionsstatus. Eine Slurm Partition ist eine Warteschlange in AWS ParallelCluster.

- `UP`: Zeigt an, dass sich die Partition in einem aktiven Zustand befindet. Dies ist der Standardstatus einer Partition. In diesem Zustand sind alle Knoten in der Partition aktiv und können verwendet werden.
- `INACTIVE`: Zeigt an, dass sich die Partition im inaktiven Zustand befindet. In diesem Zustand werden alle Instanzen, die Knoten einer inaktiven Partition unterstützen, beendet. Neue Instanzen werden für Knoten in einer inaktiven Partition nicht gestartet.

pcluster starten und beenden

Wenn `pcluster stop` es ausgeführt wird, werden alle Partitionen in den `INACTIVE` Status versetzt, und die AWS ParallelCluster Prozesse behalten die Partitionen im `INACTIVE` Status bei.

Wenn ausgeführt `pcluster start` wird, werden zunächst alle Partitionen in den `UP` Status versetzt. AWS ParallelCluster Prozesse halten die Partition jedoch nicht in einem `UP` Zustand. Sie müssen den Partitionsstatus manuell ändern. Alle statischen Knoten sind nach einigen Minuten verfügbar. Beachten Sie, dass das Einstellen einer Partition auf keine dynamische Kapazität erhöht. `UP` Wenn größer als `initial_count` ist`max_count`, ist `initial_count` es möglicherweise nicht erfüllt, wenn der Partitionsstatus in den `UP` Status geändert wird.

Wenn `pcluster start` und ausgeführt `pcluster stop` werden, können Sie den Status des Clusters überprüfen, indem Sie den `pcluster status` Befehl ausführen und das überprüfen`ComputeFleetStatus`. In der folgenden Liste sind mögliche Status aufgeführt:

- `STOP_REQUESTED`: Die `pcluster stop` Anfrage wird an den Cluster gesendet.
- `STOPPING`: Der `pcluster` Prozess stoppt derzeit den Cluster.
- `STOPPED`: Der `pcluster` Prozess hat den Stoppvorgang abgeschlossen, alle Partitionen befinden sich im `INACTIVE` Status und alle Recheninstanzen wurden beendet.
- `START_REQUESTED`: Die `pcluster start` Anfrage wird an den Cluster gesendet.
- `STARTING`: Der `pcluster` Prozess startet gerade den Cluster
- `RUNNING`: Der `pcluster` Prozess hat den Startvorgang abgeschlossen, alle Partitionen befinden sich im `UP` Status und statische Knoten sind nach einigen Minuten verfügbar.

Manuelle Steuerung von Warteschlangen

In einigen Fällen möchten Sie möglicherweise die Knoten oder die Warteschlange (auch Partition genanntSlurm) in einem Cluster manuell steuern. Sie können Knoten in einem Cluster mithilfe der folgenden gängigen Verfahren verwalten.

- Dynamische Knoten im `POWER_SAVING` Status einschalten: Führen Sie den `scontrol update nodename=`*`nodename`* `state=power_up` Befehl aus oder senden Sie einen `sleep 1` Platzhalter-Job, der eine bestimmte Anzahl von Knoten anfordert, und verlassen Sie sich daraufSlurm, dass die erforderliche Anzahl von Knoten eingeschaltet wird.
- Dynamische Knoten vorher ausschalten`scaledown_idletime`: Stellen Sie dynamische Knoten `DOWN` mit dem `scontrol update nodename=`*`nodename`* `state=down` Befehl auf ein. AWS

ParallelCluster beendet die ausgefallenen dynamischen Knoten automatisch und setzt sie zurück. Im Allgemeinen empfehlen wir nicht, Knoten so einzustellen, dass sie den Befehl `POWER_DOWN` direkt verwenden. `scontrol update nodename=nodename state=power_down` Das liegt daran, dass der Abschaltvorgang AWS ParallelCluster automatisch abgewickelt wird. Es ist kein manueller Eingriff erforderlich. Daher empfehlen wir, dass Sie versuchen, die Knoten nach `DOWN` Möglichkeit auf zu setzen.

- Deaktivieren Sie eine Warteschlange (Partition) oder stoppen Sie alle statischen Knoten in einer bestimmten Partition: Stellen Sie `INACTIVE` mit dem `scontrol update partition=queue name state=inactive` Befehl die Warteschlange spezifisch ein. Dadurch werden alle Instanzen beendet, die Knoten in der Partition unterstützen.
- Eine Warteschlange (Partition) aktivieren: Stellen Sie `INACTIVE` mit dem `scontrol update partition=queue name state=up` Befehl eine bestimmte Warteschlange ein.

Skalierungsverhalten und Anpassungen

Hier ist ein Beispiel für den normalen Skalierungsablauf:

- Der Scheduler empfängt einen Job, für den zwei Knoten erforderlich sind.
- Der Scheduler versetzt zwei Knoten in einen `POWER_UP` Status und ruft `ResumeProgram` mit den Knotennamen auf (zum Beispiel`queue1-dy-c5xlarge-[1-2]`).
- `ResumeProgram`startet zwei EC2 Instances und weist die privaten IP-Adressen und Hostnamen von `queue1-dy-c5xlarge-[1-2]` zu. Warten Sie `ResumeTimeout` (der Standardzeitraum beträgt 60 Minuten (1 Stunde)), bevor die Knoten zurückgesetzt werden.
- Die Instanzen werden konfiguriert und treten dem Cluster bei. Der Job wird auf Instanzen ausgeführt.
- Die Job ist erledigt.
- Nach Ablauf der konfigurierten `SuspendTime` Zeit (die auf gesetzt ist`scaledown_idletime`), werden die Instanzen vom Scheduler in den `POWER_SAVING` Status versetzt. Der Scheduler `queue1-dy-c5xlarge-[1-2]` versetzt den `POWER_DOWN` Status und ruft `SuspendProgram` mit den Knotennamen auf.
- `SuspendProgram`wird für zwei Knoten aufgerufen. Knoten bleiben in diesem `POWER_DOWN` Zustand, z. B. indem sie eine Zeit `idle%` lang verbleiben `SuspendTimeout` (der Standardzeitraum beträgt 120 Sekunden (2 Minuten)). Nachdem `clustermgtd` erkannt wurde, dass Knoten heruntergefahren werden, werden die unterstützenden Instances beendet. Anschließend wird die Konfiguration `queue1-dy-c5xlarge-[1-2]` in den Ruhezustand versetzt

und die private IP-Adresse und der Hostname zurückgesetzt, sodass sie für future Jobs wieder eingeschaltet werden können.

Wenn nun etwas schief geht und eine Instanz für einen bestimmten Knoten aus irgendeinem Grund nicht gestartet werden kann, passiert Folgendes.

- Der Scheduler empfängt einen Job, für den zwei Knoten erforderlich sind.
- Der Scheduler versetzt zwei Cloud-Bursting-Knoten in den `POWER_UP` Status und ruft sie `ResumeProgram` mit den Knotennamen auf (zum Beispiel). `queue1-dy-c5xlarge-[1-2]`
- `ResumeProgram`startet nur eine (1) EC2 Instanz und konfiguriert`queue1-dy-c5xlarge-1`, aber es konnte keine Instanz für gestartet werden. `queue1-dy-c5xlarge-2`
- `queue1-dy-c5xlarge-1`ist nicht betroffen und wird nach Erreichen des `POWER_UP` Status online geschaltet.
- `queue1-dy-c5xlarge-2`wird in den `POWER_DOWN` Status versetzt, und der Job wird automatisch in die Warteschlange gestellt, da ein Slurm Knotenausfall erkannt wird.
- `queue1-dy-c5xlarge-2`wird danach verfügbar `SuspendTimeout` (die Standardeinstellung ist 120 Sekunden (2 Minuten)). In der Zwischenzeit wird der Job in die Warteschlange gestellt und kann auf einem anderen Knoten ausgeführt werden.
- Der obige Vorgang wird wiederholt, bis der Job auf einem verfügbaren Knoten ausgeführt werden kann, ohne dass ein Fehler auftritt.

Es gibt zwei Timing-Parameter, die bei Bedarf angepasst werden können.

- `ResumeTimeout`(Die Standardeinstellung ist 60 Minuten (1 Stunde)): `ResumeTimeout` Steuert die Slurm Wartezeit, bis der Knoten in den Status „Heruntergefahren“ versetzt wird.
  - Es kann nützlich sein, dies zu verlängern, wenn Ihr pre/post Installationsvorgang fast so lange dauert.
  - Dies ist auch die maximale AWS ParallelCluster Wartezeit, bis ein Knoten ersetzt oder zurückgesetzt wird, falls ein Problem auftritt. Rechenknoten beenden sich selbst, wenn während des Starts oder der Einrichtung ein Fehler auftritt. Als Nächstes ersetzt der AWS ParallelCluster Prozess auch den Knoten, wenn er feststellt, dass die Instanz beendet wurde.
- `SuspendTimeout`(Die Standardeinstellung ist 120 Sekunden (2 Minuten)): `SuspendTimeout` Steuert, wie schnell Knoten wieder im System platziert und wieder einsatzbereit sind.

- Ein kürzerer Wert `SuspendTimeout` würde bedeuten, dass Knoten schneller zurückgesetzt werden und Slurm dass häufiger versucht werden kann, Instances zu starten.
- Bei einer längeren `SuspendTimeout` Einstellung werden ausgefallene Knoten langsamer zurückgesetzt. In der Zwischenzeit versucht erSlurm, andere Knoten zu verwenden. Wenn `SuspendTimeout` es länger als ein paar Minuten dauert, wird Slurm versucht, alle Knoten im System zu durchlaufen. Eine längere Laufzeit `SuspendTimeout` könnte für große Systeme (über 1.000 Knoten) von Vorteil sein, um die stress zu verringern, Slurm indem ausfallende Jobs häufig in die Warteschlange gestellt werden.
- Beachten Sie, dass sich `SuspendTimeout` dies nicht auf die Wartezeit bezieht, bis AWS ParallelCluster eine Backing-Instance für einen Knoten beendet wurde. Backing-Instances für `power down` Knoten werden sofort beendet. Der Terminierungsvorgang ist in der Regel nach wenigen Minuten abgeschlossen. Während dieser Zeit befindet sich der Knoten jedoch im ausgeschalteten Zustand und kann nicht im Scheduler verwendet werden.

Protokolle für die neue Architektur

Die folgende Liste enthält die wichtigsten Protokolle für die Architektur mit mehreren Warteschlangen. Der mit Amazon CloudWatch Logs verwendete Protokollstreamname hat das Format*`{hostname}`*`.`*`{instance_id}`*`.`*`{logIdentifier}`*, in dem die Protokollnamen *`logIdentifier`* folgen. Weitere Informationen finden Sie unter Integration mit Amazon CloudWatch Logs.

- `ResumeProgram`:

  `/var/log/parallelcluster/slurm_resume.log` (`slurm_resume`)
- `SuspendProgram`:

  `/var/log/parallelcluster/slurm_suspend.log` (`slurm_suspend`)
- `clustermgtd`:

  `/var/log/parallelcluster/clustermgtd.log` (`clustermgtd`)
- `computemgtd`:

  `/var/log/parallelcluster/computemgtd.log` (`computemgtd`)
- `slurmctld`:

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

- `slurmd`:

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

Häufige Probleme und Anleitung zum Debuggen:

Knoten, die nicht gestartet, hochgefahren oder dem Cluster nicht beigetreten sind:

- Dynamische Knoten:
  - Sehen Sie im `ResumeProgram` Protokoll nach, ob jemals mit dem Knoten aufgerufen `ResumeProgram` wurde. Falls nicht, überprüfen Sie das `slurmctld` Protokoll, um festzustellen, ob Slurm jemals versucht wurde, `ResumeProgram` mit dem Knoten anzurufen. Beachten Sie, dass falsche Berechtigungen dazu führen `ResumeProgram` können, dass der Vorgang unbemerkt fehlschlägt.
  - Wenn aufgerufen `ResumeProgram` wird, überprüfen Sie, ob eine Instanz für den Knoten gestartet wurde. Wenn die Instance nicht gestartet werden kann, sollte es eine klare Fehlermeldung geben, warum die Instance nicht gestartet werden konnte.
  - Wenn eine Instance gestartet wurde, ist möglicherweise während des Bootstrap-Vorgangs ein Problem aufgetreten. Suchen Sie die entsprechende private IP-Adresse und Instanz-ID aus dem `ResumeProgram` Protokoll und sehen Sie sich die entsprechenden Bootstrap-Protokolle für die jeweilige Instanz in CloudWatch Logs an.
- Statische Knoten:
  - Prüfen Sie im `clustermgtd` Protokoll, ob Instanzen für den Knoten gestartet wurden. Wenn nicht, sollte es eindeutige Fehler darüber geben, warum die Instances nicht gestartet werden konnten.
  - Wenn eine Instance gestartet wurde, liegt ein Problem beim Bootstrap-Vorgang vor. Suchen Sie die entsprechende private IP und Instanz-ID aus dem `clustermgtd` Protokoll und sehen Sie sich die entsprechenden Bootstrap-Protokolle für die jeweilige Instanz in CloudWatch Logs an.

Knoten wurden unerwartet ersetzt oder beendet, Knotenausfälle

- Knoten replaced/terminated unerwartet
  - In den meisten Fällen `clustermgtd` erledigt es alle Wartungsaktionen für Knoten. Um zu überprüfen, ob ein Knoten `clustermgtd` ersetzt oder beendet wurde, überprüfen Sie das `clustermgtd` Protokoll.

  - Wenn der Knoten `clustermgtd` ersetzt oder beendet wurde, sollte eine Meldung erscheinen, die den Grund für die Aktion angibt. Wenn der Grund mit dem Scheduler zusammenhängt (zum Beispiel der Knoten`DOWN`), suchen Sie im `slurmctld` Protokoll nach weiteren Einzelheiten. Wenn der Grund EC2 damit zusammenhängt, verwenden Sie Tools, um den Status oder die Protokolle für diese Instanz zu überprüfen. Sie können beispielsweise überprüfen, ob für die Instanz Ereignisse geplant waren oder ob die EC2 Integritätsprüfungen nicht bestanden haben.
  - Falls der Knoten `clustermgtd` nicht beendet wurde, überprüfen Sie, ob der Knoten `computemgtd` beendet wurde oder ob die Instance EC2 beendet wurde, um eine Spot-Instance zurückzugewinnen.
- Knotenausfälle
  - In den meisten Fällen werden Jobs automatisch in die Warteschlange gestellt, wenn ein Knoten ausfällt. Schauen Sie im `slurmctld` Protokoll nach, warum ein Job oder ein Knoten ausgefallen ist, und analysieren Sie die Situation von dort aus.

Fehler beim Ersetzen oder Beenden von Instanzen, Fehler beim Herunterfahren von Knoten

- `clustermgtd`Behandelt im Allgemeinen alle erwarteten Aktionen zum Beenden von Instanzen. Sehen Sie im `clustermgtd` Protokoll nach, warum ein Knoten nicht ersetzt oder beendet werden konnte.
- Wenn dynamische Knoten ausfallen`scaledown_idletime`, schauen Sie im `SuspendProgram` Protokoll nach, ob ein Programm `slurmctld` mit dem spezifischen Knoten als Argument gestartet wurde. Note führt eigentlich `SuspendProgram` keine bestimmte Aktion aus. Vielmehr protokolliert es nur, wenn es aufgerufen wird. Alle Instanzbeendigungen und `NodeAddr` Resets sind bis abgeschlossen`clustermgtd`. Slurmfügt Knoten in den Bereich `IDLE` danach ein`SuspendTimeout`.

Andere Probleme

- AWS ParallelCluster trifft keine Entscheidungen zur Stellenzuweisung oder Skalierung. Es versucht einfach, Ressourcen gemäß den Anweisungen zu starten, zu beenden und zu Slurm verwalten.

  Bei Problemen mit der Auftragszuweisung, der Knotenzuweisung und der Skalierungsentscheidung suchen Sie im `slurmctld` Protokoll nach Fehlern.

Torque Resource Manager (`torque`)

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von SGE oder -Schedulern AWS ParallelCluster nicht unterstützt. Torque Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.

AWS ParallelCluster Versionen 2.11.4 und früher verwenden 6.1.2. Torque Resource Manager Weitere Informationen zu Torque Resource Manager 6.1.2 finden Sie unter http://docs.adaptivecomputing.com/torque/6-1-2/releaseNotes/torquerelnote.htm. Dokumentation finden Sie unter http://docs.adaptivecomputing.com/torque/6-1-2/adminGuide/torque.htm. Sie finden den Quellcode unter https://github.com/adaptivecomputing/torque/tree/6.1.2.

AWS ParallelCluster Versionen 2.4.0 und früher verwenden 6.0.2. Torque Resource Manager Versionshinweise finden Sie unter http://docs.adaptivecomputing.com/torque/6-0-2/releaseNotes/torqueReleaseNotes6.0.2.pdf. Dokumentation finden Sie unter http://docs.adaptivecomputing.com/torque/6-0-2/adminGuide/help.htm. Sie finden den Quellcode unter https://github.com/adaptivecomputing/torque/tree/6.0.2.

AWS Batch (`awsbatch`)

Informationen zu finden Sie AWS Batch unter. AWS Batch Die Dokumentation finden Sie im AWS Batch Benutzerhandbuch.

AWS ParallelCluster CLI-Befehle für AWS Batch

Wenn Sie den `awsbatch` Scheduler verwenden, AWS Batch werden die AWS ParallelCluster CLI-Befehle für automatisch im AWS ParallelCluster Hauptknoten installiert. Die CLI verwendet AWS Batch API-Operationen und ermöglicht die folgenden Operationen:

- Übermitteln und Verwalten von Aufgaben.
- Überwachen von Aufgaben, Warteschlangen und Hosts.
- Spiegeln herkömmlicher Scheduler-Befehle.

> **Important**
>
> AWS ParallelCluster unterstützt keine GPU-Jobs für AWS Batch. Weitere Informationen finden Sie unter GPU-Jobs.

Themen

- awsbsub
- awsbstat
- awsbout
- awsbkill
- awsbqueues
- awsbhosts

awsbsub

Sendet Jobs an die Job-Warteschlange des Clusters.

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

> **Important**
>
> AWS ParallelCluster unterstützt keine GPU-Jobs für AWS Batch. Weitere Informationen finden Sie unter GPU-Jobs.

Positionale Argumente

command

Sendet den Job (der angegebene Befehl muss auf den Recheninstanzen verfügbar sein) oder den Namen der zu übertragenden Datei. Siehe auch `--command-file`.

arguments

(Optional) Gibt Argumente für den Befehl oder die Befehlsdatei an.

Benannte Argumente

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

Benennt die Aufgabe. Das erste Zeichen muss entweder ein Buchstabe oder eine Zahl sein. Der Jobname kann Buchstaben (sowohl Groß- als auch Kleinbuchstaben), Zahlen, Bindestriche und Unterstriche enthalten und bis zu 128 Zeichen lang sein.

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

Gibt den zu verwendenden Cluster an.

-cf, --command-file

Zeigt an, dass der Befehl eine Datei ist, die an Datenverarbeitungs-Instances übertragen werden soll.

Standard: False

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

Gibt den Ordner an, der als Arbeitsverzeichnis der Aufgabe verwendet werden soll. Wenn kein Arbeitsverzeichnis angegeben ist, wird der Job im `job-`*`<AWS_BATCH_JOB_ID>`* Unterordner des Home-Verzeichnisses des Benutzers ausgeführt. Sie können entweder diesen Parameter oder den Parameter `--parent-working-dir` verwenden.

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

Gibt den übergeordneten Ordner des Arbeitsverzeichnisses des Jobs an. Wenn kein übergeordnetes Arbeitsverzeichnis angegeben ist, wird standardmäßig das Basisverzeichnis des Benutzers verwendet. Ein Unterordner mit dem Namen `job-`*`<AWS_BATCH_JOB_ID>`* wird im übergeordneten Arbeitsverzeichnis erstellt. Sie können entweder diesen Parameter oder den Parameter `--working-dir` verwenden.

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

Gibt die an die Datenverarbeitungs-Instances zu übertragende Datei im Arbeitsverzeichnis der Aufgabe an. Sie können mehrere Eingabedateiparameter angeben.

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

Gibt die Anzahl von v anCPUs , die für den Container reserviert werden sollen. Wenn es zusammen mit verwendet wird`-nodes`, identifiziert es die Anzahl von v CPUs für jeden Knoten.

Standard: 1

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

Gibt die harte Grenze des Arbeitsspeichers (in MiB) für die Aufgabe an. Wenn Ihr Job versucht, das hier angegebene Speicherlimit zu überschreiten, wird der Job beendet.

Standard: 128

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

Gibt eine durch Komma getrennte Liste von Umgebungsvariablennamen zum Exportieren in die Aufgabenumgebung an. Zum Exportieren aller Umgebungsvariablen geben Sie „all“ an. Beachten Sie, dass eine Liste mit 'allen' Umgebungsvariablen nicht die im `-env-blacklist` Parameter aufgeführten Variablen oder Variablen enthält, die mit dem `AWS_*` Präfix `PCLUSTER_*` oder beginnen.

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

Gibt eine durch Komma getrennte Liste von Umgebungsvariablennamen an, die nicht in die Aufgabenumgebung exportiert werden sollen. Standardmäßig werden `HOME`, `PWD`, `USER`, `PATH`, `LD_LIBRARY_PATH`, `TERM` und `TERMCAP` nicht exportiert.

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

Gibt an, wie oft ein Job in den `RUNNABLE` Status versetzt werden soll. Sie können zwischen einem und zehn Versuche angeben. Wenn der Wert der Versuche größer als 1 ist, wird der Auftrag wiederholt, falls er fehlschlägt, bis er den angegebenen `RUNNABLE` Status erreicht hat.

Standard: 1

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

Gibt die Zeitdauer in Sekunden (gemessen anhand des `startedAt` Zeitstempels des Auftragsversuchs) an, nach deren Ablauf Ihr Job AWS Batch beendet wird, falls er noch nicht abgeschlossen ist. Der Timeout-Wert muss mindestens 60 Sekunden betragen.

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

Gibt die Anzahl der Knoten an, die für die Aufgabe zu reservieren sind. Geben Sie einen Wert für diesen Parameter an, um die parallel Übermittlung mehrerer Knoten zu ermöglichen.

> **Note**
>
> parallel Jobs mit mehreren Knoten werden nicht unterstützt, wenn der `cluster_type` Parameter auf `spot` gesetzt ist.

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

Zeigt die Größe des Arrays an. Sie können einen Wert zwischen 2 und 10.000 auswählen. Wenn Sie Array-Eigenschaften für eine Aufgabe angeben, wird sie zu einer Array-Aufgabe.

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

Gibt eine durch Strichpunkte getrennte Liste von Abhängigkeiten für eine Aufgabe an. Eine Aufgabe kann von maximal 20 Aufgaben abhängen. Sie können eine `SEQUENTIAL` Typabhängigkeit angeben, ohne eine Job-ID für Array-Jobs anzugeben. Eine sequenzielle Abhängigkeit ermöglicht jeder untergeordneten Array-Aufgabe, sequentiell abgeschlossen zu werden (beginnend mit Index 0). Sie können auch eine Abhängigkeit vom Typ „N_TO_N“ mit einer Aufgaben-ID für Array-Aufgaben angeben. Eine Abhängigkeit vom Typ N_TO_N bedeutet, dass jeder untergeordnete Index dieser Aufgabe warten muss, bis der entsprechende untergeordnete Index jeder Abhängigkeit abgeschlossen ist. Die Syntax für diesen Parameter lautet „jobID=*`<string>`*, type=*`<string>`*;...“.

`awsbstat`

Zeigt die Aufgaben in der Aufgabenwarteschlange des Clusters.

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

Positionale Argumente

`job_ids`

Gibt die durch Leerzeichen getrennte Liste von Jobs an, die in der Ausgabe angezeigt IDs werden sollen. Wenn die Aufgabe ein Aufgaben-Array ist, werden alle untergeordneten Aufgaben angezeigt. Wenn eine einzelne Aufgabe angefordert wird, wird sie in einer detaillierten Version angezeigt.

Benannte Argumente

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

Gibt den Cluster an, der verwendet werden soll.

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

Gibt eine durch Komma getrennte Liste der Job-Status an, die berücksichtigt werden sollen. Der standardmäßige Aufgabenstatus lautet „active“. Akzeptierte Werte sind: `SUBMITTED`, `PENDING`, `RUNNABLE`, `STARTING`, `RUNNING`, `SUCCEEDED`, `FAILED` und `ALL`.

Standard: „`SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING`“

-e, --expand-children

Erweitert Aufgaben mit untergeordneten Elementen (Array und Multi-Knoten parallel).

Standard: False

-d, --details

Zeigt Aufgabendetails.

Standard: False

awsbout

Zeigt die Ausgabe einer bestimmten Aufgabe an.

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

Positionale Argumente

job_id

Gibt die Aufgaben-ID an.

Benannte Argumente

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

Gibt den Cluster an, der verwendet werden soll.

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

Ruft die ersten *HEAD* Zeilen der Jobausgabe ab.

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

Ruft die letzten <tail>-Zeilen der Aufgabenausgabe ab.

-s, --stream

Ruft die Aufgabenausgabe ab und wartet dann darauf, dass weitere Ausgaben erstellt werden. Dieses Argument kann zusammen mit „–tail“ verwendet werden, um von den aktuellen <tail>-Zeilen der Aufgabenausgabe zu beginnen.

Standard: False

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

Legt den Streaming-Zeitraum fest.

Standard: 5

awsbkill

Bricht im Cluster übermittelte Aufgaben ab oder beendet sie.

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

Positionale Argumente

job_ids

Gibt die durch Leerzeichen getrennte Liste der Jobs IDs an, die abgebrochen oder beendet werden sollen.

Benannte Argumente

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

Gibt den Namen des Clusters an, der verwendet werden soll.

-r *REASON*, --reason *REASON*

Gibt die Nachricht an, die einer Aufgabe angefügt werden soll und den Grund für ihren Abbruch angibt.

Standard: „Terminated by the user“

awsbqueues

Zeigt die Aufgabenwarteschlange an, die mit dem Cluster verknüpft ist.

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

Positionale Argumente

job_queues

Gibt die durch Leerzeichen getrennte Liste von Warteschlangennamen an, die angezeigt werden sollen. Wenn eine einzelne Warteschlange angefordert wird, wird sie in einer detaillierten Version angezeigt.

Benannte Argumente

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

Gibt den Namen des Clusters an, der verwendet werden soll.

-d, --details

Gibt an, ob die Details der Warteschlangen angezeigt werden sollen.

Standard: False

awsbhosts

Zeigt die Hosts, die zur Datenverarbeitungsumgebung des Clusters gehören.

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

Positionale Argumente

instance_ids

Gibt eine durch Leerzeichen getrennte Liste von Instanzen IDs an. Wenn eine einzelne Instance angefordert wird, wird sie in einer detaillierten Version angezeigt.

Benannte Argumente

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

Gibt den Namen des Clusters an, der verwendet werden soll.

-d, --details

Gibt an, ob die Details des Hosts angezeigt werden sollen.

Standard: False

AWS ParallelCluster Ressourcen und Tagging

Mit können AWS ParallelCluster Sie Tags erstellen, um Ihre AWS ParallelCluster Ressourcen zu verfolgen und zu verwalten. Sie definieren die Tags, die Sie erstellen und AWS CloudFormation an alle Cluster-Ressourcen weitergeben möchten, im `tags` Abschnitt der Cluster-Konfigurationsdatei. Sie können auch AWS ParallelCluster automatisch generierte Tags verwenden, um Ihre Ressourcen zu verfolgen und zu verwalten.

Wenn Sie einen Cluster erstellen, werden der Cluster und seine Ressourcen mit den in diesem Abschnitt definierten Tags AWS ParallelCluster und AWS Systemtags gekennzeichnet.

AWS ParallelCluster wendet Tags auf die Cluster-Instances, Volumes und Ressourcen an. AWS CloudFormation Wendet AWS System-Tags auf die Cluster-Instances an, um den Cluster-Stack zu identifizieren. EC2 Wendet System-Tags auf die Instances an, um die EC2 Cluster-Startvorlagen zu identifizieren. Sie können diese Tags verwenden, um Ihre AWS ParallelCluster Ressourcen anzuzeigen und zu verwalten.

Sie können AWS Systemtags nicht ändern. Um Beeinträchtigungen der AWS ParallelCluster Funktionalität zu vermeiden, sollten Sie AWS ParallelCluster Tags nicht ändern.

Im Folgenden finden Sie ein Beispiel für ein AWS System-Tag für eine AWS ParallelCluster Ressource. Sie können sie nicht ändern.

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

Im Folgenden finden Sie Beispiele für AWS ParallelCluster Tags, die auf eine Ressource angewendet wurden. Ändern Sie sie nicht.

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

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

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

Sie können diese Tags im EC2 Abschnitt der einsehen AWS Management Console.

Anzeigen von Tags

1. Navigieren Sie in der EC2 Konsole unter https://console.aws.amazon.com/ec2/.
2. Um alle Cluster-Tags anzuzeigen, wählen Sie im Navigationsbereich Tags aus.
3. Um Cluster-Tags nach Instance anzuzeigen, wählen Sie im Navigationsbereich Instances aus.
4. Wählen Sie eine Cluster-Instance aus.
5. Wählen Sie in den Instanzdetails den Tab Tags verwalten und sehen Sie sich die Tags an.
6. Wählen Sie in den Instanzdetails den Tab Speicher.
7. Wählen Sie die Volume-ID aus.
8. Wählen Sie unter Volumes das Volume aus.
9. Wählen Sie in den Banddetails den Tab „Tags“ und sehen Sie sich die Tags an.

AWS ParallelCluster Instanz-Tags für den Hauptknoten

Schlüssel	Tag-Wert
`ClusterName`	*`clustername`*
`Name`	`Master`
`Application`	`parallelcluster-` *`clustername`*

Schlüssel	Tag-Wert
aws:ec2launchtemplate:id	*lt-1234567890abcdef0*
aws:ec2launchtemplate:version	*1*
aws-parallelcluster-node-type	Master
aws:cloudformation:stack-name	parallelcluster- *clustername* -MasterServerSubstack- *ABCD1234EFGH*
aws:cloudformation:logical-id	MasterServer
aws:cloudformation:stack-id	arn:aws:cloudformation: *region-id* :*ACCOUNTID* :stack/parallelcluster- *clustername* -MasterServerSubstack- *ABCD1234EFGH* /*1234abcd-12ab-12ab-12ab-1234567890abcdef0*
Version	*2.11.9*

AWS ParallelCluster Root-Volume-Tags für den Hauptknoten

Tag-Schlüssel	Tag-Wert
ClusterName	*clustername*
Application	parallelcluster- *clustername*
aws-parallelcluster-node-type	Master

AWS ParallelCluster Instanz-Tags für Knoten berechnen

Schlüssel	Tag-Wert
ClusterName	*clustername*

Schlüssel	Tag-Wert
`aws-parallelcluster-node-type`	`Compute`
`aws:ec2launchtemplate:id`	*`lt-1234567890abcdef0`*
`aws:ec2launchtemplate:version`	*`1`*
`QueueName`	*`queue-name`*
`Version`	*`2.11.9`*

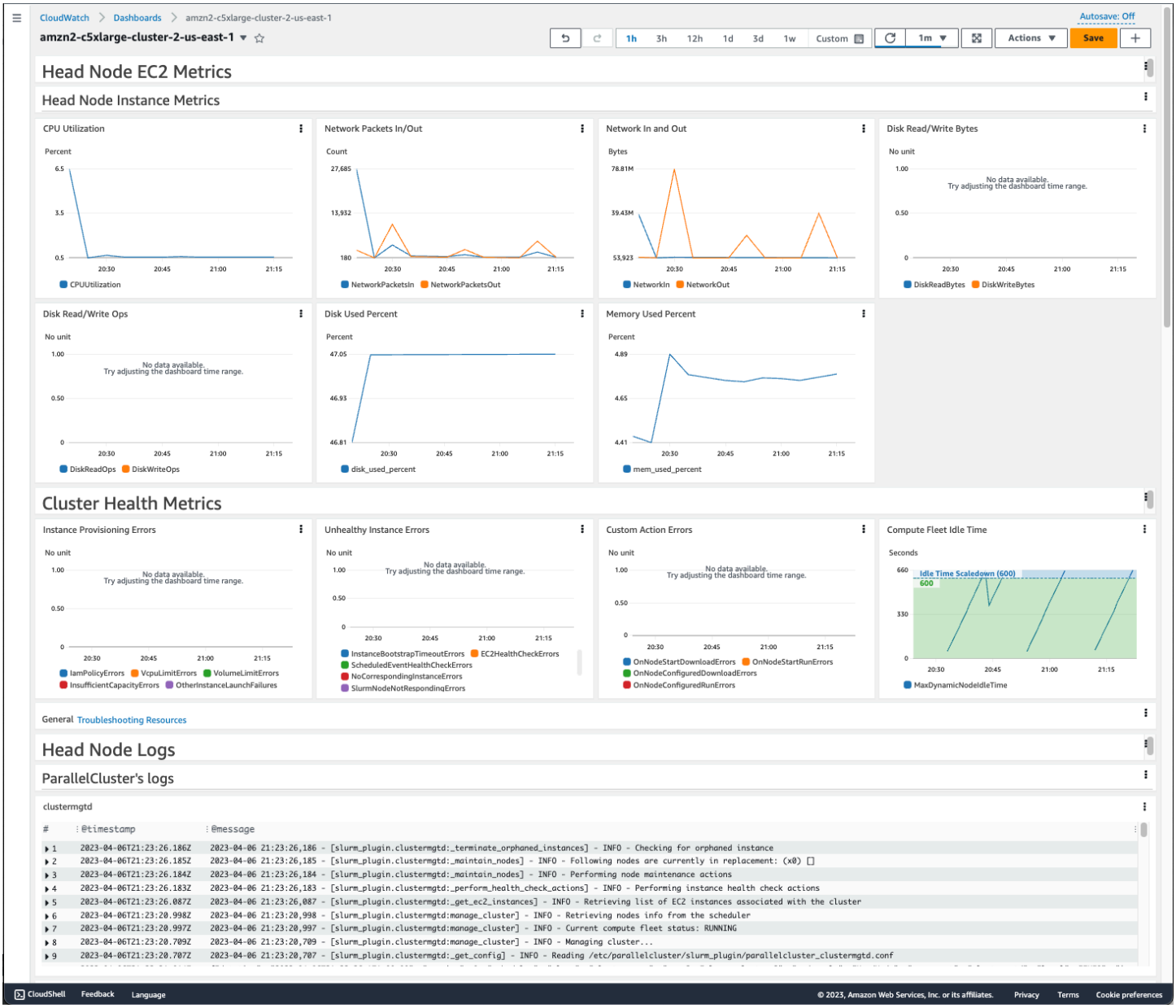
AWS ParallelCluster Root-Volume-Tags für Knoten berechnen

Tag-Schlüssel	Tag-Wert
`ClusterName`	*`clustername`*
`Application`	`parallelcluster-` *`clustername`*
`aws-parallelcluster-node-type`	`Compute`
`QueueName`	*`queue-name`*
`Version`	*`2.11.9`*

CloudWatch Amazon-Dashboard

Ab AWS ParallelCluster Version 2.10.0 wird bei der Erstellung des Clusters ein CloudWatch Amazon-Dashboard erstellt. Dies macht es einfacher, die Knoten in Ihrem Cluster zu überwachen und die in Amazon Logs gespeicherten CloudWatch Protokolle einzusehen. Der Name des Dashboards lautet`parallelcluster-`*`ClusterName`*`-`*`Region`*. *`ClusterName`*ist der Name Ihres Clusters und *`Region`* ist der AWS-Region des Clusters. Sie können in der Konsole oder durch Öffnen auf das Dashboard zugreifen`https://console.aws.amazon.com/cloudwatch/home?region=`*`Region`*`#dashboards:name=parallelcluster-`*`ClusterName`*.

Die folgende Abbildung zeigt ein CloudWatch Beispiel-Dashboard für einen Cluster.



Im ersten Abschnitt des Dashboards werden Diagramme der Head EC2 Node-Metriken angezeigt. Wenn Ihr Cluster über gemeinsam genutzten Speicher verfügt, werden im nächsten Abschnitt Messwerte für gemeinsam genutzten Speicher angezeigt. Im letzten Abschnitt werden Head Node Logs gruppiert nach ParallelCluster Logs, Scheduler-Logs, NICE DCV DCV-Integrationslogs und Systemprotokollen aufgelistet.

Weitere Informationen zu CloudWatch Amazon-Dashboards finden Sie unter Verwenden von CloudWatch Amazon-Dashboards im CloudWatch Amazon-Benutzerhandbuch.

Wenn Sie das CloudWatch Amazon-Dashboard nicht erstellen möchten, müssen Sie die folgenden Schritte ausführen: Fügen Sie Ihrer Konfigurationsdatei zunächst einen `[dashboard]`Abschnitt

hinzu und fügen Sie dann den Namen dieses Abschnitts als Wert der `dashboard_settings` Einstellung in Ihrem `[cluster]`Abschnitt hinzu. Stellen Sie in Ihrem `[dashboard]`Bereich ein`enable = false`.

Wenn Ihr `[dashboard]`Abschnitt beispielsweise benannt ist `myDashboard` und Ihr `[cluster]`Abschnitt benannt ist`myCluster`, ähneln Ihre Änderungen diesem.

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

[dashboard MyDashboard]
enable = false
```

Integration mit Amazon CloudWatch Logs

Ab AWS ParallelCluster Version 2.6.0 werden allgemeine Protokolle standardmäßig in CloudWatch Logs gespeichert. Weitere Informationen zu CloudWatch Logs finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch. Informationen zur Konfiguration der CloudWatch Logs-Integration finden Sie im `[cw_log]`Abschnitt und in der `cw_log_settings` Einstellung.

Für jeden Cluster wird eine Protokollgruppe mit einem Namen erstellt `/aws/parallelcluster/`*`cluster-name`* (z. B.`/aws/parallelcluster/testCluster`). Jedes Protokoll (oder eine Gruppe von Protokollen, falls der Pfad eine enthält`*`) auf jedem Knoten hat einen Protokollstream mit dem Namen*`{hostname}`*`.`*`{instance_id}`*`.`*`{logIdentifier}`*. (Zum Beispiel`ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`.) Protokolldaten werden CloudWatch vom CloudWatch Agenten, der wie `root` auf allen Clusterinstanzen ausgeführt wird, gesendet.

Ab AWS ParallelCluster Version 2.10.0 wird bei der Erstellung des Clusters ein CloudWatch Amazon-Dashboard erstellt. Dieses Dashboard macht es einfach, die in CloudWatch Logs gespeicherten Protokolle zu überprüfen. Weitere Informationen finden Sie unter CloudWatch Amazon-Dashboard.

Diese Liste enthält den Pfad *`logIdentifier`* und für die Protokollstreams, die für Plattformen, Scheduler und Knoten verfügbar sind.

Log-Streams sind für Plattformen, Scheduler und Knoten verfügbar

Plattform en	Schedu s	Knoten	Protokollstreams
amazon CentOS ubuntu	awsbatc Slurm	HeadNc	DCV-Authentifikator: `/var/log/parallelcluster/pcluster_dcv_authenticator.log` dcv-ext-authenticator: `/var/log/parallelcluster/pcluster_dcv_connect.log` DCV-Agent: `/var/log/dcv/agent.*.log` DCV-Sitzung: `/var/log/dcv/dcv-xsession.*.log` DCV-Server: `/var/log/dcv/server.log` dcv-session-launcher: `/var/log/dcv/sessionlauncher.log` Xdcv: `/var/log/dcv/Xdcv.*.log` cfn-init: `/var/log/cfn-init.log` Chefkunde: `/var/log/chef-client.log`
amazon CentOS ubuntu	awsbatc Slurm	Comput eet HeadNc	Cloud-Einheit: `/var/log/cloud-init.log` beaufsichtigt: `/var/log/supervisord.log`
amazon CentOS ubuntu	Slurm	Comput eet	cloud-init-output: `/var/log/cloud-init-output.log` computemgtd: `/var/log/parallelcluster/computemgtd` verunglimpft: `/var/log/slurmd.log`
amazon CentOS ubuntu	Slurmer	HeadNc	clustermgtd: `/var/log/parallelcluster/clustermgtd` slurm_resume: `/var/log/parallelcluster/slurm_resume.log`

Plattform en	Schedu s	Knoten	Protokollstreams
			slurm_suspendieren: `/var/log/parallelcluster/slurm_suspend.log` slurmctld: `/var/log/slurmctld.log`
amazon CentOS	als Batch Slurm	Comput eet HeadNc	Systemnachrichten: `/var/log/messages`
ubuntu	awsbatc Slurm	Comput eet HeadNc	Syslog: `/var/log/syslog`

Jobs in Clustern, die verwenden, AWS Batch speichern die Ausgabe von Jobs, die den `FAILED` Status, oder erreicht haben `RUNNINGSUCCEEDED`, in CloudWatch Logs. Die Protokollgruppe ist`/aws/batch/job`, und das Namensformat für den Protokollstream ist*`jobDefinitionName`*`/default/`*`ecs_task_id`*. Standardmäßig laufen diese Protokolle nie ab, Sie können den Aufbewahrungszeitraum jedoch ändern. Weitere Informationen finden Sie unter Ändern der Aufbewahrung von Protokolldaten in CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

> **Note**
>
> `chef-client`,`cloud-init-output`, `clustermgtd` `computemgtd``slurm_resume`, und `slurm_suspend` wurden in AWS ParallelCluster Version 2.9.0 hinzugefügt. Für AWS ParallelCluster Version 2.6.0 wurden `/var/log/cfn-init-cmd.log` (`cfn-init-cmd`) und `/var/log/cfn-wire.log` (`cfn-wire`) auch in CloudWatch Logs gespeichert.

Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) ist ein Netzwerkgerät, das Betriebssystem-Bypass-Funktionen für die Netzwerkkommunikation mit geringer Latenz mit anderen Instances im selben Subnetz besitzt. EFA

wird mithilfe von Libfabric verfügbar gemacht und kann von Anwendungen verwendet werden, die die Messaging Passing Interface (MPI) verwenden.

Um EFA mit zu verwenden AWS ParallelCluster, fügen Sie die Zeile dem `[queue]`Abschnitt `enable_efa = true` hinzu.

Eine Liste der EC2 Instances, die EFA unterstützen, finden Sie unter Unterstützte Instance-Typen im EC2 Amazon-Benutzerhandbuch für Linux-Instances.

Weitere Informationen zu der `enable_efa` Einstellung finden Sie `enable_efa`im `[queue]`Abschnitt.

Latenzen zwischen Instances sollten mithilfe einer Cluster-Placement-Gruppe minimiert werden. Weitere Informationen erhalten Sie unter `placement` und `placement_group`.

Weitere Informationen finden Sie unter Elastic Fabric Adapter im EC2 Amazon-Benutzerhandbuch und Skalieren von HPC-Workloads mit Elastic Fabric Adapter und AWS ParallelCluster im AWS Open Source-Blog.

> **Note**
>
> Standardmäßig wird bei Ubuntu-Distributionen der ptrace-Schutz (Ablaufverfolgung) aktiviert. Ab AWS ParallelCluster 2,6.0 ist der ptrace-Schutz deaktiviert, damit Libfabric ordnungsgemäß funktioniert. Weitere Informationen finden Sie unter Deaktivieren des Ptrace-Schutzes im EC2 Amazon-Benutzerhandbuch.

> **Note**
>
> Die Support von EFA auf ARM-basierten Graviton2-Instances wurde in Version 2.10.1 hinzugefügt. AWS ParallelCluster

Intel Select Solutions

AWS ParallelCluster ist als Intel Select Solution für Simulation und Modellierung erhältlich. Die Konfigurationen entsprechen nachweislich den in der Intel HPC-Plattformspezifikation festgelegten Standards, verwenden spezifische Instance-Typen von Intel und sind für die Verwendung der Elastic

Fabric Adapter (EFA-) Netzwerkschnittstelle konfiguriert. AWS ParallelCluster ist die erste Cloud-Lösung, die die Anforderungen für das Intel Select Solutions Programm erfüllt. Zu den unterstützten Instanztypen gehören `c5n.18xlargem5n.24xlarge`, und`r5n.24xlarge`. Im Folgenden finden Sie eine Beispielkonfiguration, die mit dem Intel Select Solutions-Standard kompatibel ist.

Example Konfiguration von Intel Select Solutions

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

[aws]
aws_region_name = <Your AWS-Region>

[scaling demo]
scaledown_idletime = 5

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

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

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

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

```
[compute_resource m5n_i1]
instance_type = m5n.24xlarge
max_count = 5

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

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

Weitere Informationen zur Intel HPC-Plattformspezifikation AWS ParallelCluster und zur Intel HPC-Plattformspezifikation finden Sie unterIntel HPC-Plattform-Spezifikation.

Intel MPI aktivieren

Intel MPI ist verfügbar auf der. AWS ParallelCluster AMIs Um Intel MPI verwenden zu können, müssen Sie die Bedingungen der vereinfachten Softwarelizenz von Intel anerkennen und akzeptieren. Standardmäßig befindet sich Open MPI im Pfad. Um Intel MPI anstelle von Open MPI zu aktivieren, müssen Sie zuerst das Intel MPI-Modul laden. Anschließend müssen Sie die neueste Version installieren, indem Sie. `module load intelmpi` Der genaue Name des Moduls ändert sich mit jedem Update. Führen Sie `module avail` aus, um anzuzeigen, welche Module verfügbar sind. Die Ausgabe sieht wie folgt aus.

```
$ module avail

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

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

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

```
$ module load intelmpi
```

Führen Sie `module list` aus, um anzuzeigen, welche Module geladen werden.

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

Führen Sie `mpirun --version` aus, um zu überprüfen, ob Intel MPI aktiviert ist.

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

Nachdem das Intel MPI-Modul geladen wurde, werden mehrere Pfade geändert, damit die Intel MPI-Tools verwendet werden können. Um den Code auszuführen, der mit den Intel MPI-Tools kompiliert wurde, laden Sie zuerst das Intel MPI-Modul.

> **Note**
>
> Intel MPI ist nicht mit AWS Graviton-basierten Instances kompatibel.

> **Note**
>
> Vor AWS ParallelCluster Version 2.5.0 war Intel MPI AWS ParallelCluster AMIs in den Regionen China (Peking) und China (Ningxia) nicht verfügbar.

Intel HPC-Plattform-Spezifikation

AWS ParallelCluster entspricht der Intel HPC-Plattformspezifikation. Die Intel HPC-Plattformspezifikation enthält eine Reihe von Rechen-, Struktur-, Speicher- und Softwareanforderungen, um einen hohen Qualitäts- und Kompatibilitätsstandard mit HPC-Workloads zu erreichen. Weitere Informationen finden Sie unter Intel HPC-Plattformspezifikation und Anwendungen, die verifiziert wurden, dass sie mit der Intel HPC-Plattformspezifikation kompatibel sind.

Um die HPC-Plattformspezifikation von Intel erfüllen zu können, müssen die folgenden Anforderungen erfüllt sein:

- Das Betriebssystem muss CentOS 7 (`base_os = centos7`) sein.
- Der Instance-Typ für die Datenverarbeitungsknoten muss über eine Intel-CPU und mindestens 64 GB Arbeitsspeicher verfügen. Für die `c5` Familie der Instance-Typen bedeutet dies, dass der Instance-Typ mindestens a `c5.9xlarge` () `compute_instance_type = c5.9xlarge` sein muss.
- Der Hauptknoten muss über mindestens 200 GB Speicher verfügen.
- Dem Endbenutzer-Lizenzvertrag für Intel Parallel Studio muss zugestimmt werden (`enable_intel_hpc_platform = true`).
- Jeder Datenverarbeitungsknoten muss über mindestens 80 GB Speicherplatz verfügen (`compute_root_volume_size = 80`).

Der Speicher kann lokal oder in einem Netzwerk sein (NFS wird vom Hauptknoten aus gemeinsam genutzt, Amazon EBS oder FSx für Lustre) und kann gemeinsam genutzt werden.

ARM-Leistungsbibliotheken

Ab AWS ParallelCluster Version 2.10.1 sind Arm Performance Libraries AWS ParallelCluster AMIs für die `ubuntu2004` Werte for`alinux2`, `centos8`ubuntu1804, und für die `base_os` Einstellung verfügbar. Die Arm Performance Libraries bieten optimierte standardmäßige mathematische Kernbibliotheken für Hochleistungsrechneranwendungen auf Arm-Prozessoren. Um Arm Performance Libraries verwenden zu können, müssen Sie die Bedingungen der Arm Performance Libraries (kostenlose Version) — Endbenutzer-Lizenzvereinbarung anerkennen und akzeptieren. Weitere Informationen zu Arm Performance Libraries finden Sie unter Kostenlose Arm Performance Libraries.

Um Arm Performance Libraries zu aktivieren, müssen Sie zuerst das Modul Arm Performance Libraries laden. `Armpl-21.0.0`benötigt GCC-9.3 als Anforderung. Wenn Sie das `armpl/21.0.0` Modul laden, wird das `gcc/9.3` Modul auch geladen. Der genaue Name des Moduls ändert sich mit jedem Update. Führen Sie `module avail` aus, um anzuzeigen, welche Module verfügbar sind. Anschließend müssen Sie die neueste Version mithilfe von `module load armpl` .installieren. Die Ausgabe sieht wie folgt aus.

```
$ module avail
```

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

Führen Sie `module load` *`modulename`* aus, um ein Modul zu laden. Sie können dies dem Skript hinzufügen, mit dem `mpirun` ausgeführt wird.

```
$ module load armpl

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

Führen Sie `module list` aus, um anzuzeigen, welche Module geladen werden.

```
$ module list
Currently Loaded Modulefiles:
1) /opt/arm/armpl/21.0.0/modulefiles/armpl/gcc-9.3
2) /opt/arm/armpl/21.0.0/modulefiles/armpl/21.0.0_gcc-9.3
3) armpl/21.0.0
```

Führen Sie Beispieltests durch, um zu überprüfen, ob die Arm Performance Libraries aktiviert sind.

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

Nachdem das Modul Arm Performance Libraries geladen wurde, werden mehrere Pfade geändert, um die Arm Performance Libraries-Tools verwenden zu können. Um Code auszuführen, der mit den Arm Performance Libraries Tools kompiliert wurde, laden Sie zuerst das Arm Performance Libraries-Modul.

> **Note**
>
> AWS ParallelCluster Versionen zwischen 2.10.1 und 2.10.4 verwenden. `armpl/20.2.1`

Stellen Sie über Amazon DCV eine Connect zum Hauptknoten her

Amazon DCV ist eine Technologie zur Fernvisualisierung, mit der Benutzer eine sichere Verbindung zu grafikintensiven 3D-Anwendungen herstellen können, die auf einem Remote-Hochleistungsserver gehostet werden. Weitere Informationen finden Sie unter Amazon DCV.

Die Amazon DCV-Software wird automatisch auf dem Hauptknoten installiert, wenn `base_os = alinux2` `base_os = centos7`, `base_os = ubuntu1804` oder `base_os = ubuntu2004` verwendet wird.

Wenn der Hauptknoten eine ARM-Instance ist, wird die Amazon DCV-Software automatisch darauf installiert, wenn `base_os = alinux2` `base_os = centos7`, oder `base_os = ubuntu1804` verwendet wird.

Um Amazon DCV auf dem Hauptknoten zu aktivieren, `dcv_settings` muss es den Namen eines `[dcv]`Abschnitts enthalten, der über `alinux2` `centos7`ubuntu1804, oder ubuntu2004 verfügt `enable = master` und auf diesen gesetzt sein `base_os` muss. Wenn es sich bei dem Hauptknoten um eine ARM-Instance handelt, `base_os` muss er aufalinux2,centos7, oder ubuntu1804 gesetzt sein. Auf diese Weise wird der Cluster-Konfigurationsparameter `shared_dir` auf den DCV-Serverspeicherordner gesetzt. AWS ParallelCluster

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

Weitere Informationen zu den Amazon DCV-Konfigurationsparametern finden Sie unter`dcv_settings`. Verwenden Sie den `pcluster dcv` Befehl, um eine Verbindung zur Amazon DCV-Sitzung herzustellen.

> **Note**
>
> Die Support für Amazon DCV on `centos8` wurde in AWS ParallelCluster Version 2.10.4 entfernt. Support für Amazon DCV on `centos8` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt. Support für Amazon DCV auf AWS Graviton-basierten Instances wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt. Support für Amazon DCV ist aktiviert `alinux2` und `ubuntu1804` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt. Support für Amazon DCV on `centos7` wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

> **Note**
>
> Amazon DCV wird auf AWS Graviton-basierten Instances in den AWS ParallelCluster Versionen 2.8.0 und 2.8.1 nicht unterstützt.

Amazon DCV HTTPS-Zertifikat

Amazon DCV generiert automatisch ein selbstsigniertes Zertifikat, um den Verkehr zwischen dem Amazon DCV-Client und dem Amazon DCV-Server zu sichern.

Um das standardmäßige selbstsignierte Amazon DCV-Zertifikat durch ein anderes Zertifikat zu ersetzen, stellen Sie zunächst eine Verbindung zum Hauptknoten her. Kopieren Sie dann das Zertifikat und den Schlüssel in den `/etc/dcv`-Ordner, bevor Sie den `pcluster dcv`-Befehl ausführen.

Weitere Informationen finden Sie unter Ändern des TLS-Zertifikats im Amazon DCV-Administratorhandbuch.

Lizenzierung von Amazon DCV

Der Amazon DCV-Server benötigt keinen Lizenzserver, wenn er auf EC2 Amazon-Instances ausgeführt wird. Der Amazon DCV-Server muss jedoch regelmäßig eine Verbindung zu einem Amazon S3 S3-Bucket herstellen, um festzustellen, ob eine gültige Lizenz verfügbar ist.

AWS ParallelCluster fügt dem automatisch die erforderlichen Berechtigungen hinzu. `ParallelClusterInstancePolicy` Wenn Sie eine benutzerdefinierte IAM-Instance-Richtlinie verwenden, verwenden Sie die in Amazon DCV on Amazon EC2 im Amazon DCV-Administratorhandbuch beschriebenen Berechtigungen.

Tipps zur Fehlerbehebung finden Sie unter Behebung von Problemen in Amazon DCV.

Verwenden von `pcluster update`

`pcluster update`Analysiert ab AWS ParallelCluster Version 2.8.0 die Einstellungen, die zur Erstellung des aktuellen Clusters verwendet wurden, und die Einstellungen in der Konfigurationsdatei auf Probleme. Wenn Probleme entdeckt werden, werden sie gemeldet, und es werden die Schritte zur Behebung der Probleme angezeigt. Wenn die `compute_instance_type` Einstellung beispielsweise auf einen anderen Instance-Typ geändert wird, muss die Rechenflotte gestoppt werden, bevor ein Update durchgeführt werden kann. Dieses Problem wird gemeldet, wenn es entdeckt wird. Wenn keine Blockierungsprobleme gemeldet wurden, werden Sie gefragt, ob Sie die Änderungen übernehmen möchten.

In der Dokumentation für jede Einstellung wird die Aktualisierungsrichtlinie für diese Einstellung definiert.

Aktualisierungsrichtlinie: Diese Einstellungen können während eines Updates geändert werden., Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

> Diese Einstellungen können geändert werden, und der Cluster kann mit aktualisiert werden`pcluster update`.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

> Diese Einstellungen können nicht geändert werden, wenn der vorhandene Cluster nicht gelöscht wurde. Entweder muss die Änderung rückgängig gemacht werden oder der Cluster muss gelöscht (mit`pcluster delete`) werden, und dann muss an der Stelle des alten Clusters ein neuer Cluster erstellt werden (mit`pcluster create`).

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

> Diese Einstellungen können geändert und der Cluster aktualisiert werden mit`pcluster update`.

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

> Diese Einstellungen können nicht geändert werden, solange die Compute-Flotte existiert. Entweder muss die Änderung rückgängig gemacht werden oder die Rechenflotte muss gestoppt (verwendet`pcluster stop`), aktualisiert (verwendet`pcluster update`) und dann eine neue Rechenflotte erstellt (mit`pcluster start`) werden.

Aktualisierungsrichtlinie: Diese Einstellung kann während eines Updates nicht verringert werden.

> Diese Einstellungen können geändert, aber nicht verringert werden. Wenn diese Einstellungen verringert werden müssen, müssen Sie den Cluster löschen (using`pcluster delete`) und einen neuen Cluster erstellen (using`pcluster create`).

Richtlinie aktualisieren: Um die Größe einer Warteschlange unter die aktuelle Anzahl von Knoten zu reduzieren, muss zuerst die Rechenflotte gestoppt werden.

> Diese Einstellungen können geändert werden, aber wenn die Änderung die Größe der Warteschlange unter die aktuelle Größe reduzieren würde, muss die Rechenflotte gestoppt (verwendet`pcluster stop`), aktualisiert (verwendet`pcluster update`) und dann eine neue Rechenflotte erstellt (mit`pcluster start`) werden.

Richtlinie aktualisieren: Um die Anzahl der statischen Knoten in einer Warteschlange zu reduzieren, muss zuerst die Rechenflotte gestoppt werden.

> Diese Einstellungen können geändert werden, aber wenn durch die Änderung die Anzahl der statischen Knoten in der Warteschlange unter die aktuelle Größe sinken würde, muss die Rechenflotte gestoppt (verwendet`pcluster stop`), aktualisiert (verwendet`pcluster update`) und dann eine neue Rechenflotte erstellt (mit`pcluster start`) werden.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig. Die Aktualisierung dieser Einstellung kann nicht erzwungen werden.

> Diese Einstellungen können nicht geändert werden, wenn der vorhandene Cluster nicht gelöscht wurde. Entweder muss die Änderung rückgängig gemacht werden oder der Cluster muss gelöscht (mit`pcluster delete`) werden, und dann muss an der Stelle des alten Clusters ein neuer Cluster erstellt werden (mit`pcluster create`).

Aktualisierungsrichtlinie: Wenn AWS ParallelCluster verwaltete Amazon FSx for Lustre-Dateisysteme in der Konfiguration nicht angegeben sind, kann diese Einstellung während eines Updates geändert werden.

> Diese Einstellung kann geändert werden, wenn sie `[cluster]fsx_settings`nicht angegeben ist oder wenn `fsx_settings` sowohl als auch `fsx-fs-id`in angegeben `[fsx fs]`sind, um ein vorhandenes externes Dateisystem FSx für Lustre zu mounten.

Dieses Beispiel zeigt a `pcluster update` mit einigen Änderungen, die das Update blockieren.

```
$ pcluster update
  Validating configuration file /home/username/.parallelcluster/config...
```

```
Retrieving configuration from CloudFormation for cluster test-1...
Found Changes:

#   section/parameter         old value                 new value
--  ------------------------  ------------------------  ------------------------
    [cluster default]
01* compute_instance_type     t2.micro                  c4.xlarge
02* ebs_settings              ebs2                      -

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

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

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

#01
Compute fleet must be empty to update "compute_instance_type"
How to fix:
Make sure that there are no jobs running, then run the following command:
  pcluster stop -c $CONFIG_FILE $CLUSTER_NAME

#02
Cannot add/remove EBS Sections
How to fix:
Revert "ebs_settings" value to "ebs2"

#04
Cannot change the mount dir of an existing EBS volume
How to fix:
Revert "my/very/very/long/shared/dir" to "shared"

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

Update aborted.
```

AMI-Patching und EC2 Instanzersatz

Um sicherzustellen, dass sich alle dynamisch gestarteten Cluster-Rechenknoten konsistent verhalten, werden automatische Betriebssystemupdates für Cluster-Instances AWS ParallelCluster deaktiviert. Darüber hinaus wird für jede Version von AWS ParallelCluster AMIs AWS ParallelCluster und der zugehörigen CLI ein bestimmter Satz von erstellt. Dieser spezifische Satz von AMIs bleibt unverändert und wird nur von der AWS ParallelCluster Version unterstützt, für die sie erstellt wurden. AWS ParallelCluster AMIsfür veröffentlichte Versionen, die nicht aktualisiert wurden.

Aufgrund aufkommender Sicherheitsprobleme möchten Kunden jedoch möglicherweise Patches zu diesen hinzufügen AMIs und dann ihre Cluster mit dem gepatchten AMI aktualisieren. Dies entspricht dem Modell der AWS ParallelCluster gemeinsamen Verantwortung.

Um den spezifischen Satz von zu sehen, der von der AWS ParallelCluster CLI-Version AWS ParallelCluster AMIs unterstützt wird, die Sie gerade verwenden, führen Sie Folgendes aus:

```
$ pcluster version
```

Sehen Sie sich dann amis.txt im AWS ParallelCluster GitHub Repository an.

Der AWS ParallelCluster Hauptknoten ist eine statische Instanz und Sie können ihn manuell aktualisieren. Der Neustart und der Neustart des Hauptknotens werden ab AWS ParallelCluster Version 2.11 vollständig unterstützt, wenn der Instanztyp keinen Instanzspeicher hat. Weitere Informationen finden Sie unter Instance-Typen mit Instance-Speicher-Volumes im EC2 Amazon-Benutzerhandbuch für Linux-Instances. Sie können ein AMI für einen vorhandenen Cluster nicht aktualisieren.

Der Neustart des Hauptknotens und der Neustart mit AMI-Updates von Cluster-Compute-Instances werden ab AWS ParallelCluster Version 3.0.0 vollständig unterstützt. Erwägen Sie ein Upgrade auf die neueste Version, um diese Funktionen nutzen zu können.

Aktualisierung oder Austausch der Headnode-Instanz

Unter bestimmten Umständen müssen Sie den Hauptknoten möglicherweise neu starten oder neu starten. Dies ist beispielsweise erforderlich, wenn Sie das Betriebssystem manuell aktualisieren oder wenn eine geplante Außerbetriebnahme einer AWS Instanz stattfindet, die einen Neustart der Hauptknoteninstanz erfordert.

Wenn Ihre Instance nicht über kurzlebige Laufwerke verfügt, können Sie sie jederzeit beenden und erneut starten. Im Falle einer geplanten Außerbetriebnahme wird beim Starten der gestoppten Instance diese migriert, sodass sie die neue Hardware verwendet.

Ebenso können Sie eine Instance, die keine Instance-Speicher hat, manuell stoppen und starten. Fahren Sie in diesem Fall und in anderen Fällen von Instances ohne ephemere Volumes fort. Stoppen und starten Sie den Hauptknoten eines Clusters

Wenn Ihre Instance über kurzlebige Laufwerke verfügt und diese gestoppt wurde, gehen die Daten im Instance-Speicher verloren. Sie können anhand der Tabelle unter Instance-Speicher-Volumes ermitteln, ob der für den Head-Knoten verwendete Instance-Typ Instance-Speicher enthält.

In den folgenden Abschnitten werden die Einschränkungen bei der Verwendung von Instances mit Instance-Speicher-Volumes beschrieben.

Einschränkungen des Instance-Speichers

Bei der Verwendung von AWS ParallelCluster Version 2.11 und Instance-Typen mit einem Instance-Speicher gelten folgende Einschränkungen:

- Wenn kurzlebige Laufwerke nicht verschlüsselt sind (der `encrypted_ephemeral`Parameter ist auf `false` oder nicht gesetzt), kann eine AWS ParallelCluster Instanz nach dem Stoppen einer Instanz nicht mehr gestartet werden. Das liegt daran, dass Informationen über alte, nicht existierende Ephemerals in das Betriebssystem geschrieben werden `fstab` und das Betriebssystem versucht, nicht vorhandenen Speicher bereitzustellen.
- Wenn kurzlebige Laufwerke verschlüsselt sind (der `encrypted_ephemeral`Parameter ist auf gesetzt`true`), kann eine AWS ParallelCluster Instanz nach einem Stopp gestartet werden, aber die neuen kurzlebigen Laufwerke sind nicht eingerichtet, gemountet oder verfügbar.
- Wenn ephemere Laufwerke verschlüsselt sind, kann eine AWS ParallelCluster Instanz neu gestartet werden, aber auf alte ephemere Laufwerke (die den Instance-Neustart überstehen) kann nicht zugegriffen werden, da der Verschlüsselungsschlüssel in dem Speicher erstellt wird, der beim Neustart verloren geht.

Der einzige unterstützte Fall ist der Instance-Neustart, bei dem kurzlebige Laufwerke nicht verschlüsselt sind. Das liegt daran, dass das Laufwerk den Neustart übersteht und aufgrund des eingegebenen Eintrags wieder gemountet wird. `fstab`

Behelfslösungen für Einschränkungen beim Instanzspeicher

Speichern Sie zunächst Ihre Daten. Um zu überprüfen, ob Sie Daten haben, die aufbewahrt werden müssen, sehen Sie sich den Inhalt des `ephemeral_dir` Ordners an (`/scratch`standardmäßig). Sie können die Daten auf das Root-Volume oder die an den Cluster angeschlossenen gemeinsam genutzten Speichersysteme wie Amazon FSx, Amazon EFS oder Amazon EBS übertragen. Beachten Sie, dass für die Datenübertragung in den Remotespeicher zusätzliche Kosten anfallen können.

Die Hauptursache für die Einschränkungen liegt in der Logik, die zum Formatieren und Mounten von Instance-Speicher-Volumes AWS ParallelCluster verwendet wird. Die Logik fügt einen Eintrag in `/etc/fstab` der folgenden Form hinzu:

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

`${ephemeral_dir}`ist der Wert des `ephemeral_dir` Parameters aus der Pcluster-Konfigurationsdatei (standardmäßig). `/scratch`

Diese Zeile wird hinzugefügt, sodass die Instance-Speicher-Volumes automatisch neu gemountet werden, falls oder wenn ein Knoten neu gestartet wird. Dies ist wünschenswert, da Daten auf kurzlebigen Laufwerken auch nach einem Neustart erhalten bleiben. Daten auf den kurzlebigen Laufwerken bleiben jedoch nicht während eines Start- oder Stoppzyklus erhalten. Das bedeutet, dass sie ohne Daten formatiert und bereitgestellt werden.

Der einzige unterstützte Fall ist der Instanzneustart, wenn kurzlebige Laufwerke nicht verschlüsselt sind. Das liegt daran, dass das Laufwerk den Neustart übersteht und wieder eingehängt wird, weil es eingeschrieben ist. `fstab`

Um die Daten in allen anderen Fällen beizubehalten, müssen Sie den Eintrag für das logische Volume entfernen, bevor Sie die Instanz beenden. Entfernen Sie beispielsweise `/dev/vg.01/lv_ephemeral` von, `/etc/fstab` bevor Sie die Instanz beenden. Danach starten Sie die Instanz, ohne die kurzlebigen Volumes zu mounten. Der Instance-Speicher ist jedoch nach dem Stoppen oder Starten der Instance wieder nicht verfügbar.

Nachdem Sie Ihre Daten gespeichert und den `fstab` Eintrag anschließend entfernt haben, fahren Sie mit dem nächsten Abschnitt fort.

Stoppen und starten Sie den Hauptknoten eines Clusters

> **Note**
>
> Ab AWS ParallelCluster Version 2.11 wird das Stoppen und Starten des Kopfknotens nur unterstützt, wenn der Instanztyp keinen Instanzspeicher hat.

1. Stellen Sie sicher, dass im Cluster keine laufenden Jobs vorhanden sind.

   Wenn Sie einen Slurm Scheduler verwenden:

   - Wenn die `sbatch --no-requeue` Option nicht angegeben ist, werden laufende Jobs in die Warteschlange gestellt.
   - Wenn die `--no-requeue` Option angegeben ist, schlagen laufende Jobs fehl.

2. Beantragen Sie einen Stopp der Cluster-Compute-Flotte:

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

3. Warten Sie, bis der Status der Compute-Flotte wie folgt lautet`STOPPED`:

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

4. Für manuelle Updates mit einem Neustart des Betriebssystems oder einer Instanz können Sie das AWS Management Console oder verwenden AWS CLI. Im Folgenden finden Sie ein Beispiel für die Verwendung von AWS CLI.

```
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
  "StoppingInstances": [
    {
```

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

5. Starten Sie die Rechenflotte des Clusters:

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

AWS ParallelCluster CLI-Befehle

`pcluster`und `pcluster-config` sind die AWS ParallelCluster CLI-Befehle. Sie verwenden `pcluster` sie, um HPC-Cluster im zu starten und zu verwalten AWS Cloud und Ihre Konfiguration `pcluster-config` zu aktualisieren.

Für die Verwendung `pcluster` benötigen Sie eine IAM-Rolle mit den für die Ausführung erforderlichen Berechtigungen.

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

Themen

- pcluster
- pcluster-config

`pcluster`

`pcluster`ist der primäre AWS ParallelCluster CLI-Befehl. Sie verwenden`pcluster`, um HPC-Cluster in der zu starten und zu verwalten. AWS Cloud

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

Argumente

`pcluster` *`command`*

Mögliche Optionen: `configure`, `create`, `createami`, `dcv`, `delete`, `instances`, `list`, `ssh`, `start`, `status`, `stop`, `update`, `version`

Unterbefehle:

Themen

- pcluster configure
- pcluster create
- pcluster createami
- pcluster dcv
- pcluster delete
- pcluster instances
- pcluster list
- pcluster ssh
- pcluster start
- pcluster status
- pcluster stop
- pcluster update
- pcluster version

pcluster configure

Beginnt mit einer AWS ParallelCluster Konfiguration. Weitere Informationen finden Sie unter Konfiguration AWS ParallelCluster.

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

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster configure`.

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

Gibt den vollständigen Pfad der zu verwendenden alternativen Konfigurationsdatei an.

Standardeinstellung: `~/.parallelcluster/config`.

Weitere Informationen finden Sie unter Konfiguration AWS ParallelCluster.

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

Gibt den AWS-Region zu verwendenden an. Wenn dies angegeben ist, überspringt die Konfiguration die AWS-Region Erkennung.

Um die Netzwerkressourcen in der VPC zu löschen, können Sie den CloudFormation Netzwerkstapel löschen. Der Stack-Name beginnt mit“parallelclusternetworking-"und enthält die Erstellungszeit im Format „YYYYMMDDHHMMSS“. Sie können die Stapel mit dem Befehl list-stacks auflisten.

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

Der Stapel kann mit dem Befehl delete-stack gelöscht werden.

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

Die VPC, die für Sie `pcluster configure` erstellt, wird nicht im CloudFormation Netzwerk-Stack erstellt. Sie können diese VPC manuell in der Konsole oder mithilfe von löschen. AWS CLI

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

pcluster create

Erstellt einen neuen Cluster.

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

Positionale Argumente

cluster_name

Definiert den Namen des Clusters. Der AWS CloudFormation Stack-Name lautet`parallelcluster-`*`cluster_name`*.

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster create`.

`-c CONFIG_FILE, --config CONFIG_FILE`

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

`-r REGION, --region REGION`

Gibt den AWS-Region zu verwendenden an. Die Prioritätsreihenfolge, die AWS-Region für die Auswahl von für einen neuen Cluster verwendet wird, lautet wie folgt:

1. `-r`oder `--region` Parameter für`pcluster create`.
2. `AWS_DEFAULT_REGION`-Umgebungsvariable.
3. `aws_region_name`Einstellung im `[aws]` Abschnitt der AWS ParallelCluster Konfigurationsdatei (der Standardspeicherort ist`~/.parallelcluster/config`.) Dies ist der Speicherort, der durch den `pcluster configure` Befehl aktualisiert wurde.
4. `region`Einstellung im `[default]` Abschnitt der AWS CLI Konfigurationsdatei (`~/.aws/config`.)

`-nw, --nowait`

Zeigt an, dass nach der Ausführung eines Stack-Befehls nicht auf Stack-Ereignisse gewartet werden soll.

Standardeinstellung: `False`.

`-nr, --norollback`

Deaktiviert -Stack-Rollback bei Fehler.

Standardeinstellung: `False`.

`-u TEMPLATE_URL, --template-url TEMPLATE_URL`

Gibt eine URL für die benutzerdefinierte AWS CloudFormation Vorlage an, falls sie bei der Erstellung verwendet wurde.

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

Zeigt an, welche Cluster-Vorlage verwendet werden soll.

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

Fügt zusätzliche Parameter zur Stack-Erstellung hinzu.

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

Gibt zusätzliche Tags an, die dem Stack hinzugefügt werden sollen.

Wenn der Befehl aufgerufen wird und mit der Abfrage des Status dieses Aufrufs beginnt, können Sie den Vorgang problemlos mit „Strg-C“ beenden. Sie können den aktuellen Status wieder anzeigen, indem Sie „`pcluster status mycluster`“ aufrufen.

Beispiele mit Version 2.11.7 AWS ParallelCluster :

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

pcluster createami

(Linux/macOS) Erstellt ein benutzerdefiniertes AMI zur Verwendung mit. AWS ParallelCluster

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

Erforderliche Abhängigkeiten

Zusätzlich zur AWS ParallelCluster CLI ist für die Ausführung die folgende Abhängigkeit erforderlich`pcluster createami`:

- Packer: Laden Sie die neueste Version von https://developer.hashicorp.com/packer/downloads herunter.

> **Note**
>
> Vor AWS ParallelCluster Version 2.8.0 musste Berkshelf (installiert mit`gem install berkshelf`) verwendet werden. `pcluster createami`

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster createami`.

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

Gibt das Basis-AMI an, das für die Erstellung des AWS ParallelCluster AMI verwendet werden soll.

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

Gibt das Betriebssystem des Basis-AMI an. Gültige Optionen sind `alinux2`, `ubuntu1804`, `ubuntu2004` und `centos7`.

> **Note**
>
> Änderungen der Betriebssystemunterstützung in verschiedenen AWS ParallelCluster Versionen:
>
> - Die Support für `centos8` wurde in AWS ParallelCluster Version 2.10.4 entfernt.
> - Support für `centos8` wurde hinzugefügt und Unterstützung für `centos6` wurde in AWS ParallelCluster Version 2.10.0 entfernt.
> - Support für alinux2 wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.
> - Support für ubuntu1804 wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

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

Gibt den Instance-Typ an, der zur Erstellung des AMI verwendet werden soll.

Standardeinstellung: `t2.xlarge`.

> **Note**
>
> Support für das `--instance-type` Argument wurde in AWS ParallelCluster Version 2.4.1 hinzugefügt.

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

Gibt den Präfixnamen des resultierenden AWS ParallelCluster AMI an.

Standardeinstellung: `custom-ami-`.

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

Gibt das Kochbuch an, das zum Erstellen des AWS ParallelCluster AMI verwendet werden soll.

`--post-install` *`POST_INSTALL_SCRIPT`*

Gibt den Pfad zum Post-Installationsskript an. Pfade müssen ein `s3://``https://`, oder `file://` URL-Schema verwenden. Beispiele sind unter anderem:

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

> **Note**
>
> Support für das `--post-install` Argument wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

`--no-public-ip`

Ordnen Sie der Instance, die zur Erstellung des AMI verwendet wurde, keine öffentliche IP-Adresse zu. Standardmäßig ist der Instance eine öffentliche IP-Adresse zugeordnet.

> **Note**
>
> Support für das `--no-public-ip` Argument wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

`-c CONFIG_FILE, --config CONFIG_FILE`

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

`-t CLUSTER_TEMPLATE, --cluster-template CLUSTER_TEMPLATE`

Gibt den Abschnitt `[cluster]` von an*`CONFIG_FILE`*, der zum Abrufen der VPC- und Subnetzeinstellungen verwendet werden soll.

> **Note**
>
> Support für das `--cluster-template` Argument wurde in AWS ParallelCluster Version 2.4.0 hinzugefügt.

`--vpc-id VPC_ID`

Gibt die ID der VPC an, die zum Erstellen des AWS ParallelCluster AMI verwendet werden soll.

> **Note**
>
> Support für das `--vpc-id` Argument wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

`--subnet-id SUBNET_ID`

Gibt die ID des Subnetzes an, das zum Erstellen des AWS ParallelCluster AMI verwendet werden soll.

> **Note**
>
> Support für das `--vpc-id` Argument wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

`-r REGION, --region REGION`

Gibt an, was verwendet AWS-Region werden soll. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

pcluster dcv

Interagiert mit dem Amazon DCV-Server, der auf dem Hauptknoten ausgeführt wird.

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

pcluster dcv *command*

Mögliche Optionen: `connect`

> **Note**
>
> Änderungen der Betriebssystemunterstützung für den `pcluster dcv` Befehl in verschiedenen AWS ParallelCluster Versionen:
>
> - Support für den `pcluster dcv` Befehl on `centos8` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.
> - Support für den `pcluster dcv` Befehl auf AWS Graviton-basierten Instanzen wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt.
> - Support für den `pcluster dcv` Befehl on `ubuntu1804` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.
> - Support für den `pcluster dcv` Befehl on `centos7` wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster dcv`.

Unterbefehle:

pcluster dcv connect

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

> **Important**
>
> Die URL läuft 30 Sekunden nach ihrer Ausgabe ab. Wenn die Verbindung nicht hergestellt wird, bevor die URL abläuft, führen Sie den Vorgang `pcluster dcv connect` erneut aus, um eine neue URL zu generieren.

Positionale Argumente

cluster_name

Gibt den Namen des Clusters an, mit dem eine Verbindung hergestellt werden soll.

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster dcv connect`.

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

Schlüsselpfad des SSH-Schlüssels, der für die Verbindung verwendet werden soll.

Der Schlüssel muss der zum Zeitpunkt der Cluster-Erstellung im `key_name`-Konfigurationsparameter angegebene sein. Dieses Argument ist optional, aber wenn es nicht angegeben ist, muss der Schlüssel standardmäßig für den SSH-Client verfügbar sein. Fügen Sie es beispielsweise dem `ssh-agent` mit `ssh-add` hinzu.

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

Gibt an, was verwendet AWS-Region werden soll. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

`-s, --show-url`

Zeigt eine einmalige URL für die Verbindung mit der Amazon DCV-Sitzung an. Der Standardbrowser wird nicht geöffnet, wenn diese Option angegeben ist.

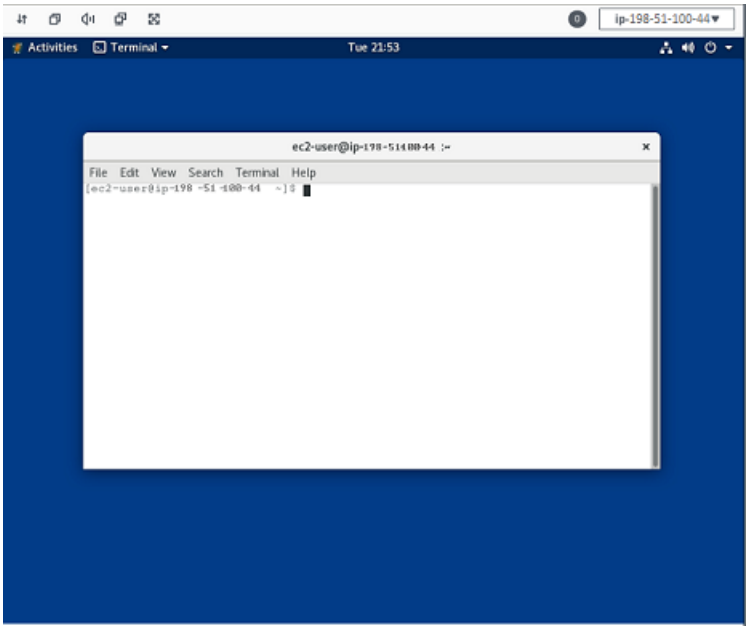
> **Note**
>
> Support für das `--show-url` Argument wurde in AWS ParallelCluster Version 2.5.1 hinzugefügt.

Beispiel mit AWS ParallelCluster Version 2.11.7:

```
$ pcluster dcv connect -k ~/.ssh/id_rsa mycluster
```

Öffnet den Standardbrowser, um eine Verbindung zu der Amazon DCV-Sitzung herzustellen, die auf dem Hauptknoten ausgeführt wird.

Eine neue Amazon DCV-Sitzung wird erstellt, falls noch keine gestartet wurde.



pcluster delete

Löscht einen Cluster.

```
pcluster delete [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] cluster_name
```

Positionale Argumente

cluster_name

Gibt den Namen des Clusters an, der gelöscht werden soll.

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster delete`.

`-c CONFIG_FILE, --config CONFIG_FILE`

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

`--keep-logs`

Behalten Sie die CloudWatch Protokolldaten nach dem Löschen des Clusters bei. Die Protokollgruppe bleibt bestehen, bis Sie sie manuell löschen, aber die Protokollereignisse laufen je nach `retention_days` Einstellung ab. Standardmäßig ist die Einstellung auf 14 Tage festgelegt.

> **Note**
>
> Support für das **`--keep-logs`** Argument wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

`-r REGION, --region REGION`

Gibt an, was verwendet AWS-Region werden soll. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

Wenn der Befehl aufgerufen wird und mit der Abfrage des Status dieses Aufrufs beginnt, können Sie den Vorgang mit „Strg-C“ beenden. Sie können den aktuellen Status wieder anzeigen, indem Sie „`pcluster status mycluster`“ aufrufen.

Beispiel mit Version 2.11.7 AWS ParallelCluster :

```
$ pcluster delete -c path/to/config -r us-east-1 mycluster
Deleting: mycluster
Status: RootRole - DELETE_COMPLETE
Cluster deleted successfully.
```

Um die Netzwerkressourcen in der VPC zu löschen, können Sie den CloudFormation Netzwerkstapel löschen. Der Stack-Name beginnt mit“parallelclusternetworking-"und enthält die Erstellungszeit im Format „YYYYMMDDHHMMSS“. Sie können die Stapel mit dem Befehl list-stacks auflisten.

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

Der Stapel kann mit dem Befehl delete-stack gelöscht werden.

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

Die VPC, die für Sie `pcluster configure` erstellt, wird nicht im CloudFormation Netzwerk-Stack erstellt. Sie können diese VPC manuell in der Konsole oder mithilfe von löschen. AWS CLI

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

pcluster instances

Zeigt eine Liste aller Instances in einem Cluster an.

```
pcluster instances [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

Positionale Argumente

cluster_name

Zeigt die Instances für den Cluster mit dem angegebenen Namen an.

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster instances`.

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

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

`-r REGION, --region REGION`

Gibt den AWS-Region zu verwendenden an. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

Beispiel mit AWS ParallelCluster Version 2.11.7:

```
$ pcluster instances -c path/to/config -r us-east-1 mycluster
MasterServer         i-1234567890abcdef0
ComputeFleet         i-abcdef01234567890
```

pcluster list

Zeigt eine Liste der Stapel an, die verknüpft sind mit. AWS ParallelCluster

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

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster list`.

`--color`

Zeigt den Cluster-Status in Farbe an.

Standardeinstellung: `False`.

`-c CONFIG_FILE, --config CONFIG_FILE`

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `c`.

`-r REGION, --region REGION`

Gibt den AWS-Region zu verwendenden an. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

Listet die Namen aller benannten AWS CloudFormation Stapel auf. `parallelcluster-*`

Beispiel mit AWS ParallelCluster Version 2.11.7:

```
$ pcluster list -c path/to/config -r us-east-1
mycluster            CREATE_IN_PROGRESS  2.11.7
myothercluster       CREATE_IN_PROGRESS  2.11.7
```

pcluster ssh

Führt einen `ssh`-Befehl mit dem bereits ausgefüllten Benutzernamen und der IP-Adresse des Clusters aus. Beliebige Argumente werden am Ende des `ssh`-Befehls angefügt. Dieser Befehl kann im Aliasnamen-Abschnitt der Konfigurationsdatei angepasst werden.

```
pcluster ssh [ -h ] [ -d ] [ -r REGION ] cluster_name
```

Positionale Argumente

cluster_name

Gibt den Namen des Clusters an, mit dem eine Verbindung hergestellt werden soll.

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster ssh`.

-d, --dryrun

Drucken Sie den Befehl aus, der ausgeführt werden würde, und beenden Sie den Vorgang.

Standardeinstellung: `False`.

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

Gibt den AWS-Region zu verwendenden an. Standardmäßig wird die Region verwendet, die mit dem Befehl `pcluster configure` angegeben wird.

Beispiele mit AWS ParallelCluster Version 2.11.7:

```
$ pcluster ssh -d mycluster -i ~/.ssh/id_rsa
SSH command: ssh ec2-user@1.1.1.1 -i /home/user/.ssh/id_rsa
```

```
$ pcluster ssh mycluster -i ~/.ssh/id_rsa
```

Führt den `ssh` Befehl aus, wobei der Benutzername und die IP-Adresse des Clusters vorausgefüllt sind:

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

Der `ssh`-Befehl wird in der globalen Konfigurationsdatei unter `[aliases]` Abschnitt definiert. Er kann wie folgt angepasst werden.

```
[ aliases ]
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

Ersetzte Variablen:

`CFN_USER`

Der Benutzername für das `base_os`, das ausgewählt ist.

`MASTER_IP`

Die IP-Adresse des Hauptknotens.

`ARGS`

Optionale Argumente, die an den `ssh`-Befehl übergeben werden sollen.

pcluster start

Startet die Datenverarbeitungsflotte für einen Cluster, der gestoppt wurde.

```
pcluster start [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

Positionale Argumente

cluster_name

Startet die Datenverarbeitungsflotte des angegebenen Cluster-Namens

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster start`.

`-c CONFIG_FILE, --config CONFIG_FILE`

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

`-r REGION, --region REGION`

Gibt den AWS-Region zu verwendenden an. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

Beispiel mit AWS ParallelCluster Version 2.11.7:

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

Mit diesem Befehl wird für die Parameter der Auto Scaling-Gruppe einer der folgenden Werte festgelegt:

- Die anfänglichen Konfigurationswerte (`max_queue_size` und `initial_queue_size`) aus der Vorlage, die zum Erstellen des Clusters verwendet wurde
- Die Konfigurationswerte, die zum Aktualisieren des Clusters verwendet wurden, seit er erstellt wurde.

`pcluster status`

Fragt den aktuellen Status des Clusters ab.

```
pcluster status [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] [ -nw ] cluster_name
```

Positionale Argumente

`cluster_name`

Zeigt den Status des Clusters mit dem angegebenen Namen an.

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster status`.

`-c CONFIG_FILE, --config CONFIG_FILE`

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

`-r REGION, --region REGION`

Gibt den AWS-Region zu verwendenden an. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

`-nw, --nowait`

Zeigt an, dass nach der Verarbeitung eines Stack-Befehls nicht auf Stack-Ereignisse gewartet werden soll.

Standardeinstellung: `False`.

Beispiel mit AWS ParallelCluster Version 2.11.7:

```
$ pcluster status -c path/to/config -r us-east-1 mycluster
Status: ComputeFleetHITSubstack - CREATE_IN_PROGRESS
```

pcluster stop

Stoppt die Rechenflotte und lässt den Hauptknoten laufen.

```
pcluster stop [ -h ] [ -c CONFIG_FILE ] [ -r REGION ] cluster_name
```

Positionale Argumente

cluster_name

Stoppt die Datenverarbeitungsflotte des angegebenen Cluster-Namens.

Beispiel mit AWS ParallelCluster Version 2.11.7:

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster stop`.

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

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

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

Gibt den AWS-Region zu verwendenden an. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

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

Setzt die Auto Scaling Scaling-Gruppenparameter auf min/max/desired = 0/0/0 und beendet die Rechenflotte. Der Kopf läuft weiter. Um alle EC2 Ressourcen zu beenden und EC2 Gebühren zu vermeiden, sollten Sie erwägen, den Cluster zu löschen.

pcluster update

Analysiert die Konfigurationsdatei, um festzustellen, ob der Cluster sicher aktualisiert werden kann. Wenn die Analyse ergibt, dass der Cluster aktualisiert werden kann, werden Sie aufgefordert, die Änderung zu bestätigen. Wenn die Analyse ergibt, dass der Cluster nicht aktualisiert werden kann, werden die Konfigurationseinstellungen, die die Ursache der Konflikte sind, mit Details aufgelistet. Weitere Informationen finden Sie unter Verwenden von `pcluster update`.

```
pcluster update [ -h ] [ -c CONFIG_FILE ] [ --force ] [ -r REGION ] [ -nr ]
                [ -nw ] [ -t CLUSTER_TEMPLATE ] [ -p EXTRA_PARAMETERS ] [ -rd ]
                [ --yes ] cluster_name
```

Positionale Argumente

cluster_name

Gibt den Namen des Clusters an, der aktualisiert werden soll.

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster update`.

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

Gibt die alternative Konfigurationsdatei an, die verwendet werden soll.

Standardeinstellung: `~/.parallelcluster/config`.

--force

Aktiviert ein Update auch dann, wenn eine oder mehrere Einstellungen eine blockierende Änderung enthalten oder wenn eine ausstehende Aktion erforderlich ist (z. B. das Stoppen der Rechenflotte), bevor das Update fortgesetzt werden kann. Dies sollte nicht mit dem `--yes` Argument kombiniert werden.

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

Gibt an AWS-Region , was verwendet werden soll. Standardmäßig wird der mit dem `pcluster configure` Befehl AWS-Region angegebene Wert verwendet.

-nr, --norollback

Deaktiviert das AWS CloudFormation Stack-Rollback bei einem Fehler.

Standardeinstellung: `False`.

-nw, --nowait

Zeigt an, dass nach der Verarbeitung eines Stack-Befehls nicht auf Stack-Ereignisse gewartet werden soll.

Standardeinstellung: `False`.

`-t CLUSTER_TEMPLATE, --cluster-template CLUSTER_TEMPLATE`

Gibt den Abschnitt der Cluster-Vorlage an, der verwendet werden soll.

`-p EXTRA_PARAMETERS, --extra-parameters EXTRA_PARAMETERS`

Fügt einer Stack-Aktualisierung zusätzliche Parameter hinzu.

`-rd, --reset-desired`

Setzt die aktuelle Kapazität einer Auto Scaling-Gruppe auf die anfängliche Konfigurationswerte zurück.

Standardeinstellung: `False`.

`--yes`

Geht automatisch davon aus, dass die Antwort auf alle Eingabeaufforderungen Ja lautet. Dies sollte nicht mit dem `--force` Argument kombiniert werden.

```
$ pcluster update -c path/to/config mycluster
Retrieving configuration from CloudFormation for cluster mycluster...
Validating configuration file .parallelcluster/config...
Found Configuration Changes:

#    parameter                   old value    new value
---  --------------------------  -----------  -----------
     [compute_resource default]
01   min_count                   1            2
02   max_count                   5            12

Validating configuration update...
Congratulations! The new configuration can be safely applied to your cluster.
Do you want to proceed with the update? - Y/N: Y
Updating: mycluster
Calling update_stack
Status: parallelcluster-mycluster - UPDATE_COMPLETE
```

Wenn der Befehl aufgerufen wird und mit der Abfrage des Status dieses Aufrufs beginnt, können Sie den Vorgang mit „Strg-C“ beenden. Sie können den aktuellen Status wieder anzeigen, indem Sie „`pcluster status mycluster`“ aufrufen.

pcluster version

Zeigt die Version an. AWS ParallelCluster

```
pcluster version [ -h ]
```

Für befehlsspezifische Flags führen Sie Folgendes aus: „`pcluster [command] --help`“

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster version`.

Wenn der Befehl aufgerufen wird und mit der Abfrage des Status dieses Aufrufs beginnt, können Sie den Vorgang mit „Strg-C“ beenden. Sie können den aktuellen Status wieder anzeigen, indem Sie „`pcluster status mycluster`“ aufrufen.

```
$ pcluster version
2.11.7
```

pcluster-config

Aktualisiert die Konfigurationsdatei AWS ParallelCluster .

```
pcluster-config [ -h ] [convert]
```

Für befehlsspezifische Flags führen Sie Folgendes aus: „`pcluster-config [command] -h`“

Benannte Argumente

-h, --help

Zeigt den Hilfetext für`pcluster-config`.

> **Note**
>
> Der `pcluster-config` Befehl wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt.

Unterbefehle:

`pcluster-config convert`

```
pcluster-config convert [ -h ] [ -c CONFIG_FILE ] [ -t CLUSTER_TEMPLATE ]
                        [ -o OUTPUT_FILE ]
```

Benannte Argumente

`-h, --help`

Zeigt den Hilfetext für`pcluster-config convert`.

`-c CONFIG_FILE, --config-file CONFIG_FILE`

Gibt den Pfad der zu lesenden Konfigurationsdatei an.

Standardeinstellung: `~/.parallelcluster/config`.

Weitere Informationen finden Sie unter Konfiguration AWS ParallelCluster.

`-t CLUSTER_TEMPLATE, --cluster-template CLUSTER_TEMPLATE`

Gibt an`[cluster]` Abschnitt, welche verwendet werden soll. Wenn dieses Argument nicht angegeben ist, `pcluster-config convert` wird die `cluster_template` Einstellung in der verwendet`[global]` Abschnitt. Wenn das nicht angegeben ist, wird der `[cluster default]` Abschnitt verwendet.

`-o OUTPUT_FILE, --output OUTPUT_FILE`

Gibt den Pfad der konvertierten Konfigurationsdatei an, die geschrieben werden soll. Standardmäßig wird in die Ausgabe geschrieben`STDOUT`.

Beispiel:

```
$ pcluster-config convert -t alpha -o ~/.parallelcluster/multiinstance
```

Konvertiert die im `[cluster alpha]` Abschnitt von angegebene Clusterkonfiguration `~/.parallelcluster/config` und schreibt die konvertierte Konfigurationsdatei in`~/.parallelcluster/multiinstance`.

Konfiguration

Standardmäßig AWS ParallelCluster verwendet die `~/.parallelcluster/config` Datei für alle Konfigurationsparameter. Sie können eine benutzerdefinierte Konfigurationsdatei angeben, indem Sie die `-c --config` Befehlszeilenoption oder oder die `AWS_PCLUSTER_CONFIG_FILE` Umgebungsvariable verwenden.

Eine Beispielkonfigurationsdatei ist mit AWS ParallelCluster im Python-Verzeichnis unter installiert`site-packages/aws-parallelcluster/examples/config`. Die Beispielkonfigurationsdatei ist auch auf GitHub, unter verfügbarhttps://github.com/aws/aws-parallelcluster/blob/v2.11.9/cli/src/pcluster/examples/config.

Aktuelle AWS ParallelCluster 2-Version: 2.11.9.

Themen

- Layout
- [global] Abschnitt
- [aws] Abschnitt
- [aliases] Abschnitt
- [cluster] Abschnitt
- [compute_resource] Abschnitt
- [cw_log] Abschnitt
- [dashboard] Abschnitt
- [dcv] Abschnitt
- [ebs] Abschnitt
- [efs] Abschnitt
- [fsx] Abschnitt
- [queue] Abschnitt
- [raid] Abschnitt
- [scaling] Abschnitt
- [vpc] Abschnitt
- Beispiele

Layout

Eine AWS ParallelCluster Konfiguration ist in mehreren Abschnitten definiert.

Die folgenden Abschnitte sind erforderlich: `[global]`Abschnitt und `[aws]`Abschnitt.

Sie müssen außerdem mindestens einen `[cluster]`Abschnitt und einen `[vpc]`Abschnitt angeben.

Ein Abschnitt beginnt mit dem Abschnittsnamen in Klammern, gefolgt von Parametern und der Konfiguration.

```
[global]
cluster_template = default
update_check = true
sanity_check = true
```

`[global]` Abschnitt

Gibt globale Konfigurationsoptionen im Zusammenhang mit `pcluster` an.

```
[global]
```

Themen

- cluster_template
- update_check
- sanity_check

`cluster_template`

Definiert den Namen des `cluster` Abschnitts, der standardmäßig für den Cluster verwendet wird. Weitere Informationen zu `cluster` Abschnitten finden Sie `[cluster]`im Abschnitt. Der Clustername muss mit einem Buchstaben beginnen, darf nicht mehr als 60 Zeichen enthalten und darf nur Buchstaben, Zahlen und Bindestriche (-) enthalten.

Beispiel: Die folgende Einstellung besagt, dass der Abschnitt, der `[cluster default]` startet, standardmäßig verwendet wird.

```
cluster_template = default
```

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

`update_check`

(Optional) Sucht nach Aktualisierungen für. `pcluster`

Der Standardwert ist `true`.

```
update_check = true
```

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

`sanity_check`

(Optional) Versucht, die Konfiguration der Ressourcen zu überprüfen, die in den Clusterparametern definiert sind.

Der Standardwert ist `true`.

> **Warning**
>
> Wenn auf gesetzt `sanity_check` ist`false`, werden wichtige Prüfungen übersprungen. Dies kann dazu führen, dass Ihre Konfiguration nicht wie vorgesehen funktioniert.

```
sanity_check = true
```

> **Note**
>
> Vor AWS ParallelCluster Version 2.5.0, `sanity_check` standardmäßig auf. `false`

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

`[aws]` Abschnitt

(Optional) Wird zur Auswahl von verwendet AWS-Region.

Bei der Clustererstellung wird diese Prioritätsreihenfolge verwendet, um Folgendes AWS-Region für einen neuen Cluster auszuwählen:

1. `-r`oder `--region` Parameter für`pcluster create`.
2. `AWS_DEFAULT_REGION`-Umgebungsvariable.
3. `aws_region_name`Einstellung im `[aws]` Abschnitt der AWS ParallelCluster Konfigurationsdatei (der Standardspeicherort ist`~/.parallelcluster/config`.) Dies ist der Speicherort, der durch den `pcluster configure` Befehl aktualisiert wurde.
4. `region`Einstellung im `[default]` Abschnitt der AWS CLI Konfigurationsdatei (`~/.aws/config`.)

> **Note**
>
> Vor AWS ParallelCluster Version 2.10.0 waren diese Einstellungen erforderlich und wurden auf alle Cluster angewendet.

Um Anmeldeinformationen zu speichern, können Sie die Umgebung, die IAM-Rollen für Amazon EC2 oder die verwenden AWS CLI, anstatt die Anmeldeinformationen in der AWS ParallelCluster Konfigurationsdatei zu speichern.

```
[aws]
aws_region_name = Region
```

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

`[aliases]` Abschnitt

Gibt Aliasnamen an und ermöglicht Ihnen, den `ssh`-Befehl anzupassen.

Beachten Sie die folgenden Standardeinstellungen:

- `CFN_USER`ist auf den Standardbenutzernamen für das Betriebssystem gesetzt
- `MASTER_IP`ist auf die IP-Adresse des Hauptknotens eingestellt
- `ARGS`ist auf die Argumente gesetzt, nach denen der Benutzer sie eingibt *`pcluster ssh cluster_name`*

```
[aliases]
# This is the aliases section, you can configure
```

```
# ssh alias here
ssh = ssh {CFN_USER}@{MASTER_IP} {ARGS}
```

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

`[cluster]` Abschnitt

Definiert eine Clustervorlage, die zum Erstellen eines Clusters verwendet werden kann. Eine Konfigurationsdatei kann mehrere `[cluster]` Abschnitte enthalten.

Dieselbe Cluster-Vorlage kann verwendet werden, um mehrere Cluster zu erstellen.

Das Format ist `[cluster` *`cluster-template-name`*`]`. Der durch die `cluster_template` Einstellung im `[cluster][global]`Abschnitt benannte Abschnitt wird standardmäßig verwendet, kann aber in der `pcluster` Befehlszeile überschrieben werden.

*`cluster-template-name`*muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[cluster default]
```

Themen

- additional_cfn_template
- additional_iam_policies
- base_os
- cluster_resource_bucket
- cluster_type
- compute_instance_type
- compute_root_volume_size
- custom_ami
- cw_log_settings
- dashboard_settings
- dcv_settings
- desired_vcpus
- disable_cluster_dns
- disable_hyperthreading

- ebs_settings
- ec2_iam_role
- efs_settings
- enable_efa
- enable_efa_gdr
- enable_intel_hpc_platform
- encrypted_ephemeral
- ephemeral_dir
- extra_json
- fsx_settings
- iam_lambda_role
- initial_queue_size
- key_name
- maintain_initial_size
- master_instance_type
- master_root_volume_size
- max_queue_size
- max_vcpus
- min_vcpus
- placement
- placement_group
- post_install
- post_install_args
- pre_install
- pre_install_args
- proxy_server
- queue_settings
- raid_settings
- s3_read_resource
- s3_read_write_resource

- scaling_settings
- scheduler
- shared_dir
- spot_bid_percentage
- spot_price
- tags
- template_url
- vpc_settings

additional_cfn_template

(Optional) Definiert eine zusätzliche AWS CloudFormation Vorlage, die zusammen mit dem Cluster gestartet wird. Diese zusätzliche Vorlage wird für die Erstellung von Ressourcen verwendet, die sich außerhalb des Clusters befinden, aber Teil des Cluster-Lebenszyklus sind.

Der Wert muss eine HTTP-URL zu einer öffentlichen Vorlage sein, in der alle Parameter angegeben sind.

Es ist kein Standardwert vorhanden.

```
additional_cfn_template = https://<bucket-name>.s3.amazonaws.com/my-cfn-template.yaml
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

additional_iam_policies

(Optional) Gibt eine Liste mit Amazon-Ressourcennamen (ARNs) von IAM-Richtlinien für Amazon EC2 an. Diese Liste ist zusätzlich zu den erforderlichen Berechtigungen durch Kommas AWS ParallelCluster getrennt an die im Cluster verwendete Root-Rolle angehängt. Ein IAM-Richtlinienname und sein ARN sind unterschiedlich. Namen können nicht als Argument für verwendet werden. `additional_iam_policies`

Wenn Sie beabsichtigen, den Standardeinstellungen für Clusterknoten zusätzliche Richtlinien hinzuzufügen, empfehlen wir Ihnen, die zusätzlichen benutzerdefinierten IAM-Richtlinien zusammen mit der `additional_iam_policies` Einstellung zu übergeben, anstatt die `ec2_iam_role`Einstellungen zum Hinzufügen Ihrer spezifischen EC2 Richtlinien zu verwenden. Dies liegt daran, `additional_iam_policies` dass zu den erforderlichen Standardberechtigungen

hinzugefügt werden AWS ParallelCluster . Ein vorhandenes `ec2_iam_role`muss alle erforderlichen Berechtigungen enthalten. Da sich die erforderlichen Berechtigungen jedoch häufig von Version zu Version ändern, wenn Funktionen hinzugefügt werden, `ec2_iam_role`kann eine bestehende Lizenz veraltet sein.

Es ist kein Standardwert vorhanden.

```
additional_iam_policies = arn:aws:iam::123456789012:policy/CustomEC2Policy
```

> **Note**
>
> Support für `additional_iam_policies`wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`base_os`

(Erforderlich) Gibt an, welcher Betriebssystemtyp im Cluster verwendet wird.

Verfügbare Optionen sind:

- `alinux2`
- `centos7`
- `ubuntu1804`
- `ubuntu2004`

> **Note**
>
> Für AWS Graviton-basierte Instances werden nur, `alinux2``ubuntu1804`, oder `ubuntu2004` unterstützt.

> **Note**
>
> Die Support für `centos8` wurde in AWS ParallelCluster Version 2.11.4 entfernt. Support für `ubuntu2004` wurde hinzugefügt und Unterstützung für `alinux` und `ubuntu1604` wurde in

AWS ParallelCluster Version 2.11.0 entfernt. Support für `centos8` wurde hinzugefügt und Unterstützung für `centos6` wurde in AWS ParallelCluster Version 2.10.0 entfernt. Support für `alinux2` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt. Support für `ubuntu1804` wurde hinzugefügt und Unterstützung für `ubuntu1404` wurde in AWS ParallelCluster Version 2.5.0 entfernt.

Mit Ausnahme der in der folgenden Tabelle AWS-Regionen aufgeführten Spezifika, die nicht unterstützt `centos7` werden. Alle anderen AWS kommerziellen Regionen unterstützen alle der folgenden Betriebssysteme.

Partition (AWS-Regionen)	**`alinux2`**	**`centos7`**	**`ubuntu1804`** und **`ubuntu2004`**
Kommerziell (Alle AWS-Regionen nicht ausdrücklich erwähnt)	True	True	True
AWS GovCloud (US-Ost) (`us-gov-east-1`)	True	False	True
AWS GovCloud (US-West) () `us-gov-west-1`	True	False	True
China (Peking) (`cn-north-1`)	True	False	True
China (Ningxia) (`cn-northwest-1`)	True	False	True

> **Note**
>
> Der `base_os` Parameter bestimmt auch den Benutzernamen, der für die Anmeldung beim Cluster verwendet wird.

- `centos7`: `centos`
- `ubuntu1804` und `ubuntu2004`: `ubuntu`

- `alinux2`: `ec2-user`

> **Note**
>
> Vor AWS ParallelCluster Version 2.7.0 war der `base_os` Parameter optional, und die Standardeinstellung war`alinux`. Ab AWS ParallelCluster Version 2.7.0 ist der `base_os` Parameter erforderlich.

> **Note**
>
> Wenn der Parameter `scheduler` `awsbatch` ist, wird nur `alinux2` unterstützt.

```
base_os = alinux2
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`cluster_resource_bucket`

(Optional) Gibt den Namen des Amazon S3 S3-Buckets an, der zum Hosten von Ressourcen verwendet wird, die bei der Erstellung des Clusters generiert werden. Für den Bucket muss die Versionierung aktiviert sein. Weitere Informationen finden Sie unter Verwenden der Versionierung im Amazon Simple Storage Service-Benutzerhandbuch. Dieser Bucket kann für mehrere Cluster verwendet werden. Der Bucket muss sich in derselben Region wie der Cluster befinden.

Wenn dieser Parameter nicht angegeben ist, wird bei der Erstellung des Clusters ein neuer Bucket erstellt. Der neue Bucket hat den Namen`parallelcluster-`*`random_string`*. In diesem Namen *`random_string`* befindet sich eine zufällige Zeichenfolge aus alphanumerischen Zeichen. Alle Clusterressourcen werden in diesem Bucket in einem Pfad mit der folgenden Form *`bucket_name`*`/`*`resource_directory`* gespeichert. `resource_directory`hat die Form*`stack_name`*`-`*`random_string`*, wobei der Name eines der AWS CloudFormation Stacks *`stack_name`* steht, die von AWS ParallelCluster verwendet werden. Der Wert von *`bucket_name`* kann dem `ResourcesS3Bucket` Wert in der Ausgabe des `parallelcluster-`*`clustername`* Stacks entnommen werden. Der Wert von *`resource_directory`* kann im Wert der `ArtifactS3RootDirectory` Ausgabe desselben Stacks gefunden werden.

Der Standardwert ist `parallelcluster-`*`random_string`*.

```
cluster_resource_bucket = amzn-s3-demo-bucket
```

> **Note**
>
> Support für `cluster_resource_bucket` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig. Die Aktualisierung dieser Einstellung kann nicht erzwungen werden.

`cluster_type`

(Optional) Definiert den Typ des Clusters, der gestartet werden soll. Wenn die `queue_settings` Einstellung definiert ist, muss diese Einstellung durch die `compute_type` Einstellungen in den `[queue]`Abschnitten ersetzt werden.

Gültige Optionen sind `ondemand` und `spot`.

Der Standardwert ist `ondemand`.

Weitere Informationen zu Spot-Instances finden Sie unter Arbeiten mit Spot-Instances.

> **Note**
>
> Die Verwendung von Spot-Instances setzt voraus, dass die `AWSServiceRoleForEC2Spot` mit dem Service verknüpfte Rolle in Ihrem Konto vorhanden ist. Führen Sie den folgenden Befehl aus AWS CLI, um diese Rolle in Ihrem Konto mithilfe von zu erstellen:
>
> ```
> aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
> ```
>
> Weitere Informationen finden Sie unter Service-verknüpfte Rolle für Spot-Instance-Anfragen im EC2 Amazon-Benutzerhandbuch.

```
cluster_type = ondemand
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

compute_instance_type

(Optional) Definiert den EC2 Amazon-Instance-Typ, der für die Cluster-Rechenknoten verwendet wird. Die Architektur des Instance-Typs muss mit der Architektur übereinstimmen, die für die `master_instance_type` Einstellung verwendet wurde. Wenn die `queue_settings` Einstellung definiert ist, muss diese Einstellung durch die `instance_type` Einstellungen in den `[compute_resource]`Abschnitten ersetzt werden.

Wenn Sie den `awsbatch` Scheduler verwenden, finden Sie unter der Erstellung von Compute Environments in der AWS Batch Benutzeroberfläche eine Liste der unterstützten Instanztypen.

Der Standardwert ist `t2.micro`, `optimal` wenn der Scheduler `awsbatch` ist.

```
compute_instance_type = t2.micro
```

> **Note**
>
> Support für AWS Graviton-basierte Instanzen (einschließlich `A1` `C6g` Und-Instanzen) wurde in AWS ParallelCluster Version 2.8.0 hinzugefügt.

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

compute_root_volume_size

(Optional) Gibt die Größe des ComputeFleet Root-Volumes in Gibibyte (GiB) an. Das AMI muss `growroot` unterstützen.

Der Standardwert ist `35`.

> **Note**
>
> Für AWS ParallelCluster Versionen zwischen 2.5.0 und 2.10.4 war der Standardwert 25. Vor AWS ParallelCluster Version 2.5.0 war die Standardeinstellung 20.

```
compute_root_volume_size = 35
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

custom_ami

(Optional) Gibt die ID eines benutzerdefinierten AMI an, das für die Head- und Compute-Knoten anstelle des veröffentlichten Standard-AMIs verwendet werden soll AMIs. Weitere Informationen finden Sie unter Ändern eines -AMIs oder Erstellen Sie ein benutzerdefiniertes AWS ParallelCluster AMI.

Es ist kein Standardwert vorhanden.

```
custom_ami = ami-00d4efc81188687a0
```

Wenn das benutzerdefinierte AMI zusätzliche Berechtigungen für seinen Start benötigt, müssen diese Berechtigungen sowohl den Benutzer- als auch den Headnode-Richtlinien hinzugefügt werden.

Wenn einem benutzerdefinierten AMI beispielsweise ein verschlüsselter Snapshot zugeordnet ist, sind die folgenden zusätzlichen Richtlinien sowohl in den Benutzer- als auch in den Headnode-Richtlinien erforderlich:

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

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

cw_log_settings

(Optional) Identifiziert den `[cw_log]` Abschnitt mit der CloudWatch Protokollkonfiguration. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im `[cw_log]`Abschnitt, und CloudWatch Amazon-Dashboard. Integration mit Amazon CloudWatch Logs

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der gestartet `[cw_log custom-cw]` wird, für die CloudWatch Logs-Konfiguration verwendet wird.

```
cw_log_settings = custom-cw
```

> **Note**
>
> Support für `cw_log_settings` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

dashboard_settings

(Optional) Identifiziert den `[dashboard]` Abschnitt mit der CloudWatch Dashboard-Konfiguration. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[dashboard]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der gestartet `[dashboard custom-dashboard` wird, für die CloudWatch Dashboard-Konfiguration verwendet wird.

```
dashboard_settings = custom-dashboard
```

> **Note**
>
> Support für `dashboard_settings` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

dcv_settings

(Optional) Identifiziert den `[dcv]` Abschnitt mit der Amazon DCV-Konfiguration. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[dcv]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der beginnt, für die Amazon DCV-Konfiguration verwendet `[dcv custom-dcv]` wird.

```
dcv_settings = custom-dcv
```

> **Note**
>
> Auf AWS Graviton-basierten Instances wird Amazon DCV nur auf unterstützt. `alinux2`

> **Note**
>
> Support für `dcv_settings` wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

desired_vcpus

(Optional) Gibt die gewünschte Anzahl von v CPUs in der Rechenumgebung an. Wird nur verwendet, wenn der Scheduler `awsbatch` ist.

Der Standardwert ist 4.

```
desired_vcpus = 4
```

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

disable_cluster_dns

(Optional) Gibt an, ob die DNS-Einträge für den Cluster nicht erstellt werden sollen. AWS ParallelCluster Erstellt standardmäßig eine Route 53-Hosting-Zone. Wenn auf gesetzt `disable_cluster_dns` ist`true`, wird die Hosting-Zone nicht erstellt.

Der Standardwert ist `false`.

```
disable_cluster_dns = true
```

> **Warning**
>
> Für den ordnungsgemäßen Betrieb des Clusters ist ein System zur Namensauflösung erforderlich. Wenn auf gesetzt `disable_cluster_dns` ist`true`, muss auch ein zusätzliches System zur Namensauflösung bereitgestellt werden.

> **Important**
>
> `disable_cluster_dns`= `true` wird nur unterstützt, wenn die `queue_settings` Einstellung angegeben ist.

> **Note**
>
> Support für `disable_cluster_dns` wurde in AWS ParallelCluster Version 2.9.1 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

disable_hyperthreading

(Optional) Deaktiviert Hyperthreading auf den Head- und Compute-Knoten. Nicht alle Instance-Typen können Hyperthreading deaktivieren. Eine Liste der Instance-Typen, die die Deaktivierung von Hyperthreading unterstützen, finden Sie im EC2 Amazon-Benutzerhandbuch unter CPU-Kerne und Threads für jeden CPU-Kern für jeden Instance-Typ. Wenn die `queue_settings` Einstellung

definiert ist, kann entweder diese Einstellung oder die `disable_hyperthreading` Einstellungen in den `[queue]`Abschnitten definiert werden.

Der Standardwert ist `false`.

```
disable_hyperthreading = true
```

> **Note**
>
> `disable_hyperthreading`wirkt sich nur auf den Kopfknoten aus, wenn`scheduler` = `awsbatch`.

> **Note**
>
> Support für `disable_hyperthreading` wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`ebs_settings`

(Optional) Identifiziert die `[ebs]` Abschnitte mit den Amazon EBS-Volumes, die auf dem Hauptknoten bereitgestellt sind. Wenn Sie mehrere Amazon EBS-Volumes verwenden, geben Sie diese Parameter in eine Liste ein, wobei jeder Parameter durch ein Komma getrennt ist. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Bis zu fünf (5) zusätzliche Amazon EBS-Volumes werden unterstützt.

Weitere Informationen finden Sie im `[ebs]`Abschnitt.

Die folgende Einstellung gibt beispielsweise an, dass die Abschnitte, die beginnen `[ebs custom1]` und für die Amazon EBS-Volumes verwendet `[ebs custom2]` werden, verwendet werden.

```
ebs_settings = custom1, custom2
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ec2_iam_role

(Optional) Definiert den Namen einer vorhandenen IAM-Rolle für Amazon EC2 , die allen Instances im Cluster zugeordnet ist. Ein IAM-Rollenname und sein Amazon-Ressourcenname (ARN) unterscheiden sich. ARNs kann nicht als Argument für verwendet werden. `ec2_iam_role`

Bei Angabe dieser Option wird die `additional_iam_policies`-Einstellung ignoriert. Wenn Sie beabsichtigen, den Standardeinstellungen für Clusterknoten zusätzliche Richtlinien hinzuzufügen, empfehlen wir Ihnen, die zusätzlichen benutzerdefinierten IAM-Richtlinien zusammen mit der `additional_iam_policies`Einstellung zu übergeben, anstatt die `ec2_iam_role` Einstellungen zu verwenden.

Wenn diese Option nicht angegeben ist, wird die AWS ParallelCluster Standard-IAM-Rolle für Amazon EC2 verwendet. Weitere Informationen finden Sie unter AWS Identity and Access Management Rollen in AWS ParallelCluster.

Es ist kein Standardwert vorhanden.

```
ec2_iam_role = ParallelClusterInstanceRole
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

efs_settings

(Optional) Gibt Einstellungen an, die sich auf das Amazon EFS-Dateisystem beziehen. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[efs]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der beginnt, für die Amazon EFS-Dateisystemkonfiguration verwendet `[efs customfs]` wird.

```
efs_settings = customfs
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

enable_efa

(Optional) Gibt an, dass der Elastic Fabric Adapter (EFA) für die Rechenknoten aktiviert ist, falls vorhanden. Eine Liste der EC2 Instances, die EFA unterstützen, finden Sie unter Unterstützte Instance-Typen im EC2 Amazon-Benutzerhandbuch für Linux-Instances. Weitere Informationen finden Sie unter Elastic Fabric Adapter. Wenn die `queue_settings` Einstellung definiert ist, kann entweder diese Einstellung oder die `enable_efa` Einstellungen im `[queue]`Abschnitt definiert werden. Latenzen zwischen Instances sollten mithilfe einer Cluster-Placement-Gruppe minimiert werden. Weitere Informationen erhalten Sie unter `placement` und `placement_group`.

```
enable_efa = compute
```

> **Note**
>
> Support für EFA auf ARM-basierten Graviton2-Instances wurde in Version 2.10.1 hinzugefügt. AWS ParallelCluster

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

enable_efa_gdr

(Optional) Ab AWS ParallelCluster Version 2.11.3 hat diese Einstellung keine Auswirkung. Die Unterstützung des Elastic Fabric Adapter (EFA) für GPUDirect RDMA (Remote Direct Memory Access) ist immer aktiviert, wenn sie sowohl vom Instance-Typ als auch vom Betriebssystem unterstützt wird.

> **Note**
>
> AWS ParallelCluster Version 2.10.0 bis 2.11.2: Wenn`compute`, gibt an, dass die Elastic Fabric Adapter (EFA) -Unterstützung für GPUDirect RDMA (Remote Direct Memory Access) für die Rechenknoten aktiviert ist. Um diese Einstellung auf festzulegen, `compute` muss die Einstellung auf gesetzt sein. `enable_efa` `compute` Die EFA-Unterstützung für GPUDirect RDMA wird von bestimmten Instance-Typen (`p4d.24xlarge`) auf bestimmten Betriebssystemen (`base_os``isalinux2`, `centos7``ubuntu1804`, oder`ubuntu2004`) unterstützt. Wenn die `queue_settings` Einstellung definiert ist, kann entweder diese Einstellung oder die `enable_efa_gdr` Einstellungen in den `[queue]`Abschnitten definiert werden. Latenzen zwischen Instances sollten mithilfe einer Cluster-Placement-

Gruppe minimiert werden. Weitere Informationen erhalten Sie unter `placement` und `placement_group`.

```
enable_efa_gdr = compute
```

> **Note**
>
> Support für `enable_efa_gdr` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

`enable_intel_hpc_platform`

(Optional) Falls vorhanden, bedeutet dies, dass die Endbenutzer-Lizenzvereinbarung für Intel Parallel Studio akzeptiert wurde. Dadurch wird Intel Parallel Studio auf dem Hauptknoten installiert und von den Rechenknoten gemeinsam genutzt. Dadurch verlängert sich die Zeit, die der Hauptknoten für das Bootstrap benötigt, um mehrere Minuten. Die `enable_intel_hpc_platform` Einstellung wird nur unterstützt auf CentOS (7 `base_os = centos7`).

Der Standardwert ist `false`.

```
enable_intel_hpc_platform = true
```

> **Note**
>
> Der `enable_intel_hpc_platform` Parameter ist nicht mit AWS Graviton-basierten Instances kompatibel.

> **Note**
>
> Support für `enable_intel_hpc_platform` wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

encrypted_ephemeral

(Optional) Verschlüsselt die temporären Instance-Speicher-Volumes mit nicht wiederherstellbaren In-Memory-Schlüsseln mithilfe von LUKS (Linux Unified Key Setup).

Weitere Informationen finden Sie unter https://gitlab.com/cryptsetup/cryptsetup/blob/master/README.md.

Der Standardwert ist `false`.

```
encrypted_ephemeral = true
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ephemeral_dir

(Optional) Definiert den Pfad, in dem Instance-Speicher-Volumes bereitgestellt werden, falls sie verwendet werden.

Der Standardwert ist `/scratch`.

```
ephemeral_dir = /scratch
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

extra_json

(Optional) Definiert das zusätzliche JSON, das mit dem zusammengeführt wird Chef `dna.json`. Weitere Informationen finden Sie unter Ein benutzerdefiniertes AWS ParallelCluster AMI erstellen.

Der Standardwert ist `{}`.

```
extra_json = {}
```

> **Note**
>
> Ab AWS ParallelCluster Version 2.6.1 werden die meisten Installationsrezepte standardmäßig beim Starten von Knoten übersprungen, um die Startzeiten zu verkürzen.

Um alle Installationsrezepte für eine bessere Abwärtskompatibilität auf Kosten der Startzeiten auszuführen, fügen Sie den `cluster` Schlüssel in der Einstellung `"skip_install_recipes" : "no"` hinzu. `extra_json` Zum Beispiel:

```
extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

fsx_settings

(Optional) Gibt den Abschnitt an, der die FSx for Lustre-Konfiguration definiert. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[fsx]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der beginnt, für die FSx for Lustre-Konfiguration verwendet `[fsx fs]` wird.

```
fsx_settings = fs
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

iam_lambda_role

(Optional) Definiert den Namen einer vorhandenen AWS Lambda Ausführungsrolle. Diese Rolle ist allen Lambda-Funktionen im Cluster zugewiesen. Weitere Informationen finden Sie unter AWS Lambda -Ausführungsrolle im AWS Lambda -Entwicklerhandbuch.

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

Ein IAM-Rollenname und sein Amazon-Ressourcenname (ARN) unterscheiden sich. ARNs kann nicht als Argument für verwendet werden. `iam_lambda_role` Wenn `ec2_iam_role`

sowohl als auch definiert `iam_lambda_role` sind und das `scheduler` ist `sgeslurm`, oder`torque`, dann werden keine Rollen erstellt. Ist dies der `scheduler` `awsbatch` Fall, werden währenddessen Rollen erstellt`pcluster start`. Richtlinien finden Sie beispielsweise unter `ParallelClusterLambdaPolicy`mit, oder SGESlurmTorque und`ParallelClusterLambdaPolicy mit awsbatch`.

Es ist kein Standardwert vorhanden.

```
iam_lambda_role = ParallelClusterLambdaRole
```

> **Note**
>
> Support für `iam_lambda_role` wurde in AWS ParallelCluster Version 2.10.1 hinzugefügt.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`initial_queue_size`

(Optional) Legt die anfängliche Anzahl von EC2 Amazon-Instances fest, die als Rechenknoten im Cluster gestartet werden sollen. Wenn die `queue_settings` Einstellung definiert ist, muss diese Einstellung entfernt und durch die `initial_count` Einstellungen in den `[compute_resource]`Abschnitten ersetzt werden.

> **Note**
>
> Ab Version 2.11.5 wird die AWS ParallelCluster Verwendung von nicht unterstützt SGE or Torque Scheduler.

Diese Einstellung gilt nur für herkömmliche Scheduler (SGE, Slurm, und Torque). Wenn die `maintain_initial_size` Einstellung ist`true`, muss die `initial_queue_size` Einstellung mindestens eins (1) sein.

Wenn der Scheduler `awsbatch` ist, verwenden Sie stattdessen `min_vcpus`.

Standardeinstellung: 2.

```
initial_queue_size = 2
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

key_name

(Optional) Benennt ein vorhandenes EC2 Amazon-Schlüsselpaar, mit dem der SSH-Zugriff auf die Instances aktiviert werden soll.

```
key_name = mykey
```

> **Note**
>
> Vor AWS ParallelCluster Version 2.11.0 `key_name` war dies eine erforderliche Einstellung.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

maintain_initial_size

> **Note**
>
> Beginnend mit Version 2.11.5 unterstützt AWS ParallelCluster es nicht die Verwendung von SGE or Torque Scheduler.

(Optional) Behält die ursprüngliche Größe der Auto Scaling Scaling-Gruppe für herkömmliche Scheduler bei (SGE, Slurm, und Torque).

Wenn der Scheduler `awsbatch` ist, verwenden Sie stattdessen `desired_vcpus`.

Bei dieser Einstellung handelt es sich um ein boolesches Flag. Wenn auf gesetzt`true`, hat die Auto Scaling Scaling-Gruppe niemals weniger Mitglieder als den Wert von`initial_queue_size`, und der Wert von `initial_queue_size` muss eins (1) oder größer sein. Der Cluster kann immer noch auf den Wert `max_queue_size` skaliert werden. Wenn `cluster_type = spot` dann kann es bei der Auto Scaling Scaling-Gruppe zu Unterbrechungen von Instanzen kommen und die Größe kann darunter fallen`initial_queue_size`.

Wenn diese Option auf gesetzt ist`false`, kann die Auto Scaling Scaling-Gruppe auf null (0) Mitglieder herunterskalieren, um zu verhindern, dass Ressourcen ungenutzt bleiben, wenn sie nicht benötigt werden.

Wenn die `queue_settings` Einstellung definiert ist, muss diese Einstellung entfernt und durch die `min_count` Einstellungen `initial_count` und in den `[compute_resource]`Abschnitten ersetzt werden.

Standardeinstellung: `false`.

```
maintain_initial_size = false
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`master_instance_type`

(Optional) Definiert den EC2 Amazon-Instance-Typ, der für den Head-Knoten verwendet wird. Die Architektur des Instance-Typs muss mit der Architektur übereinstimmen, die für die `compute_instance_type` Einstellung verwendet wurde.

Bei AWS-Regionen denen, die über ein kostenloses Kontingent verfügen, wird standardmäßig der Instance-Typ „Kostenloses Kontingent“ (`t2.micro`oder`t3.micro`) verwendet. Falls AWS-Regionen es kein kostenloses Kontingent gibt, ist standardmäßig der Wert. `t3.micro` Weitere Informationen zum AWS kostenlosen Kontingent finden Sie unter AWS Kostenloses Kontingent FAQs.

```
master_instance_type = t2.micro
```

> **Note**
>
> Vor AWS ParallelCluster Version 2.10.1 war die Standardeinstellung in all. `t2.micro` AWS-Regionen In AWS ParallelCluster Version 2.10.0 wurde das für den `p4d.24xlarge` Hauptknoten nicht unterstützt. Support für AWS Graviton-basierte Instanzen (wie `A1` und`C6g`) wurde in AWS ParallelCluster Version 2.8.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`master_root_volume_size`

(Optional) Gibt die Größe des Stammvolumens des Kopfknotens in Gibibyte (GiB) an. Das AMI muss `growroot` unterstützen.

Der Standardwert ist `35`.

> **Note**
>
> Für AWS ParallelCluster Versionen zwischen 2.5.0 und 2.10.4 war der Standardwert 25. Vor AWS ParallelCluster Version 2.5.0 war die Standardeinstellung 20.

```
master_root_volume_size = 35
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

max_queue_size

(Optional) Legt die maximale Anzahl von EC2 Amazon-Instances fest, die im Cluster gestartet werden können. Wenn die `queue_settings` Einstellung definiert ist, muss diese Einstellung entfernt und durch die `max_count` Einstellungen in den `[compute_resource]`Abschnitten ersetzt werden.

> **Note**
>
> Ab Version 2.11.5 wird die AWS ParallelCluster Verwendung von nicht unterstützt SGE or Torque Scheduler.

Diese Einstellung gilt nur für herkömmliche Scheduler (SGE, Slurm, und Torque).

Wenn der Scheduler `awsbatch` ist, verwenden Sie stattdessen `max_vcpus`.

Standardeinstellung: `10`.

```
max_queue_size = 10
```

Aktualisierungsrichtlinie: Diese Einstellung kann während eines Updates geändert werden, aber die Rechenflotte sollte gestoppt werden, wenn der Wert reduziert wird. Andernfalls können bestehende Knoten beendet werden.

max_vcpus

(Optional) Gibt die maximale Anzahl von v CPUs in der Rechenumgebung an. Wird nur verwendet, wenn der Scheduler `awsbatch` ist.

Der Standardwert ist `20`.

```
max_vcpus = 20
```

Aktualisierungsrichtlinie: Diese Einstellung kann während eines Updates nicht verringert werden.

`min_vcpus`

(Optional) Behält die ursprüngliche Größe der Auto Scaling Scaling-Gruppe für den `awsbatch` Scheduler bei.

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

Wenn der Scheduler SGE, Slurm, oder Torque, verwende `maintain_initial_size` stattdessen.

Die Datenverarbeitungsumgebung hat niemals weniger Mitglieder als der Wert von `min_vcpus`.

Standardeinstellung: `0`.

```
min_vcpus = 0
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`placement`

(Optional) Definiert die Cluster-Placement-Gruppenlogik, sodass entweder der gesamte Cluster oder nur die Compute-Instances die Cluster-Placement-Gruppe verwenden können.

Wenn die `queue_settings` Einstellung definiert ist, sollte diese Einstellung entfernt und durch `placement_group` Einstellungen für jeden der `[queue]`Abschnitte ersetzt werden. Wenn dieselbe Platzierungsgruppe für verschiedene Instance-Typen verwendet wird, ist es wahrscheinlicher, dass die Anforderung aufgrund eines Fehlers mit unzureichender Kapazität fehlschlägt. Weitere Informationen finden Sie unter Unzureichende Instance-Kapazität im EC2 Amazon-Benutzerhandbuch. Mehrere Warteschlangen können sich eine Platzierungsgruppe nur teilen, wenn sie im Voraus erstellt und in den `placement_group` Einstellungen für jede Warteschlange

konfiguriert wurde. Wenn jeder `[queue]`Abschnitt eine `placement_group` Einstellung definiert, kann sich der Hauptknoten nicht in der Platzierungsgruppe für eine Warteschlange befinden.

Gültige Optionen sind `cluster` oder `compute`.

Dieser Parameter wird nicht verwendet, wenn der Scheduler verwendet wird`awsbatch`.

Der Standardwert ist `compute`.

```
placement = compute
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`placement_group`

(Optional) Definiert die Cluster-Platzierungsgruppe. Wenn die `queue_settings` Einstellung definiert ist, sollte diese Einstellung entfernt und durch die `placement_group` Einstellungen in den `[queue]`Abschnitten ersetzt werden.

Gültige Optionen sind die folgenden Werte:

- `DYNAMIC`
- Ein vorhandener Name einer Amazon EC2 Cluster Placement-Gruppe

Wenn `DYNAMIC` festgelegt ist, wird eine eindeutige Platzierungsgruppe als Teil des Cluster-Stacks erstellt und gelöscht.

Dieser Parameter wird nicht verwendet, wenn der Scheduler aktiviert ist`awsbatch`.

Weitere Informationen zu Placement-Gruppen finden Sie unter Placement-Gruppen im EC2 Amazon-Benutzerhandbuch. Wenn dieselbe Platzierungsgruppe für verschiedene Instance-Typen verwendet wird, ist es wahrscheinlicher, dass die Anfrage aufgrund eines Fehlers mit unzureichender Kapazität fehlschlägt. Weitere Informationen finden Sie unter Unzureichende Instance-Kapazität im EC2 Amazon-Benutzerhandbuch.

Es ist kein Standardwert vorhanden.

Nicht alle Instance-Typen unterstützen Cluster-Placement-Gruppen. Beispielsweise unterstützt der Standard-Instance-Typ von `t3.micro` keine Cluster-Platzierungsgruppen. Informationen zur Liste

der Instance-Typen, die Cluster Placement-Gruppen unterstützen, finden Sie unter Regeln und Einschränkungen für Cluster Placement-Gruppen im EC2 Amazon-Benutzerhandbuch. Tipps zum Arbeiten mit Platzierungsgruppen finden Sie unter Probleme bei der Platzierung von Gruppen und beim Starten von Instances.

```
placement_group = DYNAMIC
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

post_install

(Optional) Gibt die URL eines Skripts nach der Installation an, das ausgeführt wird, nachdem alle Node-Bootstrap-Aktionen abgeschlossen sind. Weitere Informationen finden Sie unter Benutzerdefinierte Bootstrap-Aktionen.

Bei Verwendung `awsbatch` als Scheduler wird das Post-Installationsskript nur auf dem Hauptknoten ausgeführt.

Das Parameterformat kann entweder „`http://`*`hostname/path/to/script.sh`*“ oder „`s3://`*`bucket-name/path/to/script.sh`*“ sein.

Es ist kein Standardwert vorhanden.

```
post_install = s3://<bucket-name>/my-post-install-script.sh
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

post_install_args

(Optional) Gibt eine Liste von Argumenten in Anführungszeichen an, die an das Post-Installationsskript übergeben werden sollen.

Es ist kein Standardwert vorhanden.

```
post_install_args = "argument-1 argument-2"
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

pre_install

(Optional) Gibt die URL eines Vorinstallationsskripts an, das ausgeführt wird, bevor eine Bootstrap-Aktion zur Knotenbereitstellung gestartet wird. Weitere Informationen finden Sie unter Benutzerdefinierte Bootstrap-Aktionen.

Bei Verwendung `awsbatch` als Scheduler wird das Vorinstallationsskript nur auf dem Hauptknoten ausgeführt.

Das Parameterformat kann entweder „`http://hostname/path/to/script.sh`“ oder „`s3://bucket-name/path/to/script.sh`“ sein.

Es ist kein Standardwert vorhanden.

```
pre_install = s3://bucket-name/my-pre-install-script.sh
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

pre_install_args

(Optional) Gibt eine Liste von Argumenten in Anführungszeichen an, die an das Vorinstallationsskript übergeben werden sollen.

Es ist kein Standardwert vorhanden.

```
pre_install_args = "argument-3 argument-4"
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

proxy_server

(Optional) Definiert in der Regel `http://x.x.x.x:8080` einen HTTP- oder HTTPS-Proxyserver.

Es ist kein Standardwert vorhanden.

```
proxy_server = http://10.11.12.13:8080
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

queue_settings

(Optional) Gibt an, dass der Cluster Warteschlangen anstelle einer homogenen Rechenflotte verwendet, und gibt an, welche [queue]Abschnitte verwendet werden. Der erste aufgeführte [queue]Abschnitt ist die Standard-Scheduler-Warteschlange. Die queue Abschnittsnamen müssen mit einem Kleinbuchstaben beginnen, dürfen nicht mehr als 30 Zeichen enthalten und dürfen nur Kleinbuchstaben, Zahlen und Bindestriche (-) enthalten.

> **Important**
>
> queue_settingswird nur unterstützt, wenn scheduler auf gesetzt ist. slurm Die spot_price Einstellungen cluster_type compute_instance_typeinitial_queue_size,maintain_initial_size,max_queue_size,p und dürfen nicht angegeben werden. Die enable_efa Einstellungen disable_hyperthreading und können entweder im [cluster]Abschnitt oder in den [queue]Abschnitten angegeben werden, jedoch nicht in beiden.

Bis zu fünf (5) [queue]Abschnitte werden unterstützt.

Weitere Informationen finden Sie im [queue]Abschnitt.

Die folgende Einstellung gibt beispielsweise an, dass die Abschnitte beginnen [queue q1] und verwendet [queue q2] werden.

```
queue_settings = q1, q2
```

> **Note**
>
> Support für queue_settings wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt.

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

raid_settings

(Optional) Identifiziert den `[raid]` Abschnitt mit der Amazon EBS-Volume-RAID-Konfiguration. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[raid]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der beginnt, für die Auto Scaling Scaling-Konfiguration verwendet `[raid rs]` wird.

```
raid_settings = rs
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

s3_read_resource

(Optional) Gibt eine Amazon S3 S3-Ressource an, auf die AWS ParallelCluster Knoten nur Lesezugriff erhalten.

`arn:aws:s3:::`*`my_corporate_bucket`*`*`Bietet beispielsweise schreibgeschützten Zugriff auf den *`my_corporate_bucket`* Bucket und die Objekte im Bucket.

Einzelheiten zum Format finden Sie unter Arbeiten mit Amazon S3.

Es ist kein Standardwert vorhanden.

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

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

s3_read_write_resource

(Optional) Gibt eine Amazon S3 S3-Ressource an, für die AWS ParallelCluster Knoten Lese-/Schreibzugriff gewährt wird.

`arn:aws:s3:::`*`my_corporate_bucket`*`/Development/*`Bietet beispielsweise Lese-/Schreibzugriff auf alle Objekte im `Development` Ordner des Buckets. *`my_corporate_bucket`*

Einzelheiten zum Format finden Sie unter Arbeiten mit Amazon S3.

Es ist kein Standardwert vorhanden.

```
s3_read_write_resource = arn:aws:s3:::my_corporate_bucket/*
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

scaling_settings

Identifiziert den `[scaling]` Abschnitt mit der Auto Scaling Scaling-Konfiguration. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[scaling]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der beginnt, für die Auto Scaling Scaling-Konfiguration verwendet `[scaling custom]` wird.

```
scaling_settings = custom
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

scheduler

(Erforderlich) Definiert den Cluster-Scheduler.

Gültige Optionen sind die folgenden Werte:

`awsbatch`

AWS Batch

Weitere Informationen zum `awsbatch` Scheduler finden Sie unter Netzwerk-Setup undAWS Batch (`awsbatch`).

`sge`

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

Son of Grid Engine (SGE)

`slurm`

Slurm Workload Manager (Slurm)

`torque`

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

Torque Resource Manager (Torque)

> **Note**
>
> Vor AWS ParallelCluster Version 2.7.0 war der `scheduler` Parameter optional und der Standardwert war. `sge` Ab AWS ParallelCluster Version 2.7.0 ist der `scheduler` Parameter erforderlich.

```
scheduler = slurm
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

shared_dir

(Optional) Definiert den Pfad, in dem das gemeinsam genutzte Amazon EBS-Volume bereitgestellt wird.

Verwenden Sie diese Option nicht mit mehreren Amazon EBS-Volumes. Geben Sie stattdessen `shared_dir` Werte in jedem `[ebs]`Abschnitt an.

Einzelheiten zur Arbeit mit mehreren Amazon EBS-Volumes finden Sie im `[ebs]`Abschnitt.

Der Standardwert ist `/shared`.

Das folgende Beispiel zeigt ein gemeinsam genutztes Amazon EBS-Volume, das unter `/myshared` gemountet ist.

```
shared_dir = myshared
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

spot_bid_percentage

(Optional) Legt den On-Demand-Prozentsatz fest, der zur Berechnung des maximalen Spot-Preises für den ComputeFleet Terminplaner verwendet `awsbatch` wird.

Wenn Sie diesen Parameter nicht angeben, wird der aktuelle Spot-Marktpreis ausgewählt, dessen Obergrenze der On-Demand-Preis ist.

```
spot_bid_percentage = 85
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

spot_price

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

(Optional) Legt den maximalen Spot-Preis für herkömmliche Scheduler fest (ComputeFleet SGE, Slurm, und Torque). Wird nur verwendet, wenn die `cluster_type` Einstellung auf eingestellt ist`spot`. Wenn Sie keinen Wert angeben, wird Ihnen der Spot-Preis berechnet, der auf den On-Demand-Preis begrenzt ist. Wenn die `queue_settings` Einstellung definiert ist, muss diese Einstellung entfernt und durch die `spot_price` Einstellungen in den `[compute_resource]`Abschnitten ersetzt werden.

Wenn der Scheduler `awsbatch` ist, verwenden Sie stattdessen spot_bid_percentage.

Hilfe bei der Suche nach einer Spot-Instance, die Ihren Anforderungen entspricht, finden Sie im Spot-Instance-Berater.

```
spot_price = 1.50
```

> **Note**
>
> In AWS ParallelCluster Version 2.5.0 wird `cluster_type = spot` die `spot_price` Instance ohne Angabe des Fehlers gestartet ComputeFleet . Dies wurde in AWS ParallelCluster Version 2.5.1 behoben.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

tags

(Optional) Definiert Tags, von AWS CloudFormation denen verwendet werden soll.

Wenn Befehlszeilen-Tags über --tags angegeben sind, werden diese mit config-Tags zusammengeführt.

Befehlszeilen-Tags überschreiben config-Tags, die denselben Schlüssel haben.

Tags sind im JSON-Format. Verwenden Sie keine Anführungszeichen außerhalb der geschweiften Klammern.

Weitere Informationen finden Sie unter Typ von AWS CloudFormation Ressourcen-Tags im AWS CloudFormation Benutzerhandbuch.

```
tags = {"key" : "value", "key2" : "value2"}
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

> **Note**
>
> Die Aktualisierungsrichtlinie unterstützte nicht das Ändern der `tags` Einstellung für AWS ParallelCluster Version 2.8.0 bis Version 2.9.1.
> Für die Versionen 2.10.0 bis Version 2.11.7 ist die aufgeführte Aktualisierungsrichtlinie, die das Ändern der `tags` Einstellung unterstützte, nicht korrekt. Ein Cluster-Update beim Ändern dieser Einstellung wird nicht unterstützt.

template_url

(Optional) Definiert den Pfad zu der AWS CloudFormation Vorlage, die zur Erstellung des Clusters verwendet wird.

Updates verwenden die Vorlage, anhand der der Stack ursprünglich erstellt wurde.

Standardeinstellung: `https://aws_region_name-aws-parallelcluster.s3.amazonaws.com/templates/aws-parallelcluster-version.cfn.json`.

> **Warning**
>
> Dies ist ein erweiterter Parameter. Jede Änderung dieser Einstellung erfolgt auf eigenes Risiko.

```
template_url = https://us-east-1-aws-parallelcluster.s3.amazonaws.com/templates/aws-parallelcluster-2.11.9.cfn.json
```

Aktualisierungsrichtlinie: Diese Einstellung wird während eines Updates nicht analysiert.

vpc_settings

(Erforderlich) Identifiziert den `[vpc]` Abschnitt mit der Amazon VPC-Konfiguration, in dem der Cluster bereitgestellt wird. Der Abschnittsname muss mit einem Buchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Weitere Informationen finden Sie im Abschnitt. `[vpc]`

Die folgende Einstellung gibt beispielsweise an, dass der Abschnitt, der beginnt, für die Amazon VPC-Konfiguration verwendet `[vpc public]` wird.

```
vpc_settings = public
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`[compute_resource]` Abschnitt

Definiert Konfigurationseinstellungen für eine Rechenressource. `[compute_resource]`Abschnitte werden durch die `compute_resource_settings` Einstellung im `[queue]`Abschnitt referenziert. `[compute_resource]`Abschnitte werden nur unterstützt, wenn auf gesetzt `scheduler` ist`slurm`.

Das Format ist`[compute_resource <compute-resource-name>]`. *`compute-resource-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[compute_resource cr1]
instance_type = c5.xlarge
min_count = 0
initial_count = 2
max_count = 10
spot_price = 0.5
```

> **Note**
>
> Support für den `[compute_resource]`Abschnitt wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt.

Themen

- initial_count
- instance_type
- max_count
- min_count
- spot_price

`initial_count`

(Optional) Legt die anfängliche Anzahl von EC2 Amazon-Instances fest, die für diese Rechenressource gestartet werden sollen. Die Clustererstellung ist erst abgeschlossen, wenn mindestens so viele Knoten in die Rechenressource gestartet wurden. Wenn die `compute_type` Einstellung für die Warteschlange lautet `spot` und nicht genügend Spot-Instances verfügbar

sind, kann es bei der Clustererstellung zu einem Timeout kommen und ein Fehler auftreten. Jede Anzahl, die größer als die `min_count` Einstellung ist, ist abhängig von der `scaledown_idletime` Einstellung dynamische Kapazität. Diese Einstellung ersetzt die `initial_queue_size`-Einstellung.

Standardeinstellung: `0`.

```
initial_count = 2
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

`instance_type`

(Erforderlich) Definiert den EC2 Amazon-Instance-Typ, der für diese Rechenressource verwendet wird. Die Architektur des Instance-Typs muss mit der Architektur übereinstimmen, die für die `master_instance_type` Einstellung verwendet wurde. Die `instance_type` Einstellung muss für jeden `[compute_resource]`Abschnitt, auf den ein `[queue]`Abschnitt verweist, eindeutig sein. Diese Einstellung ersetzt die `compute_instance_type`-Einstellung.

```
instance_type = t2.micro
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

`max_count`

(Optional) Legt die maximale Anzahl von EC2 Amazon-Instances fest, die in dieser Rechenressource gestartet werden können. Jede Anzahl, die größer als die `initial_count` Einstellung ist, wird im Abschaltmodus gestartet. Diese Einstellung ersetzt die `max_queue_size`-Einstellung.

Standardeinstellung: `10`.

```
max_count = 10
```

Richtlinie aktualisieren: Um die Größe einer Warteschlange unter die aktuelle Anzahl von Knoten zu reduzieren, muss zuerst die Rechenflotte gestoppt werden.

> **Note**
>
> Die Aktualisierungsrichtlinie unterstützte das Ändern der `max_count` Einstellung erst, wenn die Rechenflotte für AWS ParallelCluster Version 2.0.0 bis Version 2.9.1 gestoppt wurde.

`min_count`

(Optional) Legt die Mindestanzahl von EC2 Amazon-Instances fest, die in dieser Rechenressource gestartet werden können. Bei diesen Knoten handelt es sich allesamt um statische Kapazitäten. Die Clustererstellung ist erst abgeschlossen, wenn mindestens diese Anzahl von Knoten in die Rechenressource gestartet wurde.

Standardeinstellung: `0`.

```
min_count = 1
```

Richtlinie aktualisieren: Um die Anzahl der statischen Knoten in einer Warteschlange zu reduzieren, muss zuerst die Rechenflotte gestoppt werden.

> **Note**
>
> Die Aktualisierungsrichtlinie unterstützte das Ändern der `min_count` Einstellung erst, wenn die Rechenflotte für die AWS ParallelCluster Versionen 2.0.0 bis 2.9.1 gestoppt wurde.

`spot_price`

(Optional) Legt den maximalen Spot-Preis für diese Rechenressource fest. Wird nur verwendet, wenn die `compute_type` Einstellung für die Warteschlange, die diese Rechenressourcen enthält, auf gesetzt ist`spot`. Diese Einstellung ersetzt die `spot_price`-Einstellung.

Wenn Sie keinen Wert angeben, wird Ihnen der Spot-Preis berechnet, der auf den On-Demand-Preis begrenzt ist.

Hilfe bei der Suche nach einer Spot-Instance, die Ihren Anforderungen entspricht, finden Sie im Spot-Instance-Berater.

```
spot_price = 1.50
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

`[cw_log]` Abschnitt

Definiert die Konfigurationseinstellungen für CloudWatch Logs.

Das Format ist`[cw_log cw-log-name]`. *`cw-log-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[cw_log custom-cw-log]
enable = true
retention_days = 14
```

Weitere Informationen finden Sie unter Integration mit Amazon CloudWatch Logs, CloudWatch Amazon-Dashboard und Integration mit Amazon CloudWatch Logs.

> **Note**
>
> Support für `cw_log` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

`enable`

(Optional) Gibt an, ob CloudWatch Logs aktiviert ist.

Der Standardwert ist `true`. Wird verwendet`false`, um CloudWatch Logs zu deaktivieren.

Das folgende Beispiel aktiviert CloudWatch Logs.

```
enable = true
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`retention_days`

(Optional) Gibt an, wie viele Tage CloudWatch Logs einzelne Protokollereignisse aufbewahrt.

Der Standardwert ist `14`. Die unterstützten Werte lauten 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827 und 3653.

Im folgenden Beispiel wird CloudWatch Logs so konfiguriert, dass Protokollereignisse 30 Tage lang aufbewahrt werden.

```
retention_days = 30
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`[dashboard]` Abschnitt

Definiert die Konfigurationseinstellungen für das CloudWatch Dashboard.

Das Format ist`[dashboard` *`dashboard-name`*`]`. *`dashboard-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[dashboard custom-dashboard]
enable = true
```

> **Note**
>
> Support für `dashboard` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

`enable`

(Optional) Gibt an, ob das CloudWatch Dashboard aktiviert ist.

Der Standardwert ist `true`. Wird verwendet`false`, um das CloudWatch Dashboard zu deaktivieren.

Das folgende Beispiel aktiviert das CloudWatch Dashboard.

```
enable = true
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

[dcv] Abschnitt

Definiert die Konfigurationseinstellungen für den Amazon DCV-Server, der auf dem Hauptknoten ausgeführt wird.

Um einen Amazon DCV-Server zu erstellen und zu konfigurieren, geben Sie den Cluster `dcv_settings` mit dem Namen an, den Sie im `dcv` Abschnitt definiert haben, und setzen Sie `enable` auf`master`, und `base_os` auf `alinux2centos7`, `ubuntu1804` oder`ubuntu2004`. Wenn der Hauptknoten eine ARM-Instance ist, legen Sie ihn `base_os` auf `alinux2centos7`, oder `ubuntu1804` fest.

Das Format ist`[dcv dcv-name]`. *`dcv-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[dcv custom-dcv]
enable = master
port = 8443
access_from = 0.0.0.0/0
```

Weitere Informationen finden Sie unter Stellen Sie über Amazon DCV eine Connect zum Hauptknoten her.

> **Important**
>
> Standardmäßig AWS ParallelCluster ist der Amazon DCV-Port, der eingerichtet wurde, für alle IPv4 Adressen geöffnet. Sie können jedoch nur dann eine Verbindung zu einem Amazon DCV-Port herstellen, wenn Sie die URL für die Amazon DCV-Sitzung haben und sich innerhalb von 30 Sekunden nach Rückgabe der URL mit der Amazon DCV-Sitzung verbinden. `pcluster dcv connect` Verwenden Sie die `access_from` Einstellung, um den Zugriff auf den Amazon DCV-Port mit einem CIDR-formatierten IP-Bereich weiter einzuschränken, und verwenden Sie die `port` Einstellung, um einen nicht standardmäßigen Port festzulegen.

> **Note**
>
> Die Support für den `[dcv]`Abschnitt on `centos8` wurde in AWS ParallelCluster Version 2.10.4 entfernt. Support für den `[dcv]`Abschnitt on `centos8` wurde in AWS ParallelCluster

Version 2.10.0 hinzugefügt. Support für den [dcv]Abschnitt über AWS Graviton-basierte Instanzen wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt. Support für den [dcv]Abschnitt über `alinux2` und `ubuntu1804` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt. Support für den [dcv]Abschnitt on `centos7` wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

access_from

(Optional, empfohlen) Gibt den CIDR-formatierten IP-Bereich für Verbindungen zu Amazon DCV an. Diese Einstellung wird nur bei der Erstellung AWS ParallelCluster der Sicherheitsgruppe verwendet.

Der Standardwert ist `0.0.0.0/0` und ermöglicht den Zugriff von jeder beliebigen Internetadresse.

```
access_from = 0.0.0.0/0
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

enable

(Erforderlich) Gibt an, ob Amazon DCV auf dem Hauptknoten aktiviert ist. Um Amazon DCV auf dem Hauptknoten zu aktivieren und die erforderliche Sicherheitsgruppenregel zu konfigurieren, setzen Sie die `enable` Einstellung auf`master`.

Das folgende Beispiel aktiviert Amazon DCV auf dem Hauptknoten.

```
enable = master
```

> **Note**
>
> Amazon DCV generiert automatisch ein selbstsigniertes Zertifikat, das zur Sicherung des Datenverkehrs zwischen dem Amazon DCV-Client und dem Amazon DCV-Server verwendet wird, der auf dem Hauptknoten ausgeführt wird. Informationen zum Konfigurieren Ihres eigenen Zertifikats finden Sie unter Amazon DCV HTTPS-Zertifikat.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

port

(Optional) Gibt den Port für Amazon DCV an.

Der Standardwert ist 8443.

```
port = 8443
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

[ebs] Abschnitt

Definiert Amazon EBS-Volume-Konfigurationseinstellungen für Volumes, die auf dem Hauptknoten bereitgestellt und über NFS für die Rechenknoten gemeinsam genutzt werden.

Informationen darüber, wie Sie Amazon EBS-Volumes in Ihre Cluster-Definition aufnehmen können, finden Sie unter[cluster] Abschnitt/ebs_settings.

Um ein vorhandenes Amazon EBS-Volume für dauerhaften Langzeitspeicher zu verwenden, der unabhängig vom Cluster-Lebenszyklus ist, geben Sie anebs_volume_id.

Wenn Sie keine Angabe machenebs_volume_id, AWS ParallelCluster erstellt das EBS-Volume anhand der [ebs] Einstellungen beim Erstellen des Clusters und löscht das Volume und die Daten, wenn der Cluster gelöscht wird.

Weitere Informationen finden Sie unter Bewährte Methoden: Umstellung eines Clusters auf eine neue AWS ParallelCluster Minor- oder Patch-Version.

Das Format ist. [ebs *ebs-name*] *ebs-name*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[ebs custom1]
shared_dir = vol1
ebs_snapshot_id = snap-xxxxx
volume_type = io1
volume_iops = 200
...
```

```
[ebs custom2]
shared_dir = vol2
...

...
```

Themen

- shared_dir
- ebs_kms_key_id
- ebs_snapshot_id
- ebs_volume_id
- encrypted
- volume_iops
- volume_size
- volume_throughput
- volume_type

shared_dir

(Erforderlich) Gibt den Pfad an, in dem das gemeinsam genutzte Amazon EBS-Volume bereitgestellt wird.

Dieser Parameter ist erforderlich, wenn Sie mehrere Amazon EBS-Volumes verwenden.

Wenn Sie ein Amazon EBS-Volume verwenden, überschreibt diese Option das Volume`shared_dir`, das `[cluster]` im Abschnitt angegeben ist. Im folgenden Beispiel wird das Volume auf `/vol1` gemountet.

```
shared_dir = vol1
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ebs_kms_key_id

(Optional) Gibt einen benutzerdefinierten AWS KMS Schlüssel an, der für die Verschlüsselung verwendet werden soll.

Dieser Parameter muss in Verbindung mit `encrypted = true` verwendet werden. Darüber hinaus muss er über eine benutzerdefinierte `ec2_iam_role` verfügen.

Weitere Informationen finden Sie unter Festplattenverschlüsselung mit einem benutzerdefinierten KMS-Schlüssel.

```
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ebs_snapshot_id

(Optional) Definiert die Amazon EBS-Snapshot-ID, wenn Sie einen Snapshot als Quelle für das Volume verwenden.

Es ist kein Standardwert vorhanden.

```
ebs_snapshot_id = snap-xxxxx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ebs_volume_id

(Optional) Definiert die Volume-ID eines vorhandenen Amazon EBS-Volumes, das an den Hauptknoten angehängt werden soll.

Es ist kein Standardwert vorhanden.

```
ebs_volume_id = vol-xxxxxx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

encrypted

(Optional) Gibt an, ob das Amazon EBS-Volume verschlüsselt ist. Hinweis: Nicht mit Snapshots verwenden.

Der Standardwert ist `false`.

```
encrypted = false
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

volume_iops

(Optional) Definiert die Anzahl der IOPS für Volumes `gp3` vom Typ `io1io2`, und.

Der Standardwert, die unterstützten Werte und `volume_iops` das `volume_size` Zu-Verhältnis variieren je nach `volume_type` und`volume_size`.

`volume_type` = `io1`

Standard `volume_iops` = 100

Unterstützte Werte `volume_iops` = 100—64000 †

Maximales `volume_iops` `volume_size` Verhältnis = 50 IOPS für jedes GiB. 5000 IOPS erfordern einen Wert `volume_size` von mindestens 100 GiB.

`volume_type` = `io2`

Standard `volume_iops` = 100

Unterstützte Werte `volume_iops` = 100—64000 (256000 für `io2` Block Express-Volumes) †

Maximales `volume_iops` `volume_size` Verhältnis = 500 IOPS für jedes GiB. 5000 IOPS erfordern einen Wert `volume_size` von mindestens 10 GiB.

`volume_type` = `gp3`

Standard `volume_iops` = 3000

Unterstützte Werte `volume_iops` = 3000—16000

Maximales `volume_iops` `volume_size` Verhältnis = 500 IOPS für jedes GiB. 5000 IOPS erfordern einen Wert `volume_size` von mindestens 10 GiB.

```
volume_iops = 200
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

† Maximale IOPS wird nur für Instances garantiert, die auf dem Nitro-System basieren und mit mehr als 32.000 IOPS ausgestattet sind. Andere Instanzen garantieren bis zu 32.000 IOPS. Wenn Sie das Volume nicht ändern, erreichen frühere `io1` Volumes möglicherweise nicht die volle Leistung. `io2` Block Express-Volumes unterstützen `volume_iops` Werte bis zu 256.000. Weitere Informationen finden Sie unter io2Block Express-Volumen (in der Vorschauversion) im EC2 Amazon-Benutzerhandbuch.

`volume_size`

(Optional) Gibt die Größe des zu erstellenden Volumes in GiB an (wenn Sie keinen Snapshot verwenden).

Der Standardwert und die unterstützten Werte variieren je nach`volume_type`.

`volume_type` = `standard`

Standard `volume_size` = 20 GiB

Unterstützte Werte `volume_size` = 1—1024 GiB

`volume_type=gp2`, `io1io2`, und `gp3`

Standard `volume_size` = 20 GiB

Unterstützte Werte `volume_size` = 1—16384 GiB

`volume_type=` und `sc1` `st1`

Standard `volume_size` = 500 GiB

Unterstützte Werte `volume_size` = 500—16384 GiB

```
volume_size = 20
```

> **Note**
>
> Vor AWS ParallelCluster Version 2.10.1 war der Standardwert für alle Volumetypen 20 GiB.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

volume_throughput

(Optional) Definiert den Durchsatz für gp3 Volumetypen in MiB/s.

Der Standardwert ist 125.

Unterstützte Werte volume_throughput = 125—1000 MiB/s

Das Verhältnis von volume_throughput zu volume_iops darf nicht mehr als 0,25 betragen. Der maximale Durchsatz von 1000 MiB/s setzt voraus, dass die volume_iops Einstellung mindestens 4000 beträgt.

```
volume_throughput = 1000
```

> **Note**
>
> Support für volume_throughput wurde in AWS ParallelCluster Version 2.10.1 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

volume_type

(Optional) Gibt den Amazon EBS-Volumetyp des Volumes an, das Sie starten möchten.

Gültige Optionen sind die folgenden Volumetypen:

gp2, gp3

Allzweck-SSD

io1, io2

Provisioned IOPS SSD

st1

Durchsatzoptimierte HDD

sc1

Cold HDD

`standard`

Magnetisch der vorherigen Generation

Weitere Informationen finden Sie unter Amazon EBS-Volumetypen im EC2 Amazon-Benutzerhandbuch.

Der Standardwert ist `gp2`.

```
volume_type = io2
```

> **Note**
>
> Support für `gp3` und `io2` wurde in AWS ParallelCluster Version 2.10.1 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`[efs]` Abschnitt

Definiert die Konfigurationseinstellungen für das Amazon EFS, das auf den Kopf- und Rechenknoten installiert ist. Weitere Informationen finden Sie CreateFileSystemin der Amazon EFS API-Referenz.

Informationen zum Einbeziehen von Amazon EFS-Dateisystemen in Ihre Cluster-Definition finden Sie unter`[cluster] Abschnitt`/`efs_settings`.

Um ein vorhandenes Amazon EFS-Dateisystem für dauerhaften Langzeitspeicher zu verwenden, der unabhängig vom Cluster-Lebenszyklus ist, geben Sie an`efs_fs_id`.

Wenn Sie nichts angeben`efs_fs_id`, AWS ParallelCluster erstellt das Amazon EFS-Dateisystem anhand der `[efs]` Einstellungen beim Erstellen des Clusters und löscht das Dateisystem und die Daten, wenn der Cluster gelöscht wird.

Weitere Informationen finden Sie unter Bewährte Methoden: Umstellung eines Clusters auf eine neue AWS ParallelCluster Minor- oder Patch-Version.

Das Format ist`[efs` *`efs-name`*`]`. *`efs-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[efs customfs]
shared_dir = efs
encrypted = false
performance_mode = generalPurpose
```

Themen

- efs_fs_id
- efs_kms_key_id
- encrypted
- performance_mode
- provisioned_throughput
- shared_dir
- throughput_mode

efs_fs_id

(Optional) Definiert die Amazon EFS-Dateisystem-ID für ein vorhandenes Dateisystem.

Wenn Sie diese Option angeben, werden alle anderen Amazon EFS-Optionen außer für `shared_dir` ungültig erklärt.

Wenn Sie diese Option festlegen, unterstützt sie nur die folgenden Arten von Dateisystemen:

- Dateisysteme, die kein Mount-Ziel in der Availability Zone des Stacks haben.
- Dateisysteme, die über ein vorhandenes Mount-Ziel in der Availability Zone des Stacks verfügen, von dem sowohl eingehender als auch ausgehender NFS-Verkehr zugelassen ist. `0.0.0.0/0`

Die Integritätsprüfung für die Validierung von `efs_fs_id` erfordert, dass die IAM-Rolle über die folgenden Berechtigungen verfügt:

- `elasticfilesystem:DescribeMountTargets`
- `elasticfilesystem:DescribeMountTargetSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkInterfaceAttribute`

Fügen Sie diese Berechtigungen Ihrer IAM-Rolle hinzu oder legen Sie `sanity_check = false` fest, um Fehler zu vermeiden.

> **Important**
>
> Wenn Sie ein Mount-Ziel festlegen, von dem ein- und ausgehender NFS-Verkehr zugelassen ist`0.0.0.0/0`, wird das Dateisystem NFS-Mount-Anfragen von überall in der Availability Zone des Mount-Ziels ausgesetzt. AWS empfiehlt nicht, ein Mount-Ziel in der Availability Zone des Stacks zu erstellen. Lassen Sie uns stattdessen diesen Schritt AWS erledigen. Wenn Sie ein Mount-Ziel in der Availability Zone des Stacks haben möchten, sollten Sie die Verwendung einer benutzerdefinierten Sicherheitsgruppe in Betracht ziehen, indem Sie im `[vpc]`Abschnitt eine `vpc_security_group_id` Option angeben. Fügen Sie dann diese Sicherheitsgruppe dem Mount-Ziel hinzu und schalten Sie `sanity_check` sie aus, um den Cluster zu erstellen.

Es ist kein Standardwert vorhanden.

```
efs_fs_id = fs-12345
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

efs_kms_key_id

(Optional) Identifiziert den AWS Key Management Service (AWS KMS) vom Kunden verwalteten Schlüssel, der zum Schutz des verschlüsselten Dateisystems verwendet werden soll. Wenn dies festgelegt ist , muss `encrypted`auf `true` eingestellt sein. Dies entspricht dem KmsKeyIdParameter in der Amazon EFS API-Referenz.

Es ist kein Standardwert vorhanden.

```
efs_kms_key_id = 1234abcd-12ab-34cd-56ef-1234567890ab
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

encrypted

(Optional) Gibt an, ob das Dateisystem verschlüsselt ist. Dies entspricht dem Parameter Encrypted in der Amazon EFS API-Referenz.

Der Standardwert ist `false`.

```
encrypted = true
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`performance_mode`

(Optional) Definiert den Leistungsmodus des Dateisystems. Dies entspricht dem PerformanceModeParameter in der Amazon EFS API-Referenz.

Gültige Optionen sind die folgenden Werte:

- `generalPurpose`
- `maxIO`

Bei beiden Werten wird zwischen Groß- und Kleinschreibung unterschieden.

Wir empfehlen den `generalPurpose`-Leistungsmodus für die meisten Dateisysteme.

Dateisysteme, die den `maxIO`-Leistungsmodus verwenden, können auf einen höheren Gesamtdurchsatz und mehr Vorgänge pro Sekunde skalieren. Bei den meisten Dateioperationen gibt es jedoch einen Kompromiss zwischen etwas höheren Latenzen.

Nachdem das Dateisystem erstellt wurde, kann dieser Parameter nicht mehr geändert werden.

Der Standardwert ist `generalPurpose`.

```
performance_mode = generalPurpose
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`provisioned_throughput`

(Optional) Definiert den bereitgestellten Durchsatz des Dateisystems, gemessen in MiB/s. Dies entspricht dem ProvisionedThroughputInMibpsParameter in der Amazon EFS API-Referenz.

Wenn Sie diesen Parameter verwenden, müssen Sie `throughput_mode` auf `provisioned` einstellen.

Die Quote für den Durchsatz beträgt `1024` MiB/s. Um eine Kontingenterhöhung anzufordern, wenn Sie sich an Support.

Der minimale Wert beträgt `0.0` MiB/s.

```
provisioned_throughput = 1024
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

shared_dir

(Erforderlich) Definiert den Amazon EFS-Bereitstellungspunkt auf den Kopf- und Rechenknoten.

Dieser Parameter muss angegeben werden. Der Amazon EFS-Abschnitt wird nur verwendet, wenn er angegeben `shared_dir` ist.

Verwenden Sie `NONE` oder nicht `/NONE` als gemeinsam genutztes Verzeichnis.

Im folgenden Beispiel wird Amazon EFS unter `/efs` bereitgestellt.

```
shared_dir = efs
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

throughput_mode

(Optional) Definiert den Durchsatzmodus des Dateisystems. Dies entspricht dem ThroughputModeParameter in der Amazon EFS API-Referenz.

Gültige Optionen sind die folgenden Werte:

- `bursting`
- `provisioned`

Der Standardwert ist `bursting`.

```
throughput_mode = provisioned
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`[fsx]` Abschnitt

Definiert Konfigurationseinstellungen für ein angehängtes FSx for Lustre-Dateisystem. Weitere Informationen finden Sie unter Amazon FSx CreateFileSystem in der Amazon FSx API-Referenz.

Wenn das `base_os` ist`alinux2`,`centos7`, oder `ubuntu1804`ubuntu2004, FSx für Lustre unterstützt wird.

Wenn Sie Amazon Linux verwenden, muss der Kernel `4.14.104-78.84.amzn1.x86_64` oder eine neuere Version sein. Anweisungen finden Sie unter Installation des Lustre-Clients im Amazon FSx for Lustre-Benutzerhandbuch.

> **Note**
>
> FSx for Lustre wird derzeit nicht unterstützt, wenn es `awsbatch` als Scheduler verwendet wird.

> **Note**
>
> Die Support FSx für Lustre on `centos8` wurde in AWS ParallelCluster Version 2.10.4 entfernt. Support FSx für Lustre on `ubuntu2004` wurde in AWS ParallelCluster Version 2.11.0 hinzugefügt. Support FSx für Lustre on `centos8` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt. Support FSx für Lustre on`alinux2`,`ubuntu1604`, und `ubuntu1804` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt. Support FSx für Lustre on `centos7` wurde in AWS ParallelCluster Version 2.4.0 hinzugefügt.

Bei Verwendung eines vorhandenen Dateisystems muss es einer Sicherheitsgruppe zugeordnet sein, die eingehenden TCP-Datenverkehr zu Port „988“ erlaubt. Wenn Sie die Quelle in einer Sicherheitsgruppenregel `0.0.0.0/0` auf festlegen, erhalten Sie Client-Zugriff von allen IP-Bereichen innerhalb Ihrer VPC-Sicherheitsgruppe für das Protokoll und den Portbereich für diese Regel. Um den Zugriff auf Ihre Dateisysteme weiter einzuschränken, empfehlen wir, restriktivere Quellen für Ihre Sicherheitsgruppenregeln zu verwenden. Sie können beispielsweise spezifischere CIDR-Bereiche, IP-Adressen oder Sicherheitsgruppen IDs verwenden. Dies geschieht automatisch, wenn `vpc_security_group_id` nicht verwendet wird.

Um ein vorhandenes FSx Amazon-Dateisystem für dauerhaften Langzeitspeicher zu verwenden, der unabhängig vom Cluster-Lebenszyklus ist, geben Sie an`fsx_fs_id`.

Wenn Sie nichts angeben`fsx_fs_id`, AWS ParallelCluster erstellt das FSx for Lustre-Dateisystem anhand der `[fsx]` Einstellungen beim Erstellen des Clusters und löscht das Dateisystem und die Daten, wenn der Cluster gelöscht wird.

Weitere Informationen finden Sie unter Bewährte Methoden: Umstellung eines Clusters auf eine neue AWS ParallelCluster Minor- oder Patch-Version.

Das Format ist. `[fsx fsx-name]` *`fsx-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[fsx fs]
shared_dir = /fsx
fsx_fs_id = fs-073c3803dca3e28a6
```

Verwenden Sie zum Erstellen und Konfigurieren eines neuen Dateisystems die folgenden Parameter:

```
[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
import_path = s3://bucket
weekly_maintenance_start_time = 1:00:00
```

Themen

- auto_import_policy
- automatic_backup_retention_days
- copy_tags_to_backups
- daily_automatic_backup_start_time
- data_compression_type
- deployment_type
- drive_cache_type
- export_path

- fsx_backup_id
- fsx_fs_id
- fsx_kms_key_id
- import_path
- imported_file_chunk_size
- per_unit_storage_throughput
- shared_dir
- storage_capacity
- storage_type
- weekly_maintenance_start_time

auto_import_policy

(Optional) Gibt die automatische Importrichtlinie an, um Änderungen im S3-Bucket widerzuspiegeln, der zur Erstellung des FSx for Lustre-Dateisystems verwendet wurde. Die folgenden Werte sind möglich:

`NEW`

FSx for Lustre importiert automatisch Verzeichnislisten aller neuen Objekte, die dem verknüpften S3-Bucket hinzugefügt wurden und die derzeit nicht im FSx for Lustre-Dateisystem vorhanden sind.

`NEW_CHANGED`

FSx for Lustre importiert automatisch Datei- und Verzeichnislisten aller neuen Objekte, die dem S3-Bucket hinzugefügt werden, sowie aller vorhandenen Objekte, die im S3-Bucket geändert wurden.

Dies entspricht der AutoImportPolicyEigenschaft. Weitere Informationen finden Sie unter Automatisches Importieren von Updates aus Ihrem S3-Bucket im Amazon FSx for Lustre-Benutzerhandbuch. Wenn der `auto_import_policy` Parameter angegeben ist, dürfen die `automatic_backup_retention_days` `fsx_backup_id` Parameter `copy_tags_to_backupsdaily_automatic_backup_start_time`,, und nicht angegeben werden.

Wenn die `auto_import_policy` Einstellung nicht angegeben ist, sind automatische Importe deaktiviert. FSx for Lustre aktualisiert nur Datei- und Verzeichnislisten aus dem verknüpften S3-Bucket, wenn das Dateisystem erstellt wird.

```
auto_import_policy = NEW_CHANGED
```

> **Note**
>
> Support für `auto_import_policy` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`automatic_backup_retention_days`

(Optional) Gibt an, wie viele Tage automatische Backups aufbewahrt werden sollen. Dies gilt nur für `PERSISTENT_1` Bereitstellungstypen. Wenn der `automatic_backup_retention_days` Parameter angegeben ist, dürfen die `auto_import_policy` `imported_file_chunk_size` Parameter `export_path``import_path`,, und nicht angegeben werden. Dies entspricht der AutomaticBackupRetentionDaysEigenschaft.

Der Standardwert lautet 0. Diese Einstellung deaktiviert automatische Backups. Die möglichen Werte sind ganze Zahlen zwischen 0 und einschließlich 35.

```
automatic_backup_retention_days = 35
```

> **Note**
>
> Support für `automatic_backup_retention_days` wurde in AWS ParallelCluster Version 2.8.0 hinzugefügt.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`copy_tags_to_backups`

(Optional) Gibt an, ob Tags für das Dateisystem in die Backups kopiert werden. Dies gilt nur für die Verwendung mit `PERSISTENT_1` Bereitstellungstypen. Wenn der `copy_tags_to_backups`

Parameter angegeben wird, `automatic_backup_retention_days` muss der mit einem Wert größer als 0 angegeben werden, und die `imported_file_chunk_size` Parameter `auto_import_policy` `export_pathimport_path`,, und dürfen nicht angegeben werden. Dies entspricht der CopyTagsToBackupsEigenschaft.

Der Standardwert ist `false`.

```
copy_tags_to_backups = true
```

> **Note**
>
> Support für `copy_tags_to_backups` wurde in AWS ParallelCluster Version 2.8.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

daily_automatic_backup_start_time

(Optional) Gibt die Tageszeit (UTC) an, zu der automatische Backups gestartet werden sollen. Dies gilt nur für die Verwendung mit `PERSISTENT_1` Bereitstellungstypen. Wenn der `daily_automatic_backup_start_time` Parameter angegeben wird, `automatic_backup_retention_days` muss der mit einem Wert größer als 0 angegeben werden, und die `imported_file_chunk_size` Parameter `auto_import_policy` `export_pathimport_path`,, und dürfen nicht angegeben werden. Dies entspricht der DailyAutomaticBackupStartTimeEigenschaft.

Das Format ist`HH:MM`, wobei `HH` die mit Nullen aufgefüllte Stunde des Tages (0-23) und die mit Nullen `MM` aufgefüllte Minute der Stunde ist. 1:03 Uhr UTC ist beispielsweise das Folgende.

```
daily_automatic_backup_start_time = 01:03
```

Der Standardwert ist eine zufällige Zeit zwischen `00:00` und`23:59`.

> **Note**
>
> Support für `daily_automatic_backup_start_time` wurde in AWS ParallelCluster Version 2.8.0 hinzugefügt.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

data_compression_type

(Optional) Gibt den FSx Datenkomprimierungstyp für Lustre an. Dies entspricht der DataCompressionTypeEigenschaft. Weitere Informationen finden Sie unter FSx Lustre-Datenkomprimierung im Amazon FSx for Lustre-Benutzerhandbuch.

Der einzige gültige Wert ist `LZ4`. Um die Datenkomprimierung zu deaktivieren, entfernen Sie den `data_compression_type`Parameter.

```
data_compression_type = LZ4
```

> **Note**
>
> Support für `data_compression_type` wurde in AWS ParallelCluster Version 2.11.0 hinzugefügt.

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

deployment_type

(Optional) Gibt den Bereitstellungstyp FSx für Lustre an. Dies entspricht der DeploymentTypeEigenschaft. Weitere Informationen finden Sie unter FSx Lustre-Bereitstellungsoptionen im Amazon FSx for Lustre-Benutzerhandbuch. Wählen Sie einen Scratch-Bereitstellungstyp für die temporäre Speicherung und die kurzfristigere Verarbeitung von Daten. `SCRATCH_2`ist die neueste Generation von Scratch-Dateisystemen. Es bietet einen höheren Burst-Durchsatz als den Basisdurchsatz und die Verschlüsselung von Daten bei der Übertragung.

Die gültigen Werte sind `SCRATCH_1`, `SCRATCH_2` und `PERSISTENT_1`.

`SCRATCH_1`

Der Standard-Bereitstellungstyp FSx für Lustre. Bei diesem Bereitstellungstyp hat die `storage_capacity`-Einstellung mögliche Werte von 1200, 2400 und von einem beliebigen Vielfachen von 3600. Support für `SCRATCH_1` wurde in AWS ParallelCluster Version 2.4.0 hinzugefügt.

`SCRATCH_2`

Die neueste Generation von Scratch-Dateisystemen. Sie unterstützt den bis zu sechsfachen Basisdurchsatz für hohe Workloads. Es unterstützt auch die Verschlüsselung von Daten während der Übertragung für unterstützte Instance-Typen, sofern unterstützt. AWS-Regionen Weitere Informationen finden Sie unter Verschlüsseln von Daten bei der Übertragung im Amazon FSx for Lustre-Benutzerhandbuch. Bei diesem Bereitstellungstyp hat die `storage_capacity`-Einstellung mögliche Werte von 1200 und von einem beliebigen Vielfachen von 2400. Support für `SCRATCH_2` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

`PERSISTENT_1`

Konzipiert für längerfristige Speicherung. Die Dateiserver sind hochverfügbar und die Daten werden innerhalb der Availability Zone der Dateisysteme repliziert. AWS Es unterstützt die Verschlüsselung von Daten während der Übertragung für unterstützte Instance-Typen. Bei diesem Bereitstellungstyp hat die `storage_capacity`-Einstellung mögliche Werte von 1200 und von einem beliebigen Vielfachen von 2400. Support für `PERSISTENT_1` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

Der Standardwert ist `SCRATCH_1`.

```
deployment_type = SCRATCH_2
```

> **Note**
>
> Support für `deployment_type` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`drive_cache_type`

(Optional) Gibt an, dass das Dateisystem über einen SSD-Laufwerkscache verfügt. Dies kann nur festgelegt werden, wenn die `storage_type` Einstellung auf gesetzt ist`HDD`. Dies entspricht der DriveCacheTypeEigenschaft. Weitere Informationen finden Sie unter FSx Lustre-Bereitstellungsoptionen im Amazon FSx for Lustre-Benutzerhandbuch.

Der einzige gültige Wert ist `READ`. Um den SSD-Laufwerk-Cache zu deaktivieren, geben Sie die `drive_cache_type` Einstellung nicht an.

```
drive_cache_type = READ
```

> **Note**
>
> Support für `drive_cache_type` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`export_path`

(Optional) Gibt den Amazon S3 S3-Pfad an, in den das Stammverzeichnis Ihres Dateisystems exportiert wird. Wenn der `export_path` Parameter angegeben ist, dürfen die `automatic_backup_retention_days` `fsx_backup_id` Parameter `copy_tags_to_backupsdaily_automatic_backup_start_time`,, und nicht angegeben werden. Dies entspricht der ExportPathEigenschaft. Dateidaten und Metadaten werden nicht automatisch in die exportiert`export_path`. Informationen zum Exportieren von Daten und Metadaten finden Sie unter Exportieren von Änderungen in das Daten-Repository im Amazon FSx for Lustre-Benutzerhandbuch.

Standardmäßig wird `s3://`*`import-bucket`*`/FSxLustre`*`[creation-timestamp]`* verwendet, wobei *`import-bucket`* der im Parameter `import_path` angegebene Bucket ist.

```
export_path = s3://bucket/folder
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`fsx_backup_id`

(Optional) Gibt die ID der Sicherung an, die für die Wiederherstellung des Dateisystems aus einer vorhandenen Sicherung verwendet werden soll. Wenn der `fsx_backup_id` Parameter angegeben ist, dürfen die `auto_import_policy` `per_unit_storage_throughput` Parameter `deployment_type` `export_pathfsx_kms_key_id`,`import_path`,`imported_file_chunk_size`,`storage_capacity`,, und nicht angegeben werden. Diese Parameter werden aus dem Backup gelesen. Darüber hinaus dürfen die `imported_file_chunk_size` Parameter `auto_import_policy` `export_pathimport_path`,, und nicht angegeben werden.

Dies entspricht der BackupIdEigenschaft.

```
fsx_backup_id = backup-fedcba98
```

> **Note**
>
> Support für `fsx_backup_id` wurde in AWS ParallelCluster Version 2.8.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

fsx_fs_id

(Optional) Hängt ein vorhandenes Dateisystem FSx für Lustre an.

Wenn diese Option angegeben ist, werden nur die `fsx_fs_id` Einstellungen `shared_dir` und im `[fsx]`Abschnitt verwendet und alle anderen Einstellungen im `[fsx]`Abschnitt werden ignoriert.

```
fsx_fs_id = fs-073c3803dca3e28a6
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

fsx_kms_key_id

(Optional) Gibt die Schlüssel-ID Ihres AWS Key Management Service (AWS KMS) vom Kunden verwalteten Schlüssels an.

Diese ID wird verwendet, um die Daten in Ihrem Dateisystem im Ruhezustand zu verschlüsseln.

Diese Einstellung muss mit einer benutzerdefinierten `ec2_iam_role` verwendet werden. Weitere Informationen finden Sie unter Festplattenverschlüsselung mit einem benutzerdefinierten KMS-Schlüssel. Dies entspricht dem KmsKeyIdParameter in der Amazon FSx API-Referenz.

```
fsx_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

> **Note**
>
> Support für `fsx_kms_key_id` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

import_path

(Optional) Gibt den S3-Bucket an, aus dem Daten in das Dateisystem geladen und als Export-Bucket verwendet werden sollen. Weitere Informationen finden Sie unter `export_path`. Wenn Sie den `import_path` Parameter angeben, dürfen die `automatic_backup_retention_days` `fsx_backup_id` Parameter `copy_tags_to_backupsdaily_automatic_backup_start_time`,, und nicht angegeben werden. Dies entspricht dem ImportPathParameter in der Amazon FSx API-Referenz.

Der Import findet bei der Cluster-Erstellung statt. Weitere Informationen finden Sie unter Daten aus Ihrem Daten-Repository importieren im Amazon FSx for Lustre-Benutzerhandbuch. Beim Import werden nur Dateimetadaten (Name, Besitz, Zeitstempel und Berechtigungen) importiert. Dateidaten werden erst aus dem S3-Bucket importiert, wenn auf die Datei zum ersten Mal zugegriffen wird. Informationen zum Vorladen des Dateiinhalts finden Sie unter Vorladen von Dateien in Ihr Dateisystem im Amazon FSx for Lustre-Benutzerhandbuch.

Wenn kein Wert angegeben wird, ist das Dateisystem leer.

```
import_path =  s3://bucket
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

imported_file_chunk_size

(Optional) Bestimmt die Stripe-Anzahl und die maximale Datenmenge für jede Datei (in MiB), die auf einer einzigen physischen Festplatte gespeichert ist, für Dateien, die aus einem Daten-Repository importiert werden (using`import_path`). Die maximale Anzahl von Datenträgern, über die eine einzelne Datei als Stripeset zugeordnet werden kann, ist durch die Gesamtzahl der Datenträger begrenzt, aus denen sich das Dateisystem zusammensetzt. Wenn der `imported_file_chunk_size` Parameter angegeben ist, dürfen die `automatic_backup_retention_days` `fsx_backup_id` Parameter `copy_tags_to_backupsdaily_automatic_backup_start_time`,, und nicht angegeben werden. Dies entspricht der ImportedFileChunkSizeEigenschaft.

Die Standardgröße für Chunks ist `1024` (1 GiB) und kann bis zu 512.000 MiB (500 GiB) betragen. Amazon S3-Objekte haben eine maximale Größe von 5 TB.

```
imported_file_chunk_size = 1024
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

per_unit_storage_throughput

(Erforderlich für **PERSISTENT_1**-Bereitstellungstypen) Beschreibt für den `deployment_type = PERSISTENT_1`-Bereitstellungstyp den Lese- und Schreibdurchsatz für jeweils 1 Tebibyte (TiB) Speicher in MB/s/TiB. Die Durchsatzkapazität des Dateisystems wird berechnet, indem die Dateisystemspeicherkapazität (TiB) mit (von) multipliziert wird, was einen `per_unit_storage_throughput` Dateisystemdurchsatz MB/s/TiB). For a 2.4 TiB file system, provisioning 50 MB/s/TiB von 120 MB/s `per_unit_storage_throughput` ergibt. Sie zahlen den Durchsatz, den Sie bereitstellen. Dies entspricht der Eigenschaft. PerUnitStorageThroughput

Die möglichen Werte hängen vom Wert der `storage_type` Einstellung ab.

`storage_type = SSD`

Die möglichen Werte lauten 50, 100, 200.

`storage_type = HDD`

Die möglichen Werte sind 12, 40.

```
per_unit_storage_throughput = 200
```

> **Note**
>
> Support für `per_unit_storage_throughput` wurde in AWS ParallelCluster Version 2.6.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

shared_dir

(Erforderlich) Definiert den Einhängepunkt für das FSx for Lustre-Dateisystem auf den Kopf- und Rechenknoten.

Verwenden Sie `NONE` oder nicht `/NONE` als gemeinsam genutztes Verzeichnis.

Im folgenden Beispiel wird das Dateisystem unter `/fsx` gemountet.

```
shared_dir = /fsx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

storage_capacity

(Erforderlich) Gibt die Speicherkapazität des Dateisystems in GiB an. Dies entspricht der StorageCapacityEigenschaft.

Die möglichen Werte für die Speicherkapazität variieren je nach `deployment_type`-Einstellung.

`SCRATCH_1`

Die möglichen Werte sind 1200, 2400 und ein beliebiges Vielfaches von 3600.

`SCRATCH_2`

Die möglichen Werte sind 1200 und ein beliebiges Vielfaches von 2400.

`PERSISTENT_1`

Die möglichen Werte variieren je nach den Werten anderer Einstellungen.

`storage_type = SSD`

Die möglichen Werte sind 1200 und ein beliebiges Vielfaches von 2400.

`storage_type = HDD`

Die möglichen Werte variieren je nach Einstellung der `per_unit_storage_throughput` Einstellung.

`per_unit_storage_throughput = 12`

Die möglichen Werte sind ein beliebiges Vielfaches von 6000.

`per_unit_storage_throughput = 40`

Die möglichen Werte sind ein beliebiges Vielfaches von 1800.

```
storage_capacity = 7200
```

> **Note**
>
> In den AWS ParallelCluster Versionen 2.5.0 und 2.5.1 wurden mögliche Werte von 1200, 2400 und jedes Vielfache von 3600 `storage_capacity` unterstützt. `storage_capacity`Hatte für Versionen vor AWS ParallelCluster Version 2.5.0 eine Mindestgröße von 3600.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`storage_type`

(Optional) Gibt den Speichertyp des Dateisystems an. Dies entspricht der StorageTypeEigenschaft. Die möglichen Werte sind `SSD` und `HDD`. Der Standardwert ist `SSD`.

Der Speichertyp ändert die möglichen Werte anderer Einstellungen.

`storage_type = SSD`

Gibt einen SSD-Speichertyp (Sold-State Drive) an.

`storage_type = SSD`ändert die möglichen Werte mehrerer anderer Einstellungen.

`drive_cache_type`

Diese Einstellung kann nicht angegeben werden.

`deployment_type`

Diese Einstellung kann auf `SCRATCH_1``SCRATCH_2`, oder gesetzt werden`PERSISTENT_1`.

`per_unit_storage_throughput`

Diese Einstellung muss angegeben werden, wenn `deployment_type` auf `PERSISTENT_1` gesetzt wird. Die möglichen Werte sind 50, 100 oder 200.

`storage_capacity`

Diese Einstellung muss angegeben werden. Die möglichen Werte variieren je nach`deployment_type`.

`deployment_type = SCRATCH_1`

`storage_capacity`kann 1200, 2400 oder ein beliebiges Vielfaches von 3600 sein.

`deployment_type = SCRATCH_2` oder `deployment_type = PERSISTENT_1`

`storage_capacity`kann 1200 oder ein beliebiges Vielfaches von 2400 sein.

`storage_type = HDD`

Gibt einen Speichertyp für ein Festplattenlaufwerk (HDD) an.

`storage_type = HDD`ändert die möglichen Werte anderer Einstellungen.

`drive_cache_type`

Diese Einstellung kann angegeben werden.

`deployment_type`

Diese Einstellung muss auf gesetzt sein`PERSISTENT_1`.

`per_unit_storage_throughput`

Diese Einstellung muss angegeben werden. Die möglichen Werte sind 12 oder 40.

`storage_capacity`

Diese Einstellung muss angegeben werden. Die möglichen Werte variieren je nach `per_unit_storage_throughput` Einstellung.

`storage_capacity = 12`

`storage_capacity`kann ein beliebiges Vielfaches von 6000 sein.

`storage_capacity = 40`

`storage_capacity`kann ein beliebiges Vielfaches von 1800 sein.

```
storage_type = SSD
```

> **Note**
>
> Support für die `storage_type`Einstellung wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`weekly_maintenance_start_time`

(Optional) Gibt den bevorzugten Zeitpunkt zum Durchführen der wöchentlichen Wartung in koordinierter Weltzeit (UTC) an. Dies entspricht der WeeklyMaintenanceStartTimeEigenschaft.

Das Format ist [Wochentag]: [Stunde des Tages]:[Minute der Stunde]. Montag um Mitternacht sieht zum Beispiel wie folgt aus.

```
weekly_maintenance_start_time = 1:00:00
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

`[queue]` Abschnitt

Definiert Konfigurationseinstellungen für eine einzelne Warteschlange. `[queue]`Abschnitte werden nur unterstützt, wenn auf gesetzt `scheduler` ist`slurm`.

Das Format ist`[queue` *`<queue-name>`*`]`. *`queue-name`*muss mit einem Kleinbuchstaben beginnen, darf nicht mehr als 30 Zeichen enthalten und darf nur Kleinbuchstaben, Zahlen und Bindestriche (-) enthalten.

```
[queue q1]
compute_resource_settings = i1,i2
placement_group = DYNAMIC
enable_efa = true
disable_hyperthreading = false
compute_type = spot
```

> **Note**
>
> Support für den `[queue]`Abschnitt wurde in AWS ParallelCluster Version 2.9.0 hinzugefügt.

Themen

- compute_resource_settings
- compute_type
- disable_hyperthreading

- enable_efa
- enable_efa_gdr
- placement_group

compute_resource_settings

(Erforderlich) Identifiziert die [compute_resource]Abschnitte, die die Rechenressourcenkonfigurationen für diese Warteschlange enthalten. Die Abschnittsnamen müssen mit einem Buchstaben beginnen, dürfen nicht mehr als 30 Zeichen enthalten und dürfen nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

Für jeden Abschnitt werden bis zu drei (3) [compute_resource]Abschnitte unterstützt. [queue]

Beispielsweise gibt die folgende Einstellung an, dass die Abschnitte beginnen [compute_resource cr1] und verwendet [compute_resource cr2] werden.

```
compute_resource_settings = cr1, cr2
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

compute_type

(Optional) Definiert den Typ der Instances, die für diese Warteschlange gestartet werden sollen. Diese Einstellung ersetzt die cluster_type-Einstellung.

Gültige Optionen sind ondemand und spot.

Der Standardwert ist ondemand.

Weitere Informationen zu Spot-Instances finden Sie unter Arbeiten mit Spot-Instances.

> **Note**
>
> Die Verwendung von Spot-Instances setzt voraus, dass die AWSServiceRoleForEC2Spot mit dem Service verknüpfte Rolle in Ihrem Konto vorhanden ist. Führen Sie den folgenden Befehl aus AWS CLI, um diese Rolle in Ihrem Konto mithilfe von zu erstellen:
>
> ```
> aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
> ```

Weitere Informationen finden Sie unter Service-verknüpfte Rolle für Spot-Instance-Anfragen im EC2 Amazon-Benutzerhandbuch.

Das folgende Beispiel verwendet SpotInstances für die Rechenknoten in dieser Warteschlange.

```
compute_type = spot
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

disable_hyperthreading

(Optional) Deaktiviert Hyperthreading auf den Knoten in dieser Warteschlange. Nicht alle Instance-Typen können Hyperthreading deaktivieren. Eine Liste der Instance-Typen, die die Deaktivierung von Hyperthreading unterstützen, finden Sie im EC2 Amazon-Benutzerhandbuch unter CPU-Kerne und Threads für jeden CPU-Kern pro Instance-Typ. Wenn die `disable_hyperthreading` Einstellung in dem `[cluster]`Abschnitt definiert ist, kann diese Einstellung nicht definiert werden.

Der Standardwert ist `false`.

```
disable_hyperthreading = true
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

enable_efa

(Optional) Wenn auf gesetzt`true`, gibt dies an, dass der Elastic Fabric Adapter (EFA) für die Knoten in dieser Warteschlange aktiviert ist. Eine Liste der EC2 Instances, die EFA unterstützen, finden Sie unter Unterstützte Instance-Typen im EC2 Amazon-Benutzerhandbuch für Linux-Instances. Wenn die `enable_efa` Einstellung in dem `[cluster]`Abschnitt definiert ist, kann diese Einstellung nicht definiert werden. Latenzen zwischen Instances sollten mithilfe einer Cluster-Placement-Gruppe minimiert werden. Weitere Informationen erhalten Sie unter `placement` und `placement_group`.

```
enable_efa = true
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

enable_efa_gdr

(Optional) Ab AWS ParallelCluster Version 2.11.3 hat diese Einstellung keine Auswirkung. Die Unterstützung des Elastic Fabric Adapter (EFA) für GPUDirect RDMA (Remote Direct Memory Access) ist für die Rechenknoten aktiviert. Sie ist immer aktiviert, wenn sie vom Instance-Typ unterstützt wird.

> **Note**
>
> AWS ParallelCluster Version 2.10.0 bis 2.11.2: Wenn`true`, gibt an, dass Elastic Fabric Adapter (EFA) GPUDirect RDMA (Remote Direct Memory Access) für die Knoten in dieser Warteschlange aktiviert ist. Die Einstellung auf `true` setzt voraus, dass die `enable_efa` Einstellung auf `true` .EFA gesetzt ist. GPUDirect RDMA wird von den folgenden Instance-Typen (`p4d.24xlarge`) auf diesen Betriebssystemen unterstützt (,, oder). `alinux2` `centos7` `ubuntu1804` `ubuntu2004` Wenn die `enable_efa_gdr` Einstellung in dem `[cluster]`Abschnitt definiert ist, kann diese Einstellung nicht definiert werden. Latenzen zwischen Instances sollten mithilfe einer Cluster-Placement-Gruppe minimiert werden. Weitere Informationen erhalten Sie unter `placement` und `placement_group`.

Der Standardwert ist `false`.

```
enable_efa_gdr = true
```

> **Note**
>
> Support für `enable_efa_gdr` wurde in AWS ParallelCluster Version 2.10.0 hinzugefügt.

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

placement_group

(Optional) Definiert, falls vorhanden, die Platzierungsgruppe für diese Warteschlange. Diese Einstellung ersetzt die `placement_group`-Einstellung.

Gültige Optionen sind die folgenden Werte:

- `DYNAMIC`
- Name einer vorhandenen Amazon EC2 Cluster Placement-Gruppe

Wenn diese Option auf gesetzt ist`DYNAMIC`, wird eine eindeutige Platzierungsgruppe für diese Warteschlange erstellt und als Teil des Cluster-Stacks gelöscht.

Weitere Informationen zu Placement-Gruppen finden Sie unter Placement-Gruppen im EC2 Amazon-Benutzerhandbuch. Wenn dieselbe Platzierungsgruppe für verschiedene Instance-Typen verwendet wird, ist es wahrscheinlicher, dass die Anfrage aufgrund eines Fehlers mit unzureichender Kapazität fehlschlägt. Weitere Informationen finden Sie unter Unzureichende Instance-Kapazität im EC2 Amazon-Benutzerhandbuch.

Es ist kein Standardwert vorhanden.

Nicht alle Instance-Typen unterstützen Cluster-Placement-Gruppen. Unterstützt beispielsweise `t2.micro` keine Cluster-Platzierungsgruppen. Informationen zur Liste der Instance-Typen, die Cluster Placement-Gruppen unterstützen, finden Sie unter Regeln und Einschränkungen für Cluster Placement-Gruppen im EC2 Amazon-Benutzerhandbuch. Tipps zum Arbeiten mit Platzierungsgruppen finden Sie unter Probleme bei der Platzierung von Gruppen und beim Starten von Instances.

```
placement_group = DYNAMIC
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

`[raid]` Abschnitt

Definiert Konfigurationseinstellungen für ein RAID-Array, das aus einer Reihe identischer Amazon EBS-Volumes aufgebaut ist. Das RAID-Laufwerk ist auf dem Hauptknoten montiert und wird mit NFS auf Rechenknoten exportiert.

Das Format ist`[raid raid-name]`. *`raid-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[raid rs]
```

```
shared_dir = raid
raid_type = 1
num_of_raid_volumes = 2
encrypted = true
```

Themen

- shared_dir
- ebs_kms_key_id
- encrypted
- num_of_raid_volumes
- raid_type
- volume_iops
- volume_size
- volume_throughput
- volume_type

shared_dir

(Erforderlich) Definiert den Einhängepunkt für das RAID-Array auf den Kopf- und Rechenknoten.

Das RAID-Laufwerk wird nur erstellt, wenn dieser Parameter angegeben ist.

Verwenden Sie `NONE` oder nicht `/NONE` als gemeinsam genutztes Verzeichnis.

Im folgenden Beispiel wird das Array unter `/raid` gemountet.

```
shared_dir = raid
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ebs_kms_key_id

(Optional) Gibt einen benutzerdefinierten AWS KMS Schlüssel an, der für die Verschlüsselung verwendet werden soll.

Dieser Parameter muss in Verbindung mit `encrypted = true` verwendet werden und eine benutzerdefinierte `ec2_iam_role` aufweisen.

Weitere Informationen finden Sie unter Festplattenverschlüsselung mit einem benutzerdefinierten KMS-Schlüssel.

```
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

encrypted

(Optional) Gibt an, ob das Dateisystem verschlüsselt ist.

Der Standardwert ist `false`.

```
encrypted = false
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

num_of_raid_volumes

(Optional) Definiert die Anzahl der Amazon EBS-Volumes, aus denen das RAID-Array zusammengestellt werden soll.

Die Mindestanzahl von Volumes ist2.

Die maximale Anzahl von Bänden ist5.

Der Standardwert ist 2.

```
num_of_raid_volumes = 2
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

raid_type

(Erforderlich) Definiert den RAID-Typ für das RAID-Array.

Das RAID-Laufwerk wird nur erstellt, wenn dieser Parameter angegeben ist.

Gültige Optionen sind die folgenden Werte:

- `0`
- `1`

Weitere Informationen zu RAID-Typen finden Sie unter RAID-Informationen im EC2 Amazon-Benutzerhandbuch.

Im folgenden Beispiel wird ein RAID-`0`-Array erstellt:

```
raid_type = 0
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`volume_iops`

(Optional) Definiert die Anzahl der IOPS für Volumes `gp3` vom Typ `io1io2`, und.

Der Standardwert, die unterstützten Werte und `volume_iops` das `volume_size` Zu-Verhältnis variieren je nach `volume_type` und`volume_size`.

`volume_type` = `io1`

Standard `volume_iops` = 100

Unterstützte Werte `volume_iops` = 100—64000 †

Maximales `volume_iops` `volume_size` Verhältnis = 50 IOPS pro GiB. 5000 IOPS erfordern einen Wert `volume_size` von mindestens 100 GiB.

`volume_type` = `io2`

Standard `volume_iops` = 100

Unterstützte Werte `volume_iops` = 100—64000 (256000 für `io2` Block Express-Volumes) †

Maximales `volume_iops` `volume_size` Verhältnis = 500 IOPS pro GiB. 5000 IOPS erfordern einen Wert `volume_size` von mindestens 10 GiB.

`volume_type` = `gp3`

Standard `volume_iops` = 3000

Unterstützte Werte `volume_iops` = 3000—16000

Maximales `volume_iops` `volume_size` Verhältnis = 500 IOPS pro GiB. 5000 IOPS erfordern einen Wert `volume_size` von mindestens 10 GiB.

```
volume_iops = 3000
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

† Maximale IOPS wird nur für Instances garantiert, die auf dem Nitro-System basieren und mit mehr als 32.000 IOPS ausgestattet sind. Andere Instanzen garantieren bis zu 32.000 IOPS. Ältere `io1` Volumes erreichen möglicherweise nicht die volle Leistung, es sei denn, Sie ändern das Volume. `io2` Block Express-Volumes unterstützen `volume_iops` Werte bis zu 256000. Weitere Informationen finden Sie unter `io2`Block Express-Volumen (in der Vorschauversion) im EC2 Amazon-Benutzerhandbuch.

`volume_size`

(Optional) Definiert die Größe des zu erstellenden Volumes in GiB.

Der Standardwert und die unterstützten Werte variieren je nach`volume_type`.

`volume_type` = `standard`

Standard `volume_size` = 20 GiB

Unterstützte Werte `volume_size` = 1—1024 GiB

`volume_type=gp2`, `io1io2`, und `gp3`

Standard `volume_size` = 20 GiB

Unterstützte Werte `volume_size` = 1—16384 GiB

`volume_type=` und `sc1` `st1`

Standard `volume_size` = 500 GiB

Unterstützte Werte `volume_size` = 500—16384 GiB

```
volume_size = 20
```

> **Note**
> Vor AWS ParallelCluster Version 2.10.1 war der Standardwert für alle Volumetypen 20 GiB.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`volume_throughput`

(Optional) Definiert den Durchsatz für `gp3` Volumetypen in MiB/s.

Der Standardwert ist `125`.

Unterstützte Werte `volume_throughput` = 125—1000 MiB/s

Das Verhältnis von `volume_throughput` zu `volume_iops` darf nicht mehr als 0,25 betragen. Der maximale Durchsatz von 1000 MiB/s setzt voraus, dass die `volume_iops` Einstellung mindestens 4000 beträgt.

```
volume_throughput = 1000
```

> **Note**
> Support für `volume_throughput` wurde in AWS ParallelCluster Version 2.10.1 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`volume_type`

(Optional) Definiert den Typ des zu erstellenden Volumes.

Gültige Optionen sind die folgenden Werte:

`gp2`, `gp3`

Allzweck-SSD

`io1`, `io2`

Provisioned IOPS SSD

`st1`

Durchsatzoptimierte HDD

`sc1`

Cold HDD

`standard`

Magnetisch der vorherigen Generation

Weitere Informationen finden Sie unter Amazon EBS-Volumetypen im EC2 Amazon-Benutzerhandbuch.

Der Standardwert ist `gp2`.

```
volume_type = io2
```

> **Note**
>
> Support für `gp3` und `io2` wurde in AWS ParallelCluster Version 2.10.1 hinzugefügt.

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

`[scaling]` Abschnitt

Themen

- scaledown_idletime

Gibt Einstellungen an, die definieren, wie die Datenverarbeitungsknoten skaliert werden.

Das Format ist`[scaling` *`scaling-name`*`]`. *`scaling-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[scaling custom]
scaledown_idletime = 10
```

scaledown_idletime

(Optional) Gibt die Zeit in Minuten ohne Job an, nach der der Rechenknoten beendet wird.

Dieser Parameter wird nicht verwendet, wenn es sich um den Scheduler `awsbatch` handelt.

Der Standardwert ist `10`.

```
scaledown_idletime = 10
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

[vpc] Abschnitt

Gibt die Amazon VPC-Konfigurationseinstellungen an. Weitere Informationen finden Sie VPCs unter Was ist Amazon VPC? und bewährte Sicherheitsmethoden für Ihre VPC im Amazon VPC-Benutzerhandbuch.

Das Format ist. `[vpc vpc-name]` *`vpc-name`*muss mit einem Buchstaben beginnen, nicht mehr als 30 Zeichen enthalten und darf nur Buchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

```
[vpc public]
vpc_id = vpc-xxxxxx
master_subnet_id = subnet-xxxxxx
```

Themen

- additional_sg
- compute_subnet_cidr
- compute_subnet_id
- master_subnet_id
- ssh_from
- use_public_ips
- vpc_id
- vpc_security_group_id

additional_sg

(Optional) Stellt eine zusätzliche Amazon VPC-Sicherheitsgruppen-ID für alle Instances bereit.

Es ist kein Standardwert vorhanden.

```
additional_sg = sg-xxxxxx
```

compute_subnet_cidr

(Optional) Spezifiziert einen CIDR-Block (Classless Inter-Domain Routing). Verwenden Sie diesen Parameter, wenn Sie ein AWS ParallelCluster Compute-Subnetz erstellen möchten.

```
compute_subnet_cidr = 10.0.100.0/24
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

compute_subnet_id

(Optional) Gibt die ID eines vorhandenen Subnetzes an, in dem die Rechenknoten bereitgestellt werden sollen.

Wenn nicht angegeben, verwendet `compute_subnet_id` der Wert von `master_subnet_id`.

Wenn das Subnetz privat ist, müssen Sie NAT für den Internetzugriff einrichten.

```
compute_subnet_id = subnet-xxxxxx
```

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

master_subnet_id

(Erforderlich) Gibt die ID eines vorhandenen Subnetzes an, in dem der Hauptknoten bereitgestellt werden soll.

```
master_subnet_id = subnet-xxxxxx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

ssh_from

(Optional) Gibt einen IP-Bereich im CIDR-Format an, von dem aus SSH-Zugriff möglich ist.

Dieser Parameter wird nur verwendet, wenn AWS ParallelCluster die Sicherheitsgruppe erstellt wird.

Der Standardwert ist `0.0.0.0/0`.

```
ssh_from = 0.0.0.0/0
```

Richtlinie aktualisieren: Diese Einstellung kann während eines Updates geändert werden.

use_public_ips

(Optional) Definiert, ob Computing-Instances öffentliche IP-Adressen zugewiesen werden sollen.

Wenn auf gesetzt`true`, wird dem Hauptknoten eine Elastic IP-Adresse zugeordnet.

Wenn auf gesetzt`false`, hat der Hauptknoten eine öffentliche IP (oder nicht), je nach dem Wert des Subnetz-Konfigurationsparameters „Automatische Zuweisung öffentlicher IP“.

Beispiele finden Sie unter Netzwerkkonfiguration.

Der Standardwert ist `true`.

```
use_public_ips = true
```

> **Important**
>
> Standardmäßig AWS-Konten sind alle auf jeweils fünf (5) Elastic IP-Adressen beschränkt. AWS-Region Weitere Informationen finden Sie unter Elastic IP address limit im EC2 Amazon-Benutzerhandbuch.

Aktualisierungsrichtlinie: Die Rechenflotte muss gestoppt werden, damit diese Einstellung für ein Update geändert werden kann.

vpc_id

(Erforderlich) Gibt die ID der Amazon VPC an, in der der Cluster bereitgestellt werden soll.

```
vpc_id = vpc-xxxxxx
```

Aktualisierungsrichtlinie: Wenn diese Einstellung geändert wird, ist das Update nicht zulässig.

vpc_security_group_id

(Optional) Gibt die Verwendung einer vorhandenen Sicherheitsgruppe für alle Instances an.

Es ist kein Standardwert vorhanden.

```
vpc_security_group_id = sg-xxxxxx
```

Die von erstellte Sicherheitsgruppe AWS ParallelCluster ermöglicht den SSH-Zugriff über Port 22 von den in der `ssh_from` Einstellung angegebenen Adressen oder von allen IPv4 Adressen (`0.0.0.0/0`) aus, wenn die `ssh_from` Einstellung nicht angegeben ist. Wenn Amazon DCV aktiviert ist, ermöglicht die Sicherheitsgruppe den Zugriff auf Amazon DCV über Port 8443 (oder was auch immer in der `port` Einstellung angegeben ist) von den in der `access_from` Einstellung angegebenen Adressen oder von allen IPv4 Adressen (`0.0.0.0/0`), falls die `access_from` Einstellung nicht angegeben ist.

> **Warning**
>
> Sie können den Wert dieses Parameters ändern und den Cluster aktualisieren, falls `[cluster]fsx_settings`nicht oder beides angegeben ist `fsx_settings` und ein externes, FSx für Lustre vorhandenes Dateisystem für in angegeben ist. `fsx-fs-id[fsx fs]`
> Sie können den Wert dieses Parameters nicht ändern, wenn in `fsx_settings` und ein FSx für Lustre AWS ParallelCluster verwaltetes Dateisystem angegeben ist. `[fsx fs]`

Aktualisierungsrichtlinie: Wenn AWS ParallelCluster verwaltete Amazon FSx for Lustre-Dateisysteme in der Konfiguration nicht angegeben sind, kann diese Einstellung während eines Updates geändert werden.

Beispiele

Die folgenden Beispielkonfigurationen demonstrieren AWS ParallelCluster Konfigurationen mit Slurm, Torque, und AWS Batch Scheduler.

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

Inhalt

- Slurm Workload Manager (slurm)
- Son of Grid Engine (sge) und Torque Resource Manager (torque)
- AWS Batch (awsbatch)

Slurm Workload Manager (`slurm`)

Das folgende Beispiel startet einen Cluster mit dem Scheduler `slurm`. In der Beispielkonfiguration wird 1 Cluster mit 2 Jobwarteschlangen gestartet. In der ersten Warteschlange `spot` sind zunächst 2 `t3.micro` Spot-Instances verfügbar. Sie kann auf maximal 10 Instances und auf mindestens 1 Instanz herunterskaliert werden, wenn 10 Minuten lang keine Jobs ausgeführt wurden (einstellbar über die `scaledown_idletime` Einstellung). Die zweite Warteschlange beginnt ohne Instanzen und kann auf maximal 5 `t3.micro` On-Demand-Instanzen skaliert werden. `ondemand`

```
[global]
update_check = true
sanity_check = true
cluster_template = slurm

[aws]
aws_region_name = <your AWS-Region>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>

[cluster slurm]
key_name = <your EC2 keypair name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = t3.micro     # optional, defaults to t3.micro
vpc_settings = public
queue_settings = spot,ondemand
```

```
[queue spot]
compute_resource_settings = spot_i1
compute_type = spot                 # optional, defaults to ondemand

[compute_resource spot_i1]
instance_type = t3.micro
min_count = 1                       # optional, defaults to 0
initial_count = 2                   # optional, defaults to 0

[queue ondemand]
compute_resource_settings = ondemand_i1

[compute_resource ondemand_i1]
instance_type = t3.micro
max_count = 5                       # optional, defaults to 10
```

Son of Grid Engine (**`sge`**) und Torque Resource Manager (**`torque`**)

> **Note**
>
> Dieses Beispiel gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Beginnend mit Version 2.11.5 unterstützt AWS ParallelCluster es nicht die Verwendung von SGE or Torque Scheduler.

Im folgenden Beispiel wird ein Cluster mit dem `sge` Scheduler `torque` OR gestartet. Zur Verwendung SGE, wechseln Sie `scheduler = torque` zu. `scheduler = sge` Die Beispielkonfiguration erlaubt maximal 5 gleichzeitige Knoten und wird auf zwei herunterskaliert, wenn 10 Minuten lang keine Jobs ausgeführt wurden.

```
[global]
update_check = true
sanity_check = true
cluster_template = torque

[aws]
aws_region_name = <your AWS-Region>

[vpc public]
master_subnet_id = <your subnet>
```

```
vpc_id = <your VPC>

[cluster torque]
key_name = <your EC2 keypair name>but they aren't eligible for future updates
base_os = alinux2                   # optional, defaults to alinux2
scheduler = torque                  # optional, defaults to sge
master_instance_type = t3.micro     # optional, defaults to t3.micro
vpc_settings = public
initial_queue_size = 2              # optional, defaults to 0
maintain_initial_size = true        # optional, defaults to false
max_queue_size = 5                  # optional, defaults to 10
```

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler. Wenn Sie diese Versionen verwenden, können Sie sie weiterhin verwenden oder den Support der AWS Service- und Support-Teams AWS bei der Fehlerbehebung unterstützen.

AWS Batch (`awsbatch`)

Das folgende Beispiel startet einen Cluster mit dem Scheduler `awsbatch`. Es ist so eingestellt, dass es auf der Grundlage Ihrer Anforderungen an die Jobressourcen den besseren Instanztyp auswählt.

Die Beispielkonfiguration erlaubt maximal 40 gleichzeitige V und wird auf Null herunterskaliertCPUs, wenn 10 Minuten lang keine Jobs ausgeführt wurden (einstellbar über die `scaledown_idletime` Einstellung).

```
[global]
update_check = true
sanity_check = true
cluster_template = awsbatch

[aws]
aws_region_name = <your AWS-Region>

[vpc public]
master_subnet_id = <your subnet>
vpc_id = <your VPC>
```

```
[cluster awsbatch]
scheduler = awsbatch
compute_instance_type = optimal # optional, defaults to optimal
min_vcpus = 0                   # optional, defaults to 0
desired_vcpus = 0               # optional, defaults to 4
max_vcpus = 40                  # optional, defaults to 20
base_os = alinux2               # optional, defaults to alinux2, controls the base_os
 of
                                # the head node and the docker image for the compute
 fleet
key_name = <your EC2 keypair name>
vpc_settings = public
```

Wie AWS ParallelCluster funktioniert

AWS ParallelCluster wurde nicht nur als Möglichkeit zur Verwaltung von Clustern konzipiert, sondern auch als Referenz für die Nutzung von AWS Diensten zum Aufbau Ihrer HPC-Umgebung.

Themen

- AWS ParallelCluster Prozesse
- AWS Dienste, die genutzt werden von AWS ParallelCluster
- AWS ParallelCluster Auto Scaling

AWS ParallelCluster Prozesse

Dieser Abschnitt gilt nur für HPC-Cluster, die mit einem der unterstützten herkömmlichen Job-Scheduler bereitgestellt werden (SGE, Slurm, oder Torque). Wenn es mit diesen Schedulern verwendet wird, AWS ParallelCluster verwaltet es die Bereitstellung und Entfernung von Rechenknoten, indem es sowohl mit der Auto Scaling Scaling-Gruppe als auch mit dem zugrunde liegenden Job-Scheduler interagiert.

Bei HPC-Clustern, die auf basieren, ist auf die Funktionen AWS ParallelCluster angewiesen AWS Batch, die von der AWS Batch für die Compute-Knotenverwaltung bereitgestellt werden.

> **Note**
>
> Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler. Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.

Themen

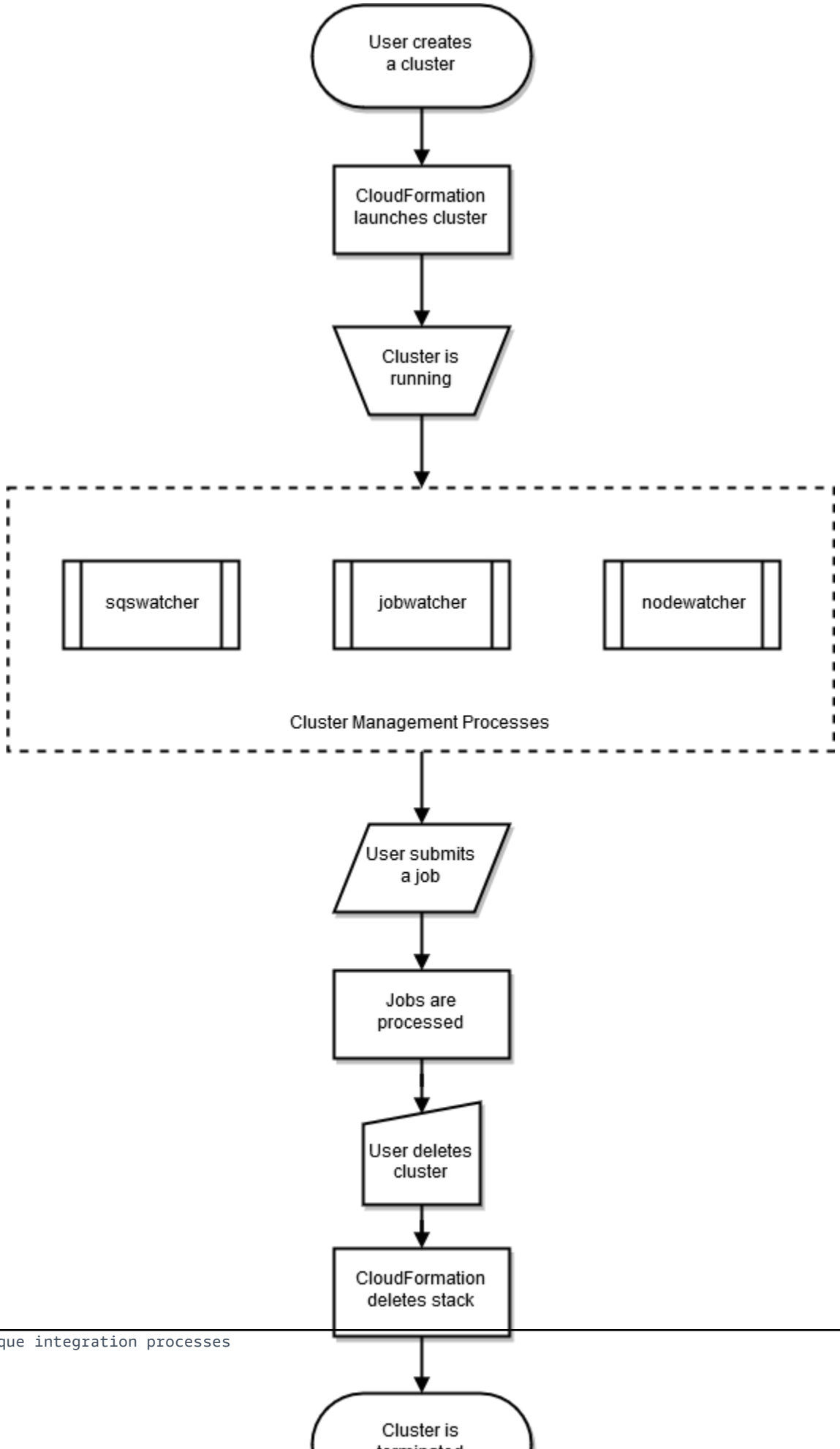
- SGE and Torque integration processes
- Slurm integration processes

SGE and Torque integration processes

> **Note**
>
> Dieser Abschnitt gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE and Torque Scheduler, Amazon SNS und Amazon SQS.

Allgemeiner Überblick

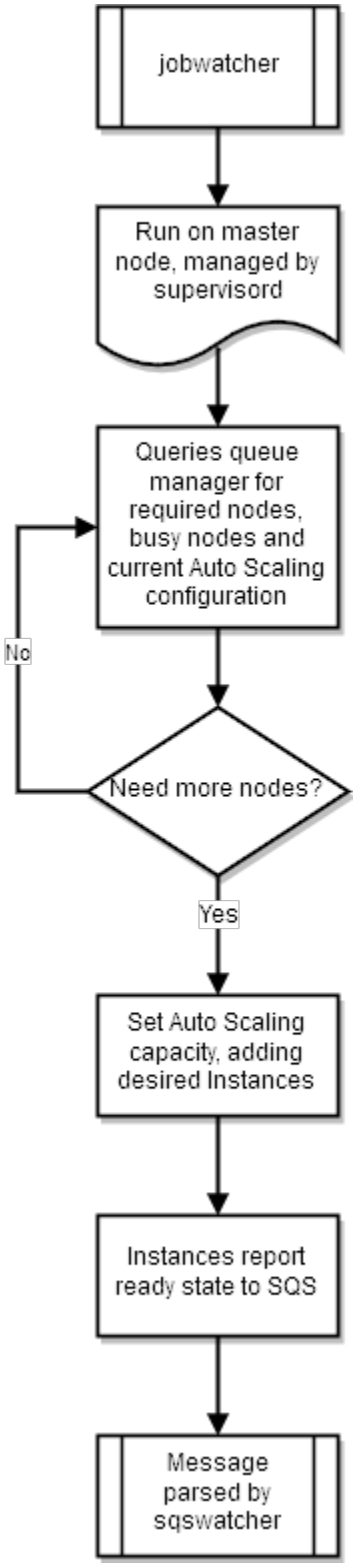
Der Lebenszyklus eines Clusters beginnt, nachdem er von einem Benutzer erstellt wurde. In der Regel wird ein Cluster über die Befehlszeilenschnittstelle (Command Line Interface, CLI) erstellt. Nach seiner Erstellung besteht ein Cluster solange, bis er gelöscht wird. AWS ParallelCluster Daemons werden auf den Clusterknoten ausgeführt, hauptsächlich um die Elastizität des HPC-Clusters zu verwalten. Das folgende Diagramm zeigt ein Benutzer-Workflow und den Cluster-Lebenszyklus. In den folgenden Abschnitten werden die AWS ParallelCluster Daemons beschrieben, die zur Verwaltung des Clusters verwendet werden.


User creates a cluster
CloudFormation launches cluster
Cluster is running
sqswatcher
jobwatcher
nodewatcher
Cluster Management Processes
User submits a job
Jobs are processed
User deletes cluster
CloudFormation deletes stack
Cluster is

Mit SGE and Torque Scheduler `nodewatcherjobwatcher`, AWS ParallelCluster Verwendungen und `sqswatcher` Prozesse.

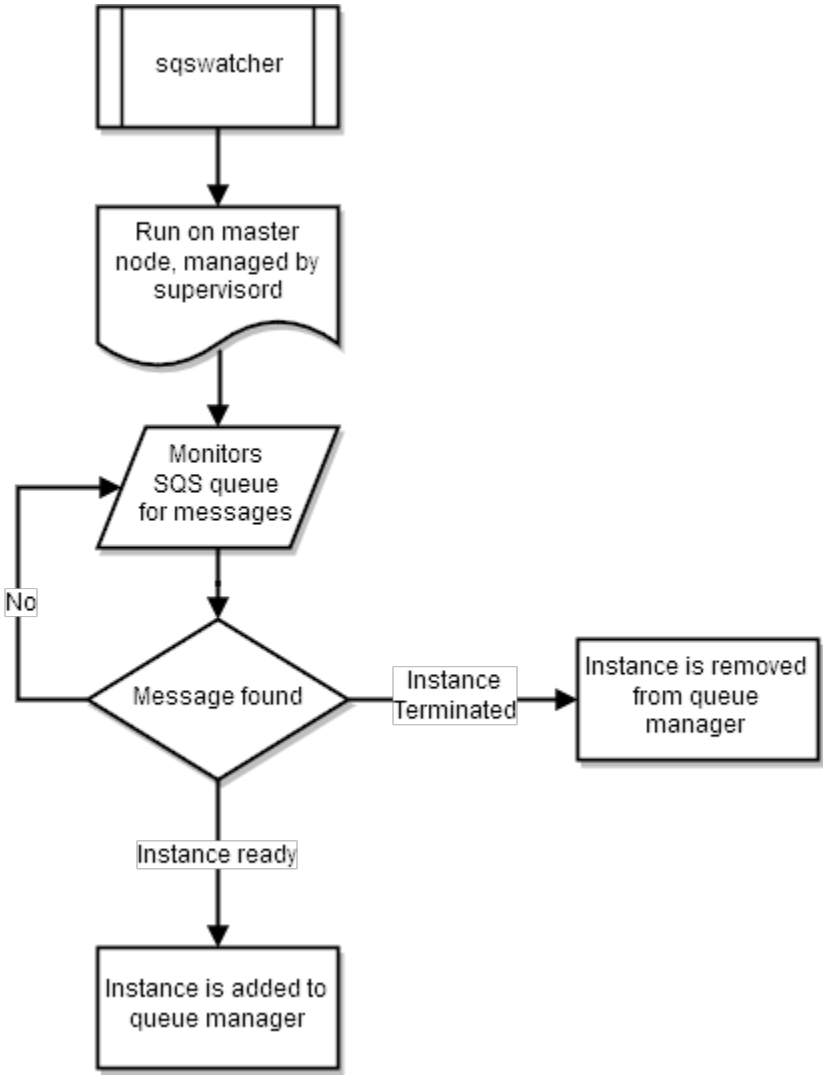
`jobwatcher`

Wenn ein Cluster läuft, überwacht ein Prozess, der dem Root-Benutzer gehört, den konfigurierten Scheduler (SGE or Torque). Jede Minute wird die Warteschlange ausgewertet, um zu entscheiden, wann sie hochskaliert werden soll.


jobwatcher
Run on master node, managed by supervisord
Queries queue manager for required nodes, busy nodes and current Auto Scaling configuration
No
Need more nodes?
Yes
Set Auto Scaling capacity, adding desired Instances
Instances report ready state to SQS
Message parsed by sqswatcher

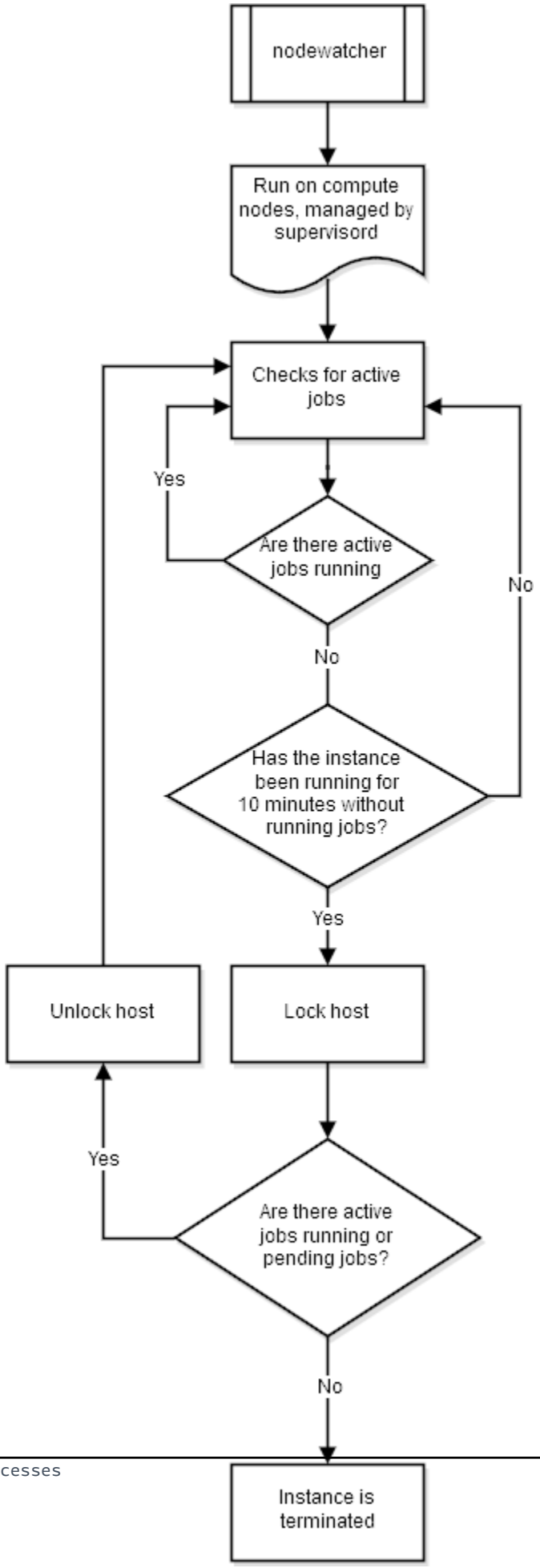
sqswatcher

Der `sqswatcher` Prozess sucht nach Amazon SQS SQS-Nachrichten, die von Auto Scaling gesendet werden, um Sie über Statusänderungen innerhalb des Clusters zu informieren. Wenn eine Instance online geht, sendet sie eine „Instance Ready“ -Meldung an Amazon SQS. Diese Nachricht wird von`sqs_watcher`, Running on the Head Node, abgeholt. Diese Nachrichten werden verwendet, um den Warteschlangenmanager zu benachrichtigen, wenn neue Instances online geschaltet oder beendet werden, sodass sie der Warteschlange hinzugefügt bzw. daraus entfernt werden können.



nodewatcher

Der `nodewatcher`-Prozess wird auf jedem Knoten in der Datenverarbeitungsflotte ausgeführt. Nach dem vom Benutzer definierten `scaledown_idletime`-Zeitraum wird die Instance beendet.


nodewatcher
Run on compute nodes, managed by supervisord
Checks for active jobs
Yes
Are there active jobs running
No
No
Has the instance been running for 10 minutes without running jobs?
Yes
Unlock host
Lock host
Yes
Are there active jobs running or pending jobs?
No
Instance is terminated

Slurm integration processes

Mit Slurm Scheduler, AWS ParallelCluster Verwendungen `clustermgtd` und `computemgt` Prozesse.

clustermgtd

Cluster, die im heterogenen Modus ausgeführt werden (dies wird durch Angabe eines `queue_settings` Werts angezeigt), verfügen über einen Daemon-Prozess (`clustermgtd`) für die Clusterverwaltung, der auf dem Hauptknoten ausgeführt wird. Diese Aufgaben werden vom Clusterverwaltungs-Daemon ausgeführt.

- Säuberung inaktiver Partitionen
- Statische Kapazitätsverwaltung: Stellen Sie sicher, dass die statische Kapazität immer verfügbar und funktionsfähig ist
- Synchronisieren Sie den Scheduler mit Amazon EC2.
- Bereinigung verwaister Instances
- Den Status des Scheduler-Knotens bei Amazon wiederherstellen — EC2 Kündigung, die außerhalb des Suspend-Workflows erfolgt
- Verwaltung fehlerhafter EC2 Amazon-Instanzen (fehlgeschlagene EC2 Amazon-Gesundheitschecks)
- Verwaltung von geplanten Wartungsereignissen
- Verwaltung fehlerhafter Scheduler-Knoten (fehlgeschlagene Zustandsprüfungen im Scheduler)

computemgtd

Cluster, die im heterogenen Modus ausgeführt werden (dies wird durch Angabe eines `queue_settings` Werts angezeigt), verfügen über Compute-Management-Daemon (`computemgtd`) -Prozesse, die auf jedem Rechenknoten ausgeführt werden. Alle fünf (5) Minuten bestätigt der Compute-Management-Daemon, dass der Hauptknoten erreichbar ist und fehlerfrei ist. Wenn fünf (5) Minuten vergehen, in denen der Hauptknoten nicht erreicht werden kann oder nicht fehlerfrei ist, wird der Rechenknoten heruntergefahren.

AWS Dienste, die genutzt werden von AWS ParallelCluster

Die folgenden Amazon Web Services (AWS) -Dienste werden von verwendet AWS ParallelCluster.

Themen

- AWS Auto Scaling
- AWS Batch
- AWS CloudFormation
- Amazon CloudWatch
- CloudWatch Amazon-Protokolle
- AWS CodeBuild
- Amazon-DynamoDB
- Amazon Elastic Block Store
- Amazon Elastic Compute Cloud
- Amazon Elastic Container Registry
- Amazon EFS
- Amazon FSx für Lustre
- AWS Identity and Access Management
- AWS Lambda
- Amazon DCV
- Amazon Route 53
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Simple Storage Service
- Amazon VPC

AWS Auto Scaling

> **Note**
>
> Dieser Abschnitt gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt. AWS Auto Scaling

AWS Auto Scaling ist ein Service, der Ihre Anwendungen überwacht und die Kapazität automatisch an Ihre spezifischen und sich ändernden Serviceanforderungen anpasst. Dieser Service verwaltet Ihre ComputeFleet Instances als Auto Scaling Scaling-Gruppe. Die Gruppe kann elastisch an Ihre sich ändernde Arbeitslast angepasst oder durch Ihre anfänglichen Instanzkonfigurationen statisch festgelegt werden.

AWS Auto Scaling wird mit ComputeFleet Instances verwendet, wird jedoch nicht mit AWS Batch Clustern verwendet.

Weitere Informationen zu AWS Auto Scaling finden Sie unter https://aws.amazon.com/autoscaling/und https://docs.aws.amazon.com/autoscaling/.

AWS Batch

AWS Batch ist ein AWS verwalteter Job Scheduler-Dienst. Er stellt dynamisch die optimale Menge und Art von Rechenressourcen (z. B. CPU- oder speicheroptimierte Instances) in Clustern bereit. AWS Batch Diese Ressourcen werden auf der Grundlage der spezifischen Anforderungen Ihrer Batch-Jobs bereitgestellt, einschließlich der Volumenanforderungen. Mit AWS Batch müssen Sie keine zusätzliche Batch-Computing-Software oder Servercluster installieren oder verwalten, um Ihre Jobs effektiv auszuführen.

AWS Batch wird nur mit AWS Batch Clustern verwendet.

Weitere Informationen zu AWS Batch finden Sie unter https://aws.amazon.com/batch/und https://docs.aws.amazon.com/batch/.

AWS CloudFormation

AWS CloudFormation ist ein infrastructure-as-code Dienst, der eine gemeinsame Sprache für die Modellierung AWS und Bereitstellung von Anwendungsressourcen von Drittanbietern in Ihrer Cloud-Umgebung bereitstellt. Es ist der Hauptdienst, der von verwendet wird AWS ParallelCluster. Jeder Cluster AWS ParallelCluster wird als Stack dargestellt, und alle Ressourcen, die von jedem Cluster benötigt werden, sind in der AWS ParallelCluster AWS CloudFormation Vorlage definiert. In den meisten Fällen entsprechen AWS ParallelCluster CLI-Befehle direkt AWS CloudFormation Stack-Befehlen wie Create-, Update- und Delete-Befehlen. Instances, die innerhalb eines Clusters gestartet werden, senden HTTPS-Aufrufe an den AWS CloudFormation Endpunkt, auf AWS-Region dem der Cluster gestartet wurde.

Weitere Informationen zu AWS CloudFormation finden Sie unter https://aws.amazon.com/cloudformation/und https://docs.aws.amazon.com/cloudformation/.

Amazon CloudWatch

Amazon CloudWatch (CloudWatch) ist ein Überwachungs- und Beobachtbarkeitsservice, der Ihnen Daten und umsetzbare Erkenntnisse liefert. Diese Erkenntnisse können verwendet werden, um Ihre Anwendungen zu überwachen, auf Leistungsänderungen und Serviceausnahmen zu reagieren und die Ressourcennutzung zu optimieren. In AWS ParallelCluster CloudWatch wird für ein Dashboard verwendet, um die Schritte zur Erstellung von Docker-Images und die Ausgabe der AWS Batch Jobs zu überwachen und zu protokollieren.

Vor AWS ParallelCluster Version 2.10.0 CloudWatch wurde es nur mit AWS Batch Clustern verwendet.

Weitere Informationen zu finden Sie CloudWatch unter https://aws.amazon.com/cloudwatch/und https://docs.aws.amazon.com/cloudwatch/.

CloudWatch Amazon-Protokolle

Amazon CloudWatch Logs (CloudWatch Logs) ist eine der Kernfunktionen von Amazon CloudWatch. Sie können damit die Protokolldateien für viele der von verwendeten Komponenten überwachen, speichern, anzeigen und durchsuchen AWS ParallelCluster.

Vor AWS ParallelCluster Version 2.6.0 wurde CloudWatch Logs nur mit AWS Batch Clustern verwendet.

Weitere Informationen finden Sie unter Integration mit Amazon CloudWatch Logs.

AWS CodeBuild

AWS CodeBuild (CodeBuild) ist ein AWS verwalteter Dienst für kontinuierliche Integration, der Quellcode einhält, Tests durchführt und Softwarepakete erstellt, die sofort bereitgestellt werden können. In CodeBuild wird verwendet AWS ParallelCluster, um automatisch und transparent Docker-Images zu erstellen, wenn Cluster erstellt werden.

CodeBuild wird nur mit AWS Batch Clustern verwendet.

Weitere Informationen zu CodeBuild finden Sie unter https://aws.amazon.com/codebuild/und https://docs.aws.amazon.com/codebuild/.

Amazon-DynamoDB

Amazon DynamoDB (DynamoDB) ist ein schneller und flexibler NoSQL-Datenbankservice. Er wird verwendet, um die minimalen Statusinformationen des Clusters zu speichern. Der Hauptknoten verfolgt bereitgestellte Instanzen in einer DynamoDB-Tabelle.

DynamoDB wird nicht mit AWS Batch Clustern verwendet.

Weitere Informationen zu DynamoDB finden Sie unter https://aws.amazon.com/dynamodb/und. https://docs.aws.amazon.com/dynamodb/

Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) ist ein leistungsstarker Blockspeicherservice, der persistenten Speicher für gemeinsam genutzte Volumes bereitstellt. Alle Amazon EBS-Einstellungen können durch die Konfiguration übergeben werden. Amazon EBS-Volumes können entweder leer oder aus einem vorhandenen Amazon EBS-Snapshot initialisiert werden.

Weitere Informationen zu Amazon EBS finden Sie unter https://aws.amazon.com/ebs/und https://docs.aws.amazon.com/ebs/.

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) bietet die Rechenkapazität für AWS ParallelCluster. Die Head- und Compute-Knoten sind EC2 Amazon-Instances. Es kann jeder Instance-Typ ausgewählt werden, der HVM unterstützt. Bei den Head- und Compute-Knoten kann es sich um unterschiedliche Instance-Typen handeln. Wenn mehrere Warteschlangen verwendet werden, können außerdem einige oder alle Rechenknoten auch als Spot-Instance gestartet werden. Auf den Instances vorhandene Instance-Speicher-Volumes werden als Striping-LVM-Volume gemountet.

Weitere Informationen zu Amazon EC2 finden Sie unter https://aws.amazon.com/ec2/und https://docs.aws.amazon.com/ec2/.

Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) ist eine vollständig verwaltete Docker-Container-Registry, die das Speichern, Verwalten und Bereitstellen von Docker-Container-Images vereinfacht. In AWS ParallelCluster speichert Amazon ECR die Docker-Images, die bei der Erstellung von Clustern erstellt werden. Die Docker-Images werden dann verwendet, AWS Batch um die Container für die eingereichten Jobs auszuführen.

Amazon ECR wird nur mit AWS Batch Clustern verwendet.

Weitere Informationen erhalten Sie unter https://aws.amazon.com/ecr/ und https://docs.aws.amazon.com/ecr/.

Amazon EFS

Amazon Elastic File System (Amazon EFS) bietet ein einfaches, skalierbares und vollständig verwaltetes elastisches NFS-Dateisystem zur Verwendung mit AWS Cloud Services und lokalen Ressourcen. Amazon EFS wird verwendet, wenn die `efs_settings` Einstellung angegeben ist und sich auf einen `[efs]`Abschnitt bezieht. Support für Amazon EFS wurde in AWS ParallelCluster Version 2.1.0 hinzugefügt.

Weitere Informationen zu Amazon EFS finden Sie unter https://aws.amazon.com/efs/und https://docs.aws.amazon.com/efs/.

Amazon FSx für Lustre

FSx for Lustre bietet ein Hochleistungsdateisystem, das das Open-Source-Dateisystem Lustre verwendet. FSx for Lustre wird verwendet, wenn die `fsx_settings` Einstellung angegeben ist und sich auf einen Abschnitt bezieht. `[fsx]` Support FSx für Lustre wurde in AWS ParallelCluster Version 2.2.1 hinzugefügt.

Weitere Informationen zu FSx for Lustre finden Sie unter https://aws.amazon.com/fsx/ lustre/ und. https://docs.aws.amazon.com/fsx/

AWS Identity and Access Management

AWS Identity and Access Management (IAM) wird innerhalb verwendet AWS ParallelCluster , um Amazon eine IAM-Rolle mit den geringsten Rechten für die Instance bereitzustellen, die EC2 für jeden einzelnen Cluster spezifisch ist. AWS ParallelCluster Instances erhalten nur Zugriff auf die spezifischen API-Aufrufe, die für die Bereitstellung und Verwaltung des Clusters erforderlich sind.

Bei AWS Batch Clustern werden bei der Clustererstellung auch IAM-Rollen für die Komponenten erstellt, die am Prozess der Docker-Image-Erstellung beteiligt sind. Zu diesen Komponenten gehören die Lambda-Funktionen, mit denen Docker-Images zum Amazon ECR-Repository hinzugefügt und daraus gelöscht werden können. Sie enthalten auch die Funktionen, mit denen der Amazon S3 S3-Bucket gelöscht werden kann, der für den Cluster und das CodeBuild Projekt erstellt wurde. Es gibt auch Rollen für AWS Batch Ressourcen, Instances und Jobs.

Weitere Informationen zu IAM finden Sie unter https://aws.amazon.com/iam/und https://docs.aws.amazon.com/iam/.

AWS Lambda

AWS Lambda (Lambda) führt die Funktionen aus, die die Erstellung von Docker-Images orchestrieren. Lambda verwaltet auch die Bereinigung von benutzerdefinierten Cluster-Ressourcen wie Docker-Images, die im Amazon ECR-Repository und in Amazon S3 gespeichert sind.

Weitere Informationen zu Lambda finden Sie unter https://aws.amazon.com/lambda/und https://docs.aws.amazon.com/lambda/.

Amazon DCV

Amazon DCV ist ein leistungsstarkes Remote-Display-Protokoll, das eine sichere Möglichkeit bietet, Remote-Desktops und Anwendungsstreaming über unterschiedliche Netzwerkbedingungen auf jedem Gerät bereitzustellen. Amazon DCV wird verwendet, wenn die `dcv_settings` Einstellung angegeben ist und sich auf einen `[dcv]`Abschnitt bezieht. Support für Amazon DCV wurde in AWS ParallelCluster Version 2.5.0 hinzugefügt.

Weitere Informationen zu Amazon DCV finden Sie unter https://aws.amazon.com/hpc/dcv/ und. https://docs.aws.amazon.com/dcv/

Amazon Route 53

Amazon Route 53 (Route 53) wird verwendet, um Hostzonen mit Hostnamen und vollqualifizierten Domainnamen für jeden der Rechenknoten zu erstellen.

Weitere Informationen zu Route 53 finden Sie unter https://aws.amazon.com/route53/und https://docs.aws.amazon.com/route53/.

Amazon Simple Notification Service

> **Note**
>
> Dieser Abschnitt gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von Amazon Simple Notification Service AWS ParallelCluster nicht unterstützt.

Amazon Simple Notification Service (Amazon SNS) empfängt Benachrichtigungen von Auto Scaling. Diese Ereignisse werden als Lebenszyklusereignisse bezeichnet und generiert, wenn eine Instance in einer Auto Scaling Scaling-Gruppe gestartet oder beendet wird. Darin AWS ParallelCluster ist das Amazon SNS SNS-Thema für die Auto Scaling Scaling-Gruppe für eine Amazon SQS SQS-Warteschlange abonniert.

Amazon SNS wird nicht mit AWS Batch Clustern verwendet.

Weitere Informationen zu Amazon SNS finden Sie unter https://aws.amazon.com/sns/und https://docs.aws.amazon.com/sns/.

Amazon Simple Queue Service

> **Note**
>
> Dieser Abschnitt gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von Amazon Simple Queue Service AWS ParallelCluster nicht unterstützt.

Amazon Simple Queue Service (Amazon SQS) speichert Benachrichtigungen, die von Auto Scaling gesendet wurden, Benachrichtigungen, die über Amazon SNS gesendet wurden, und Benachrichtigungen, die von den Rechenknoten gesendet wurden. Amazon SQS entkoppelt das Senden von Benachrichtigungen vom Empfangen von Benachrichtigungen. Auf diese Weise kann der Hauptknoten Benachrichtigungen über einen Abfrageprozess verarbeiten. In diesem Prozess führt der Hauptknoten Amazon aus SQSwatcher und fragt die Warteschlange ab. Auto Scaling und die Rechenknoten senden Nachrichten an die Warteschlange.

Amazon SQS wird nicht mit AWS Batch Clustern verwendet.

Weitere Informationen zu Amazon SQS finden Sie unter https://aws.amazon.com/sqs/und https://docs.aws.amazon.com/sqs/.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) speichert AWS ParallelCluster Vorlagen, die sich jeweils darin befinden AWS-Region. AWS ParallelCluster kann so konfiguriert werden, dass CLI/SDK-Tools Amazon S3 verwenden können.

Wenn Sie AWS Batch Cluster verwenden, wird ein Amazon S3 S3-Bucket in Ihrem Konto zum Speichern verwandter Daten verwendet. Im Bucket werden beispielsweise Artefakte gespeichert, die entstehen, wenn ein Docker-Image und Skripts aus eingereichten Jobs erstellt werden.

Weitere Informationen erhalten Sie unter https://aws.amazon.com/s3/ und https://docs.aws.amazon.com/s3/.

Amazon VPC

Amazon VPC definiert ein Netzwerk, das von den Knoten in Ihrem Cluster verwendet wird. Die VPC-Einstellungen für den Cluster sind im `[vpc]`Abschnitt definiert.

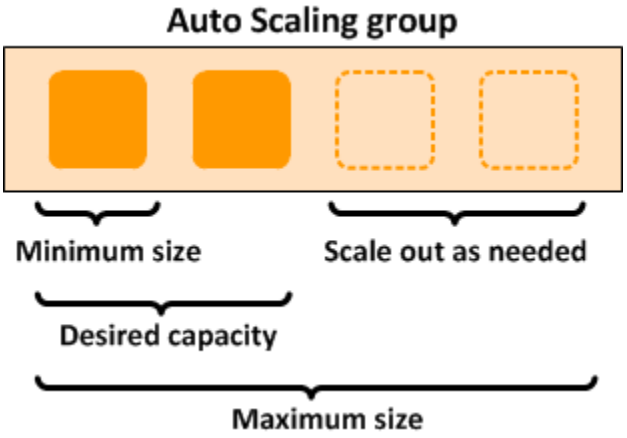
Weitere Informationen zu Amazon VPC finden Sie unter https://aws.amazon.com/vpc/und https://docs.aws.amazon.com/vpc/.

AWS ParallelCluster Auto Scaling

> **Note**
>
> Dieser Abschnitt gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler. Sie können sie weiterhin in Versionen bis einschließlich 2.11.4 verwenden, sie haben jedoch keinen Anspruch auf future Updates oder Support bei der Fehlerbehebung durch die AWS Service- und AWS Support-Teams.
> Ab AWS ParallelCluster Version 2.9.0 wird Auto Scaling nicht für die Verwendung mit unterstützt Slurm Workload Manager (Slurm). Um mehr darüber zu erfahren Slurm und die Skalierung mehrerer Warteschlangen finden Sie unterTutorial zum Modus mit mehreren Warteschlangen.

Die in diesem Thema beschriebene Auto Scaling-Strategie gilt für HPC-Cluster, die mit einer der folgenden Optionen bereitgestellt werden Son of Grid Engine (SGE) oder Torque Resource Manager (Torque). Wenn es mit einem dieser Scheduler bereitgestellt wird, AWS ParallelCluster implementiert es die Skalierungsfunktionen, indem es die Auto Scaling Scaling-Gruppe der Rechenknoten verwaltet und dann die Scheduler-Konfiguration nach Bedarf ändert. Bei HPC-Clustern, die auf basieren AWS Batch, AWS ParallelCluster stützt sich auf die elastischen Skalierungsfunktionen, die vom AWS Managed Job Scheduler bereitgestellt werden. Weitere Informationen finden Sie unter Was ist Amazon EC2 Auto Scaling im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Cluster, die mit bereitgestellt werden, AWS ParallelCluster sind in mehrfacher Hinsicht elastisch. Die Einstellung von `initial_queue_size` gibt den Mindestgrößenwert der ComputeFleet Auto Scaling Scaling-Gruppe sowie den gewünschten Kapazitätswert an. Die Einstellung von `max_queue_size` gibt den maximalen Größenwert der ComputeFleet Auto Scaling Scaling-Gruppe an.



Hochskalieren

Jede Minute wird auf dem Hauptknoten ein sogenannter Prozess `jobwatcher`ausgeführt. Er wertet die aktuelle Anzahl von Instances aus, die für die ausstehenden Aufgaben in der Warteschlange erforderlich sind. Wenn die Gesamtzahl der ausgelasteten Knoten und der angeforderten Knoten den aktuellen gewünschten Wert in der Auto Scaling Scaling-Gruppe übersteigt, werden weitere Instances hinzugefügt. Wenn Sie weitere Jobs einreichen, wird die Warteschlange neu bewertet und die Auto Scaling Scaling-Gruppe bis zum angegebenen `max_queue_size` Wert aktualisiert.

Mit einem SGE Scheduler, für die Ausführung jedes Jobs sind mehrere Steckplätze erforderlich (ein Steckplatz entspricht einer Verarbeitungseinheit, z. B. einer vCPU). Zum Auswerten der Anzahl der Instances, die für die aktuell ausstehenden Aufgaben erforderlich sind, teilt der `jobwatcher` die Gesamtzahl der angeforderten Slots durch die Kapazität eines einzelnen Datenverarbeitungsknotens. Die Kapazität eines Rechenknotens, die der Anzahl der verfügbaren v entspricht, CPUs hängt vom EC2 Amazon-Instance-Typ ab, der in der Cluster-Konfiguration angegeben ist.

Mit Slurm (vor AWS ParallelCluster Version 2.9.0) und Torque Bei Schedulern kann jeder Job je nach den Umständen sowohl eine Anzahl von Knoten als auch eine Anzahl von Steckplätzen für jeden Knoten benötigen. Für jede Anforderung bestimmt der `jobwatcher` die Anzahl der Datenverarbeitungsknoten, die erforderlich sind, um die neuen Anforderungen an die Rechenleistung zu erfüllen. Gehen wir beispielsweise von einem Cluster mit `c5.2xlarge` (8 vCPU) als Datenverarbeitungs-Instance-Typ und drei in der Warteschlange ausstehenden Aufgaben mit den folgenden Anforderungen aus:

- Aufgabe 1: 2 Knoten mit je 4 Slots
- Aufgabe 2: 3 Knoten mit 2 Slots
- Aufgabe 3: 1 Knoten mit 4 Slots

In diesem Beispiel sind drei neue Recheninstanzen in der Auto Scaling Scaling-Gruppe `jobwatcher` erforderlich, um die drei Jobs zu bedienen.

Aktuelle Einschränkung: Die auto Skalierungslogik berücksichtigt teilweise ausgelastete ausgelastete Knoten nicht. Beispielsweise gilt ein Knoten, auf dem ein Job ausgeführt wird, als ausgelastet, auch wenn es leere Steckplätze gibt.

Herunterskalieren

Auf jedem Datenverarbeitungsknoten wird ein Prozess namens `nodewatcher` ausgeführt, der die Leerlaufzeit des Knotens auswertet. Eine Instance ist beendet, wenn beide der folgenden Bedingungen erfüllt sind:

- Eine Instance hat keine Aufgaben für einen bestimmten Zeitraum länger als die `scaledown_idletime` (die Standardeinstellung ist 10 Minuten)
- Es gibt keine ausstehenden Aufgaben im Cluster

Um eine Instance zu beenden, `nodewatcher` ruft die TerminateInstanceInAutoScalingGroupAPI-Operation auf, die eine Instance entfernt, wenn die Größe der Auto Scaling Scaling-Gruppe mindestens der Auto Scaling Scaling-Gruppen-Mindestgröße entspricht. Dieser Vorgang skaliert einen Clusters nach unten, ohne dass dies Auswirkungen auf aktive Aufgaben hätte. Es ermöglicht auch einen elastischen Cluster mit einer festen Basisanzahl von Instanzen.

Statischer Cluster

Der Wert von Auto Scaling ist bei HPC der gleiche wie bei anderen Workloads. Der einzige Unterschied besteht darin, dass AWS ParallelCluster Code aufweist, mit dem er intelligenter interagiert. Wenn beispielsweise ein statischer Cluster erforderlich ist, setzen Sie die `max_queue_size` Parameter `initial_queue_size` und auf die exakte Größe des Clusters, der erforderlich ist, und setzen dann den `maintain_initial_size` Parameter auf true. Dies führt dazu, dass die ComputeFleet Auto Scaling Scaling-Gruppe denselben Wert für minimale, maximale und gewünschte Kapazität hat.

Tutorials

Die folgenden Tutorials zeigen Ihnen, wie Sie mit einigen häufig auftretenden Aufgaben beginnen können AWS ParallelCluster, und bieten Anleitungen zu bewährten Verfahren.

Themen

- Du führst deinen ersten Job am AWS ParallelCluster
- Ein benutzerdefiniertes AWS ParallelCluster AMI erstellen
- Einen MPI-Job mit AWS ParallelCluster einem Scheduler ausführen awsbatch
- Festplattenverschlüsselung mit einem benutzerdefinierten KMS-Schlüssel
- Tutorial zum Modus mit mehreren Warteschlangen

Du führst deinen ersten Job am AWS ParallelCluster

In diesem Tutorial erfahren Sie, wie Sie Ihren ersten Hello World-Job am ausführen AWS ParallelCluster.

Voraussetzungen

- AWS ParallelCluster ist installiert.
- Das AWS CLI ist installiert und konfiguriert.
- Sie haben ein EC2 key pair.
- Sie haben eine IAM-Rolle mit den für die Ausführung der `pcluster` CLI erforderlichen Berechtigungen.

Überprüfen der Installation

Zunächst überprüfen wir, ob die Installation und Konfiguration korrekt AWS ParallelCluster ist.

```
$ pcluster version
```

Dies gibt die laufende Version von zurück AWS ParallelCluster. Wenn Sie in der Ausgabe eine Meldung zur Konfiguration erhalten, müssen Sie zur Konfiguration Folgendes ausführen AWS ParallelCluster:

```
$ pcluster configure
```

Erstellen Sie Ihren ersten Cluster

Jetzt ist es an der Zeit, Ihren ersten Cluster zu erstellen. Da die Workload für dieses Tutorial nicht leistungsintensiv ist, können wir die Standard-Instance-Größe `t2.micro` verwenden. (Für Produktions-Workloads wählen Sie eine Instance-Größe, die Ihren Anforderungen am ehesten entspricht.)

Nennen Sie Ihren Cluster „hello-world“.

```
$ pcluster create hello-world
```

Wenn der Cluster erstellt wurde, wird eine Ausgabe ähnlich der Folgenden angezeigt:

```
Starting: hello-world
Status: parallelcluster-hello-world - CREATE_COMPLETE
MasterPublicIP = 54.148.x.x
ClusterUser: ec2-user
MasterPrivateIP = 192.168.x.x
GangliaPrivateURL = http://192.168.x.x/ganglia/
GangliaPublicURL = http://54.148.x.x/ganglia/
```

Die Nachricht `CREATE_COMPLETE` zeigt, dass der Cluster erfolgreich erstellt wurde. Die Ausgabe liefert uns auch die öffentlichen und privaten IP-Adressen unseres Hauptknotens. Diese IP-Adresse benötigen wir für die Anmeldung.

Loggen Sie sich in Ihren Headnode ein

Verwenden Sie Ihre OpenSSH-PEM-Datei, um sich bei Ihrem Headnode anzumelden.

```
pcluster ssh hello-world -i /path/to/keyfile.pem
```

Sobald Sie angemeldet sind, führen Sie den Befehl `qhost` aus, um zu bestätigen, dass Ihre Datenverarbeitungsknoten eingerichtet und konfiguriert sind.

```
$ qhost
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR  LOAD  MEMTOT  MEMUSE  SWAPTO
 SWAPUS
```

```
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -
      -
ip-192-168-1-125        lx-amd64        2    1    2    2  0.15    3.7G  130.8M 1024.0M
    0.0
ip-192-168-1-126        lx-amd64        2    1    2    2  0.15    3.7G  130.8M 1024.0M
    0.0
```

Sie Ausgabe zeigt, dass wir zwei Datenverarbeitungsknoten in unserem Cluster haben. Beiden stehen zwei Threads zur Verfügung.

Führen Sie Ihren ersten Job aus mit SGE

> **Note**
>
> Dieses Beispiel gilt nur für AWS ParallelCluster Versionen bis einschließlich Version 2.11.4. Ab Version 2.11.5 wird die Verwendung von AWS ParallelCluster nicht unterstützt SGE or Torque Scheduler.

Als Nächstes erstellen wir eine Aufgabe, die einen Moment inaktiv ist und dann ihren eigenen Hostnamen ausgibt.

Erstellen Sie eine Datei mit dem Namen `hellojob.sh` und den folgenden Inhalten:

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

Als Nächstes übermitteln Sie die Aufgabe mit `qsub` und überprüfen, dass sie ausgeführt wird.

```
$ qsub hellojob.sh
Your job 1 ("hellojob.sh") has been submitted
```

Jetzt können Sie Ihre Warteschlange anzeigen und den Status der Aufgabe überprüfen.

```
$ qstat
job-ID  prior   name       user         state submit/start at     queue
          slots ja-task-ID
----------------------------------------------------------------------------------------------
```

```
     1 0.55500 hellojob.s ec2-user     r     03/24/2015 22:23:48
 all.q@ip-192-168-1-125.us-west     1
```

Die Ausgabe zeigt, dass die Aufgabe zurzeit ausgeführt wird. Warten Sie 30 Sekunden, bis die Aufgabe beendet wird, und führen Sie `qstat` dann erneut aus.

```
$ qstat
$
```

Jetzt, wo sich keine Aufgaben in der Warteschlange befinden, können wir die Ausgabe in unserem aktuellen Verzeichnis überprüfen.

```
$ ls -l
total 8
-rw-rw-r-- 1 ec2-user ec2-user 48 Mar 24 22:34 hellojob.sh
-rw-r--r-- 1 ec2-user ec2-user  0 Mar 24 22:34 hellojob.sh.e1
-rw-r--r-- 1 ec2-user ec2-user 34 Mar 24 22:34 hellojob.sh.o1
```

In der Ausgabe sehen wir in unserem Job-Skript eine Datei „`e1`“ und eine Datei „`o1`“. Da die `e1` Datei leer ist, gab es keine Ausgabe auf stderr. Wenn wir die Datei `o1` anzeigen, können wir die Ausgabe unserer Aufgabe anzeigen.

```
$ cat hellojob.sh.o1
Hello World from ip-192-168-1-125
```

Die Ausgabe zeigt auch, dass die Aufgabe auf Instance `ip-192-168-1-125` erfolgreich ausgeführt wurde.

Weitere Informationen zum Erstellen und Verwenden von Clustern finden Sie unter. Bewährte Methoden

Ein benutzerdefiniertes AWS ParallelCluster AMI erstellen

> **Important**
>
> Wir empfehlen nicht, ein benutzerdefiniertes AMI als Ansatz für die Anpassung zu AWS ParallelCluster erstellen.
> Dies liegt daran, dass Sie, nachdem Sie Ihr eigenes AMI erstellt haben, in future Versionen von keine Updates oder Bugfixes mehr erhalten AWS ParallelCluster. Wenn Sie ein

benutzerdefiniertes AMI erstellen, müssen Sie außerdem die Schritte, die Sie zur Erstellung Ihres benutzerdefinierten AMI verwendet haben, mit jeder neuen AWS ParallelCluster Version wiederholen.

Bevor Sie weiterlesen, empfehlen wir Ihnen, zunächst den Abschnitt Benutzerdefinierte Bootstrap-Aktionen zu lesen, um festzustellen, ob die Änderungen, die Sie vornehmen möchten, skriptgesteuert werden können und in future Versionen unterstützt werden können. AWS ParallelCluster

Auch wenn die Erstellung eines benutzerdefinierten AMI nicht ideal ist (aus den oben genannten Gründen), gibt es immer noch Szenarien, für die die Erstellung eines benutzerdefinierten AMI erforderlich AWS ParallelCluster ist. Dieses Tutorial führt Sie durch den Prozess der Erstellung eines benutzerdefinierten AMI für diese Szenarien.

> **Note**
>
> Ab AWS ParallelCluster Version 2.6.1 werden die meisten Installationsrezepte beim Starten von Knoten standardmäßig übersprungen. Dies dient dazu, die Startzeiten zu verbessern. Um alle Installationsrezepte für eine bessere Abwärtskompatibilität auf Kosten der Startzeiten auszuführen, fügen Sie `"skip_install_recipes" : "no"` den `cluster` Schlüssel in der `extra_json` Einstellung hinzu. Zum Beispiel:
>
> ```
> extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
> ```

Voraussetzungen

- AWS ParallelCluster ist installiert.
- Das AWS CLI ist installiert und konfiguriert.
- Sie haben ein EC2 key pair.
- Sie haben eine IAM-Rolle mit den für die Ausführung der `pcluster` CLI erforderlichen Berechtigungen.

So passen Sie das AWS ParallelCluster AMI an

Es gibt drei Möglichkeiten, ein benutzerdefiniertes AWS ParallelCluster AMI zu verwenden, die in den nächsten Abschnitten beschrieben werden. Bei zwei dieser drei Methoden müssen Sie ein neues

AMI erstellen, das unter Ihrem verfügbar ist AWS-Konto. Die dritte Methode (Use a Custom AMI at Runtime) erfordert nicht, dass Sie etwas im Voraus erstellen, erhöht jedoch das Risiko für die Bereitstellung. Wählen Sie die Methode, die Ihren Anforderungen am besten entspricht.

Ändern eines -AMIs

Dies ist die sicherste und am meisten empfohlene Methode. Da das AWS ParallelCluster Basis-AMI häufig mit neuen Versionen aktualisiert wird, verfügt dieses AMI über alle Komponenten, die für AWS ParallelCluster den Betrieb bei der Installation und Konfiguration erforderlich sind. Sie können es als Ausgangsbasis verwenden.

New EC2 console

1. Suchen Sie in der AWS ParallelCluster AMI-Liste nach dem AMI, das dem spezifischen entspricht AWS-Region , das Sie verwenden. Die von Ihnen gewählte AMI-Liste muss mit der Version übereinstimmen AWS ParallelCluster , die Sie verwenden. Führen Sie `pcluster version` aus, um die Version zu überprüfen. Für AWS ParallelCluster Version 2.11.9 wechseln Sie zu https://github.com/aws/aws-.txt. parallelcluster/blob/v2.11.9/amis Um eine andere Version auszuwählen, verwenden Sie denselben Link, wählen Sie die Schaltfläche Tag: 2.11.9, wählen Sie die Registerkarte Tags und wählen Sie dann die entsprechende Version aus.
2. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
3. Wählen Sie im EC2 Amazon-Dashboard die Option Launch instance aus.
4. Wählen Sie unter Anwendungs- und Betriebssystemimages die Option Mehr durchsuchen aus AMIs AMIs, navigieren Sie zu Community und geben Sie die AWS ParallelCluster AMI-ID für Sie AWS-Region in das Suchfeld ein.
5. Wählen Sie das AMI aus, wählen Sie Ihren Instance-Typ und Ihre Eigenschaften, wählen Sie Ihr Schlüsselpaar und starten Sie die Instance.
6. Melden Sie sich bei Ihrer Instance mithilfe des Betriebssystembenutzers und Ihres SSH-Schlüssels an. Für weitere Informationen navigieren Sie zu Instances, wählen Sie die neue Instance aus und klicken Sie auf Connect.
7. Passen Sie Ihre Instance nach Bedarf an.
8. Führen Sie den folgenden Befehl aus, um Ihre Instance für die AMI-Erstellung vorzubereiten:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Navigieren Sie zu Instances, wählen Sie die neue Instance aus, wählen Sie Instance state und Stop instance aus.
10. Erstellen Sie mit der EC2 Konsole oder dem AWS CLI Create-Image ein neues AMI von der Instance aus.

    Von der Konsole EC2

    a. Wählen Sie im Navigationsbereich Instances aus.
    b. Wählen Sie die Instanz aus, die Sie erstellt und geändert haben.
    c. Wählen Sie unter Aktionen die Option Bild und Vorlagen und dann Bild erstellen aus.
    d. Wählen Sie Image erstellen aus.
11. Geben Sie die neue AMI-ID in das Feld custom_ami in Ihrer Cluster-Konfiguration ein.

Old EC2 console

1. Suchen Sie in der AWS ParallelCluster AMI-Liste nach dem AMI, das dem spezifischen entspricht AWS-Region , das Sie verwenden. Die von Ihnen gewählte AMI-Liste muss mit der Version übereinstimmen AWS ParallelCluster , die Sie verwenden. Führen Sie `pcluster version` aus, um die Version zu überprüfen. Für AWS ParallelCluster Version 2.11.9 wechseln Sie zu https://github.com/aws/aws-.txt. parallelcluster/blob/v2.11.9/amis Um eine andere Version auszuwählen, verwenden Sie denselben Link, wählen Sie die Schaltfläche Tag: 2.11.9, wählen Sie die Registerkarte Tags und wählen Sie dann die entsprechende Version aus.
2. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
3. Wählen Sie im EC2 Amazon-Dashboard die Option Launch instance aus.
4. Wählen Sie Community AMIs, suchen Sie nach der AWS ParallelCluster AMI-ID und wählen Sie sie aus.
5. Wählen Sie Ihren Instance-Typ und klicken Sie auf Weiter: Instance-Details konfigurieren oder Überprüfen und starten, um Ihre Instance zu starten.
6. Wählen Sie Launch, wählen Sie Ihr Schlüsselpaar und dann Launch Instances aus.
7. Melden Sie sich bei Ihrer Instance mithilfe des Betriebssystembenutzers und Ihres SSH-Schlüssels an. Für weitere Informationen navigieren Sie zu Instances, wählen Sie die neue Instance aus und klicken Sie auf Connect.
8. Passen Sie Ihre Instance nach Bedarf an.

9. Führen Sie den folgenden Befehl aus, um Ihre Instance für die AMI-Erstellung vorzubereiten:

```
sudo /usr/local/sbin/ami_cleanup.sh
```

10. Navigieren Sie zu Instances, wählen Sie die neue Instance aus, wählen Sie Instance State und Stop
11. Erstellen Sie mit der EC2 Konsole oder dem AWS CLI Create-Image ein neues AMI von der Instance aus.

    Von der Konsole EC2

    a. Wählen Sie im Navigationsbereich Instances aus.
    b. Wählen Sie die Instanz aus, die Sie erstellt und geändert haben.
    c. Wählen Sie unter Aktionen die Option Image und dann Create Image aus.
    d. Wählen Sie Image erstellen aus.
12. Geben Sie die neue AMI-ID in das Feld custom_ami in Ihrer Cluster-Konfiguration ein.

Erstellen Sie ein benutzerdefiniertes AWS ParallelCluster AMI

Wenn Sie ein angepasstes AMI und bereits eingerichteter Software haben, können Sie die für AWS ParallelCluster erforderlichen Änderungen zusätzlich anwenden.

1. Installieren Sie Folgendes zusammen mit der AWS ParallelCluster CLI in Ihrem lokalen System:
   - Packer: Suchen Sie auf der Packer-Website nach der neuesten Version für Ihr Betriebssystem und installieren Sie sie. Die Version muss mindestens 1.4.0 sein, aber die neueste Version wird empfohlen. Stellen Sie sicher, dass der `packer` Befehl in Ihrem PATH verfügbar ist.

> **Note**
>
> Vor AWS ParallelCluster Version 2.8.0 musste Berkshelf (das mit using installiert wirdgem `install berkshelf`) verwendet werden. `pcluster createami`

2. Konfigurieren Sie Ihre AWS-Konto Anmeldeinformationen so, dass Packer in Ihrem Namen AWS API-Operationen aufrufen kann. Der Mindestsatz der für Packer erforderlichen Berechtigungen wird im Abschnitt IAM-Aufgabe oder Instance-Rolle des Themas Amazon AMI Builder in der Packer-Dokumentation behandelt.

3. Verwenden Sie den Befehl `createami` in der AWS ParallelCluster CLI, um ausgehend von dem, das Sie als Basis angeben, ein AWS ParallelCluster AMI zu erstellen:

```
pcluster createami --ami-id <BASE_AMI> --os <BASE_AMI_OS>
```

> **Important**
>
> Sie sollten kein AWS ParallelCluster AMI aus einem laufenden Cluster wie *<BASE_AMI>* für den `createami` Befehl verwenden. Andernfalls schlägt der Befehl fehl.

   Weitere Parameter finden Sie unter`pcluster createami`.

4. Mit dem Befehl in Schritt 4 wird Packer ausgeführt, der insbesondere Folgendes ausführt:
   a. Startet eine Instance mit dem angegebenen AMI als Ausgangsbasis.
   b. Wendet das AWS ParallelCluster Kochbuch auf die Instanz an, um die entsprechende Software zu installieren und andere notwendige Konfigurationsaufgaben durchzuführen.
   c. Hält die Instance an.
   d. Erstellt ein neues AMI anhand der Instance.
   e. Beendet die Instance, nachdem das AMI erstellt wurde.
   f. Gibt die neue AMI-ID-Zeichenfolge aus, die zum Erstellen Ihres Clusters verwendet wird.
5. Zum Erstellen Ihres Clusters geben Sie die AMI-ID in das Feld custom_ami in Ihrer Cluster-Konfiguration ein.

> **Note**
>
> Der Instance-Typ, der zum Erstellen eines benutzerdefinierten AWS ParallelCluster AMI verwendet wird, ist`t2.xlarge`. Dieser Instance-Typ kommt nicht für das AWS kostenlose Kontingent in Frage, sodass Ihnen alle Instances in Rechnung gestellt werden, die beim Erstellen dieses AMI erstellt werden.

Verwenden eines benutzerdefinierten AMIs zur Laufzeit

> **Warning**
>
> Um das Risiko der Verwendung eines AMI zu vermeiden, das nicht kompatibel mit ist AWS ParallelCluster, empfehlen wir, diese Methode nicht zu verwenden.
> Wenn Compute-Knoten zur Laufzeit mit potenziell ungetesteten AMIs Knoten gestartet werden, können Inkompatibilitäten mit der Runtime-Installation der AWS ParallelCluster erforderlichen Software AWS ParallelCluster dazu führen, dass sie nicht mehr funktionieren.

Wenn Sie im Voraus nichts erstellen möchten, können Sie Ihr AMI verwenden und AWS ParallelCluster aus diesem AMI ein erstellen.

Bei dieser Methode dauert es länger, AWS ParallelCluster bis das erstellt ist, da die gesamte Software, die bis zur AWS ParallelCluster Erstellung des Clusters benötigt wird, installiert sein muss. Darüber hinaus dauert die Skalierung auch länger.

- Geben Sie die AMI-ID in das Feld custom_ami innerhalb Ihrer Cluster-Konfiguration ein.

Einen MPI-Job mit AWS ParallelCluster einem Scheduler ausführen `awsbatch`

Dieses Tutorial führt Sie durch das Ausführen einer MPI-Aufgabe mit `awsbatch` als Scheduler.

Voraussetzungen

- AWS ParallelCluster ist installiert.
- Das AWS CLI ist installiert und konfiguriert.
- Sie haben ein EC2 key pair.
- Sie haben eine IAM-Rolle mit den für die Ausführung der `pcluster` CLI erforderlichen Berechtigungen.

Erstellen des Clusters

Zuerst erstellen wir eine Konfiguration für einen Cluster, der `awsbatch` als Scheduler verwendet. Stellen Sie sicher, dass Sie die fehlenden Daten im Abschnitt `vpc` und im Feld `key_name` um die Ressourcen ergänzen, die Sie bei der Konfiguration erstellt haben.

```
[global]
sanity_check = true

[aws]
aws_region_name = us-east-1

[cluster awsbatch]
base_os = alinux
# Replace with the name of the key you intend to use.
key_name = key-#######
vpc_settings = my-vpc
scheduler = awsbatch
compute_instance_type = optimal
min_vcpus = 2
desired_vcpus = 2
max_vcpus = 24

[vpc my-vpc]
# Replace with the id of the vpc you intend to use.
vpc_id = vpc-#######
# Replace with id of the subnet for the Head node.
master_subnet_id = subnet-#######
# Replace with id of the subnet for the Compute nodes.
# A NAT Gateway is required for MNP.
compute_subnet_id = subnet-#######
```

Sie können nun mit der Erstellung des Clusters beginnen. Rufen wir unseren Cluster *`awsbatch-tutorial`* an.

```
$ pcluster create -c /path/to/the/created/config/aws_batch.config -t awsbatch awsbatch-tutorial
```

Wenn der Cluster erstellt wurde, wird eine Ausgabe ähnlich der Folgenden angezeigt:

```
Beginning cluster creation for cluster: awsbatch-tutorial
```

```
Creating stack named: parallelcluster-awsbatch
Status: parallelcluster-awsbatch - CREATE_COMPLETE
MasterPublicIP: 54.160.xxx.xxx
ClusterUser: ec2-user
MasterPrivateIP: 10.0.0.15
```

Loggen Sie sich in Ihren Hauptknoten ein

Die AWS ParallelCluster Batch-CLI-Befehle sind alle auf dem Client-Computer verfügbar, auf dem sie AWS ParallelCluster installiert sind. Wir werden jedoch eine SSH-Verbindung zum Hauptknoten herstellen und die Jobs von dort aus einreichen. Auf diese Weise können wir das NFS-Volume nutzen, das vom Head und allen Docker-Instances, die Jobs ausführen, gemeinsam genutzt wird. AWS Batch

Verwenden Sie Ihre SSH-PEM-Datei, um sich bei Ihrem Headnode anzumelden.

```
$ pcluster ssh awsbatch-tutorial -i /path/to/keyfile.pem
```

Wenn Sie angemeldet sind, führen Sie die Befehle aus `awsbqueues` und `awsbhosts` zeigen Sie die konfigurierte AWS Batch Warteschlange und die laufenden Amazon ECS-Instances an.

```
[ec2-user@ip-10-0-0-111 ~]$ awsbqueues
jobQueueName                      status
---------------------------------  --------
parallelcluster-awsbatch-tutorial  VALID

[ec2-user@ip-10-0-0-111 ~]$ awsbhosts
ec2InstanceId        instanceType    privateIpAddress    publicIpAddress
 runningJobs
-------------------  --------------  ------------------  -----------------
 -------------
i-0d6a0c8c560cd5bed  m4.large        10.0.0.235          34.239.174.236
 0
```

Wie Sie in der Ausgabe sehen können, wird ein einzelner Host ausgeführt. Diese Tatsache ist auf den Wert zurückzuführen, den wir in der Konfiguration für `min_vcpus` ausgewählt haben. Wenn Sie zusätzliche Details zur AWS Batch Warteschlange und zu den Hosts anzeigen möchten, fügen Sie dem Befehl das `-d` Flag hinzu.

Führen Sie Ihren ersten Job aus mit AWS Batch

Vor dem Wechsel zu MPI erstellen wir eine Dummy-Aufgabe, die einen Moment inaktiv ist und anschließend ihren eigenen Hostnamen ausgibt und den als Parameter übergebenen Namen begrüßt.

Erstellen Sie eine Datei mit dem Namen "hellojob.sh" und folgendem Inhalt.

```
#!/bin/bash

sleep 30
echo "Hello $1 from $HOSTNAME"
echo "Hello $1 from $HOSTNAME" > "/shared/secret_message_for_${1}_by_
${AWS_BATCH_JOB_ID}"
```

Als Nächstes übermitteln Sie die Aufgabe mit `awsbsub` und stellen sicher, dass sie ausgeführt wird.

```
$ awsbsub -jn hello -cf hellojob.sh Luca
Job 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2 (hello) has been submitted.
```

Zeigen Sie Ihre Warteschlange an und prüfen Sie den Status der Aufgabe.

```
$ awsbstat
jobId                                    jobName      status    startedAt
 stoppedAt    exitCode
-------------------------------------  -----------  --------  -------------------
 -----------  ----------
6efe6c7c-4943-4c1a-baf5-edbfeccab5d2  hello        RUNNING   2018-11-12 09:41:29  -
        -
```

Die Ausgabe enthält detaillierte Informationen zur Aufgabe.

```
$ awsbstat 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
jobId                    : 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
jobName                  : hello
createdAt                : 2018-11-12 09:41:21
startedAt                : 2018-11-12 09:41:29
stoppedAt                : -
status                   : RUNNING
statusReason             : -
jobDefinition            : parallelcluster-exampleBatch:1
```

```
jobQueue                 : parallelcluster-exampleBatch
command                  : /bin/bash -c 'aws s3 --region us-east-1 cp
 s3://amzn-s3-demo-bucket/batch/job-hellojob_sh-1542015680924.sh /tmp/batch/job-
hellojob_sh-1542015680924.sh; bash /tmp/batch/job-hellojob_sh-1542015680924.sh Luca'
exitCode                 : -
reason                   : -
vcpus                    : 1
memory[MB]               : 128
nodes                    : 1
logStream                : parallelcluster-exampleBatch/default/c75dac4a-5aca-4238-
a4dd-078037453554
log                      : https://console.aws.amazon.com/cloudwatch/home?region=us-
east-1#logEventViewer:group=/aws/batch/job;stream=parallelcluster-exampleBatch/default/
c75dac4a-5aca-4238-a4dd-078037453554
--------------------------
```

Die Aufgabe befindet sich derzeit im Status `RUNNING`. Warten Sie 30 Sekunden, bis die Aufgabe beendet wird, und führen Sie `awsbstat` dann erneut aus.

```
$ awsbstat
jobId                                   jobName      status    startedAt
 stoppedAt    exitCode
------------------------------------  -----------  --------  -------------------
 -----------  ----------
```

Jetzt können Sie sehen, dass die Aufgabe im Status `SUCCEEDED` ist.

```
$ awsbstat -s SUCCEEDED
jobId                                   jobName      status     startedAt
 stoppedAt              exitCode
------------------------------------  -----------  ---------  -------------------
 -------------------  ----------
6efe6c7c-4943-4c1a-baf5-edbfeccab5d2  hello        SUCCEEDED  2018-11-12 09:41:29
 2018-11-12 09:42:00           0
```

Weil sich keine Aufgaben in der Warteschlange befinden, können wir die Ausgabe mit dem Befehl `awsbout` überprüfen.

```
$ awsbout 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
2018-11-12 09:41:29: Starting Job 6efe6c7c-4943-4c1a-baf5-edbfeccab5d2
download: s3://amzn-s3-demo-bucket/batch/job-hellojob_sh-1542015680924.sh to tmp/batch/
job-hellojob_sh-1542015680924.sh
```

```
2018-11-12 09:42:00: Hello Luca from ip-172-31-4-234
```

Wir können sehen, dass unsere Aufgabe auf der Instance „ip-172-31-4-234“ erfolgreich ausgeführt wurde.

Im `/shared`-Verzeichnis finden Sie eine an Sie gerichtete geheime Nachricht.

Alle verfügbaren Funktionen, die nicht Teil dieses Tutorials sind, finden Sie in der AWS ParallelCluster -Batch-CLI-Dokumentation. Wenn Sie bereit sind, mit dem Tutorial fortzufahren, machen wir weiter und erfahren, wie eine MPI-Aufgabe übermittelt wird.

Ausführen eines MPI-Auftrags in einer parallelen Umgebung mit mehreren Knoten

Erstellen Sie eine Datei in dem `/shared` Verzeichnis mit dem Namen, während Sie noch im Hauptknoten angemeldet sind`mpi_hello_world.c`. Fügen Sie der Datei das folgende MPI-Programm hinzu:

```
// Copyright 2011 www.mpitutorial.com
//
// An intro MPI hello world program that uses MPI_Init, MPI_Comm_size,
// MPI_Comm_rank, MPI_Finalize, and MPI_Get_processor_name.
//
#include <mpi.h>
#include <stdio.h>
#include <stddef.h>

int main(int argc, char** argv) {
  // Initialize the MPI environment. The two arguments to MPI Init are not
  // currently used by MPI implementations, but are there in case future
  // implementations might need the arguments.
  MPI_Init(NULL, NULL);

  // Get the number of processes
  int world_size;
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);

  // Get the rank of the process
  int world_rank;
  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
```

```
  // Get the name of the processor
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int name_len;
  MPI_Get_processor_name(processor_name, &name_len);

  // Print off a hello world message
  printf("Hello world from processor %s, rank %d out of %d processors\n",
         processor_name, world_rank, world_size);

  // Finalize the MPI environment. No more MPI calls can be made after this
  MPI_Finalize();
}
```

Speichern Sie jetzt den folgenden Code unter `submit_mpi.sh`:

```
#!/bin/bash
echo "ip container: $(/sbin/ip -o -4 addr list eth0 | awk '{print $4}' | cut -d/ -f1)"
echo "ip host: $(curl -s "http://169.254.169.254/latest/meta-data/local-ipv4")"

# get shared dir
IFS=',' _shared_dirs=(${PCLUSTER_SHARED_DIRS})
_shared_dir=${_shared_dirs[0]}
_job_dir="${_shared_dir}/${AWS_BATCH_JOB_ID%#*}-${AWS_BATCH_JOB_ATTEMPT}"
_exit_code_file="${_job_dir}/batch-exit-code"

if [[ "${AWS_BATCH_JOB_NODE_INDEX}" -eq  "${AWS_BATCH_JOB_MAIN_NODE_INDEX}" ]]; then
    echo "Hello I'm the main node $HOSTNAME! I run the mpi job!"

    mkdir -p "${_job_dir}"

    echo "Compiling..."
    /usr/lib64/openmpi/bin/mpicc -o "${_job_dir}/mpi_hello_world" "${_shared_dir}/
mpi_hello_world.c"

    echo "Running..."
    /usr/lib64/openmpi/bin/mpirun --mca btl_tcp_if_include eth0 --allow-run-as-root --
machinefile "${HOME}/hostfile" "${_job_dir}/mpi_hello_world"

    # Write exit status code
    echo "0" > "${_exit_code_file}"
    # Waiting for compute nodes to terminate
    sleep 30
else
```

```
    echo "Hello I'm the compute node $HOSTNAME! I let the main node orchestrate the mpi
 processing!"
    # Since mpi orchestration happens on the main node, we need to make sure the
 containers representing the compute
    # nodes are not terminated. A simple trick is to wait for a file containing the
 status code to be created.
    # All compute nodes are terminated by AWS Batch if the main node exits abruptly.
    while [ ! -f "${_exit_code_file}" ]; do
        sleep 2
    done
    exit $(cat "${_exit_code_file}")
fi
```

Wir sind nun bereit zum Übermitteln unserer ersten MPI-Aufgabe, die auf drei Knoten gleichzeitig ausgeführt wird:

```
$ awsbsub -n 3 -cf submit_mpi.sh
```

Lassen Sie uns nun den Aufgabenstatus überwachen. Warten Sie, bis die Aufgabe in den Status `RUNNING` wechselt:

```
$ watch awsbstat -d
```

Wenn die Aufgabe in den Status `RUNNING` wechselt, können wir uns die entsprechende Ausgabe ansehen. Fügen Sie `#0` der Aufgaben-ID an, um die Ausgabe des Hauptknotens anzuzeigen. Verwenden Sie `#1` und `#2`, um die Ausgabe der Datenverarbeitungsknoten anzuzeigen:

```
[ec2-user@ip-10-0-0-111 ~]$ awsbout -s 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#0
2018-11-27 15:50:10: Job id: 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#0
2018-11-27 15:50:10: Initializing the environment...
2018-11-27 15:50:10: Starting ssh agents...
2018-11-27 15:50:11: Agent pid 7
2018-11-27 15:50:11: Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
2018-11-27 15:50:11: Mounting shared file system...
2018-11-27 15:50:11: Generating hostfile...
2018-11-27 15:50:11: Detected 1/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:50:26: Detected 1/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:50:41: Detected 1/3 compute nodes. Waiting for all compute nodes to
 start.
```

```
2018-11-27 15:50:56: Detected 3/3 compute nodes. Waiting for all compute nodes to
 start.
2018-11-27 15:51:11: Starting the job...
download: s3://amzn-s3-demo-bucket/batch/job-submit_mpi_sh-1543333713772.sh to tmp/
batch/job-submit_mpi_sh-1543333713772.sh
2018-11-27 15:51:12: ip container: 10.0.0.180
2018-11-27 15:51:12: ip host: 10.0.0.245
2018-11-27 15:51:12: Compiling...
2018-11-27 15:51:12: Running...
2018-11-27 15:51:12: Hello I'm the main node! I run the mpi job!
2018-11-27 15:51:12: Warning: Permanently added '10.0.0.199' (RSA) to the list of known
 hosts.
2018-11-27 15:51:12: Warning: Permanently added '10.0.0.147' (RSA) to the list of known
 hosts.
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-180.ec2.internal, rank 1 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-199.ec2.internal, rank 5 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-180.ec2.internal, rank 0 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-199.ec2.internal, rank 4 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-147.ec2.internal, rank 2 out
 of 6 processors
2018-11-27 15:51:13: Hello world from processor ip-10-0-0-147.ec2.internal, rank 3 out
 of 6 processors

[ec2-user@ip-10-0-0-111 ~]$ awsbout -s 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#1
2018-11-27 15:50:52: Job id: 5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d#1
2018-11-27 15:50:52: Initializing the environment...
2018-11-27 15:50:52: Starting ssh agents...
2018-11-27 15:50:52: Agent pid 7
2018-11-27 15:50:52: Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
2018-11-27 15:50:52: Mounting shared file system...
2018-11-27 15:50:52: Generating hostfile...
2018-11-27 15:50:52: Starting the job...
download: s3://amzn-s3-demo-bucket/batch/job-submit_mpi_sh-1543333713772.sh to tmp/
batch/job-submit_mpi_sh-1543333713772.sh
2018-11-27 15:50:53: ip container: 10.0.0.199
2018-11-27 15:50:53: ip host: 10.0.0.227
2018-11-27 15:50:53: Compiling...
2018-11-27 15:50:53: Running...
```

```
2018-11-27 15:50:53: Hello I'm a compute node! I let the main node orchestrate the mpi
 execution!
```

Wir können jetzt bestätigen, dass die Aufgabe erfolgreich abgeschlossen wurde:

```
[ec2-user@ip-10-0-0-111 ~]$ awsbstat -s ALL
jobId                                   jobName        status     startedAt
 stoppedAt            exitCode
------------------------------------  -------------  ---------  -------------------
 -------------------  ----------
5b4d50f8-1060-4ebf-ba2d-1ae868bbd92d  submit_mpi_sh  SUCCEEDED  2018-11-27 15:50:10
 2018-11-27 15:51:26  -
```

Falls Sie eine Aufgabe vorzeitig beenden möchten, können Sie den Befehl `awsbkill` verwenden.

Festplattenverschlüsselung mit einem benutzerdefinierten KMS-Schlüssel

AWS ParallelCluster unterstützt die Konfigurationsoptionen `ebs_kms_key_id` und `fsx_kms_key_id`. Mit diesen Optionen können Sie einen benutzerdefinierten AWS KMS Schlüssel für die Amazon EBS-Festplattenverschlüsselung oder FSx für Lustre angeben. Geben Sie eine `ec2_iam_role` an, um sie zu verwenden.

Damit der Cluster erstellt werden kann, muss der AWS KMS Schlüssel den Namen der Clusterrolle kennen. Dadurch wird verhindert, dass Sie die bei der Cluster-Erstellung generierte Rolle verwenden, die eine `ec2_iam_role` erfordert.

Voraussetzungen

- AWS ParallelCluster ist installiert.
- Das AWS CLI ist installiert und konfiguriert.
- Sie haben ein EC2 key pair.
- Sie haben eine IAM-Rolle mit den für die Ausführung der `pcluster` CLI erforderlichen Berechtigungen.

Erstellen der -Rolle

Zuerst erstellen Sie eine Richtlinie:

1. Gehen Sie zur IAM-Konsole: https://console.aws.amazon.com/iam/ Home.
2. Klicken Sie unter Richtlinien, Richtlinie erstellen auf die Registerkarte JSON.
3. Fügen Sie die Instance-Richtlinie als Text der Richtlinie hinzu. Achten Sie darauf, alle Vorkommnisse von `<AWS ACCOUNT ID>` und `<REGION>` zu ersetzen.
4. Geben Sie für die Richtlinie den Namen `ParallelClusterInstancePolicy` und klicken Sie dann auf Richtlinie erstellen.

Erstellen Sie als Nächstes eine Rolle:

1. Erstellen Sie unter Rollen eine Rolle.
2. Klicken Sie auf `EC2` als vertrauenswürdige Entität.
3. Suchen Sie unter Berechtigungen nach der `ParallelClusterInstancePolicy`-Rolle, die Sie gerade erstellt haben, und fügen Sie diese an.
4. Geben Sie für die Rolle den Namen `ParallelClusterInstanceRole` und klicken Sie dann auf Rolle erstellen.

Erteilen Sie Ihre Schlüsselberechtigungen

Klicken Sie in der AWS KMS Konsole > Vom Kunden verwaltete Schlüssel > auf den Alias oder die Schlüssel-ID Ihres Schlüssels.

Klicken Sie im Feld Hauptbenutzer unter dem Tab Schlüsselrichtlinie auf die Schaltfläche Hinzufügen und suchen ParallelClusterInstanceRoleSie nach den soeben erstellten Benutzern. Fügen Sie sie an.

Erstellen des Clusters

Erstellen Sie nun eine Cluster. Im Folgenden finden Sie ein Beispiel für einen Cluster mit verschlüsselten `Raid 0`-Laufwerken:

```
[cluster default]
...
raid_settings = rs
ec2_iam_role = ParallelClusterInstanceRole

[raid rs]
shared_dir = raid
raid_type = 0
```

```
num_of_raid_volumes = 2
volume_size = 100
encrypted = true
ebs_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Im Folgenden finden Sie ein Beispiel FSx für das for Lustre-Dateisystem:

```
[cluster default]
...
fsx_settings = fs
ec2_iam_role = ParallelClusterInstanceRole

[fsx fs]
shared_dir = /fsx
storage_capacity = 3600
imported_file_chunk_size = 1024
export_path = s3://bucket/folder
import_path = s3://bucket
weekly_maintenance_start_time = 1:00:00
fsx_kms_key_id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Ähnliche Konfigurationen gelten für Amazon EBS und FSx Amazon-basierte Dateisysteme.

Tutorial zum Modus mit mehreren Warteschlangen

Ihre Jobs im Modus AWS ParallelCluster mit mehreren Warteschlangen ausführen

In diesem Tutorial erfahren Sie, wie Sie Ihren ersten Hello World-Job AWS ParallelCluster mit ausführenModus mit mehreren Warteschlangen.

Voraussetzungen

- AWS ParallelCluster ist installiert.
- Das AWS CLI ist installiert und konfiguriert.
- Sie haben ein EC2 key pair.
- Sie haben eine IAM-Rolle mit den für die Ausführung der `pcluster` CLI erforderlichen Berechtigungen.

> **Note**
> Der Modus mit mehreren Warteschlangen wird nur für AWS ParallelCluster Version 2.9.0 oder höher unterstützt.

Konfiguration Ihres Clusters

Stellen Sie zunächst sicher, dass die Installation korrekt AWS ParallelCluster ist, indem Sie den folgenden Befehl ausführen.

```
$ pcluster version
```

Mehr über `pcluster version` erfahren Sie unter `pcluster version`.

Dieser Befehl gibt die laufende Version von zurück AWS ParallelCluster.

Führen Sie als Nächstes den Befehl aus, `pcluster configure` um eine grundlegende Konfigurationsdatei zu generieren. Folgen Sie allen Anweisungen, die diesem Befehl folgen.

```
$ pcluster configure
```

Weitere Informationen zum Befehl `pcluster configure` finden Sie unter `pcluster configure`.

Nachdem Sie diesen Schritt abgeschlossen haben, sollten Sie eine grundlegende Konfigurationsdatei unter `~/.parallelcluster/config` haben. Diese Datei sollte eine grundlegende Clusterkonfiguration und einen VPC-Abschnitt enthalten.

Im nächsten Teil des Tutorials wird beschrieben, wie Sie Ihre neu erstellte Konfiguration ändern und einen Cluster mit mehreren Warteschlangen starten.

> **Note**
> Einige Instances, die in diesem Tutorial verwendet werden, kommen nicht für das kostenlose Kontingent in Frage.

Verwenden Sie für dieses Tutorial die folgende Konfiguration.

```
[global]
```

```
update_check = true
sanity_check = true
cluster_template = multi-queue

[aws]
aws_region_name = <Your AWS-Region>

[scaling demo]
scaledown_idletime = 5              # optional, defaults to 10 minutes

[cluster multi-queue-special]
key_name = < Your key name >
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge    # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo             # optional, defaults to no custom scaling settings
queue_settings = efa,gpu

[cluster multi-queue]
key_name = <Your SSH key name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge    # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = spot,ondemand

[queue spot]
compute_resource_settings = spot_i1,spot_i2
compute_type = spot                 # optional, defaults to ondemand

[compute_resource spot_i1]
instance_type = c5.xlarge
min_count = 0                       # optional, defaults to 0
max_count = 10                      # optional, defaults to 10

[compute_resource spot_i2]
instance_type = t2.micro
min_count = 1
initial_count = 2

[queue ondemand]
compute_resource_settings = ondemand_i1
```

```
disable_hyperthreading = true       # optional, defaults to false

[compute_resource ondemand_i1]
instance_type = c5.2xlarge
```

Ihren Cluster erstellen

In diesem Abschnitt wird beschrieben, wie Sie den Cluster im Modus mit mehreren Warteschlangen erstellen.

Benennen Sie zunächst Ihren Cluster `multi-queue-hello-world` und erstellen Sie den Cluster gemäß dem `multi-queue` Cluster-Abschnitt, der im vorherigen Abschnitt definiert wurde.

```
$ pcluster create multi-queue-hello-world -t multi-queue
```

Mehr über `pcluster create` erfahren Sie unter `pcluster create`.

Wenn der Cluster erstellt wird, wird die folgende Ausgabe angezeigt:

```
Beginning cluster creation for cluster: multi-queue-hello-world
Creating stack named: parallelcluster-multi-queue-hello-world
Status: parallelcluster-multi-queue-hello-world - CREATE_COMPLETE
MasterPublicIP: 3.130.xxx.xx
ClusterUser: ec2-user
MasterPrivateIP: 172.31.xx.xx
```

Die Meldung `CREATE_COMPLETE` weist darauf hin, dass der Cluster erfolgreich erstellt wurde. Die Ausgabe enthält auch die öffentlichen und privaten IP-Adressen des Hauptknotens.

Loggen Sie sich in Ihren Hauptknoten ein

Verwenden Sie Ihre private SSH-Schlüsseldatei, um sich bei Ihrem Headnode anzumelden.

```
$ pcluster ssh multi-queue-hello-world -i ~/path/to/keyfile.pem
```

Mehr über `pcluster ssh` erfahren Sie unter `pcluster ssh`.

Führen Sie nach der Anmeldung den `sinfo` Befehl aus, um zu überprüfen, ob Ihre Scheduler-Warteschlangen eingerichtet und konfiguriert sind.

Weitere Informationen zu finden Sie `sinfo` unter sinfo im Slurm Dokumentation.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     18  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      2   idle spot-dy-t2micro-1,spot-st-t2micro-1
```

Die Ausgabe zeigt, dass Sie zwei `t2.micro` Rechenknoten im `idle` Status haben, die in Ihrem Cluster verfügbar sind.

> **Note**
>
> - `spot-st-t2micro-1`ist ein statischer Knoten mit `st` im Namen. Dieser Knoten ist immer verfügbar und entspricht dem `min_count = 1` in Ihrer Clusterkonfiguration.
> - `spot-dy-t2micro-1`ist ein dynamischer Knoten mit `dy` im Namen. Dieser Knoten ist derzeit verfügbar, da er Ihrer Clusterkonfiguration entspricht. `initial_count - min_count = 1` Dieser Knoten wird nach der von Ihnen festgelegten Dauer `scaledown_idletime` von fünf Minuten herunterskaliert.

Andere Knoten befinden sich alle im Energiesparmodus, was durch das ~ Suffix im Knotenstatus angezeigt wird, ohne dass sie von EC2 Instanzen unterstützt werden. Die Standardwarteschlange wird durch ein `*` Suffix hinter ihrem Warteschlangennamen gekennzeichnet, ebenso `spot` wie Ihre Standard-Auftragswarteschlange.

Job wird im Modus mit mehreren Warteschlangen ausgeführt

Versuchen Sie als Nächstes, einen Job für eine Weile im Ruhezustand auszuführen. Der Job gibt später seinen eigenen Hostnamen aus. Stellen Sie sicher, dass dieses Skript vom aktuellen Benutzer ausgeführt werden kann.

```
$ cat hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

Senden Sie den Job mit dem `sbatch` Befehl ab. Fordern Sie mit der `-N 2` Option zwei Knoten für diesen Job an und stellen Sie sicher, dass der Job erfolgreich gesendet wurde. Weitere Informationen über `sbatch` finden Sie unter sbatchin der Slurm-Dokumentation.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 2
```

Sie können Ihre Warteschlange einsehen und den Status des Jobs mit dem `squeue` Befehl überprüfen. Beachten Sie, dass die Standardwarteschlange (`spot`) verwendet wird, da Sie keine bestimmte Warteschlange angegeben haben. Weitere Informationen über `squeue` finden Sie unter squeue in der SlurmDokumentation.

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 2      spot     wrap ec2-user  R       0:10      2 spot-dy-
t2micro-1,spot-st-t2micro-1
```

Die Ausgabe zeigt, dass die Aufgabe zurzeit ausgeführt wird. Warten Sie 30 Sekunden, bis die Aufgabe beendet wird, und führen Sie `squeue` dann erneut aus.

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

Nachdem alle Jobs in der Warteschlange abgeschlossen sind, suchen Sie `slurm-2.out` in Ihrem aktuellen Verzeichnis nach der Ausgabedatei.

```
$ cat slurm-2.out
Hello World from spot-dy-t2micro-1
Hello World from spot-st-t2micro-1
```

Die Ausgabe zeigt auch, dass unser Job auf den `spot-st-t2micro-2` Knoten `spot-st-t2micro-1` und erfolgreich ausgeführt wurde.

Reichen Sie nun denselben Job ein, indem Sie mit den folgenden Befehlen Einschränkungen für bestimmte Instanzen angeben.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 3
```

Sie haben diese Parameter für verwendet`sbatch`.

- `-N 3`— fordert drei Knoten an
- `-p spot`— sendet den Job an die `spot` Warteschlange. Sie können einen Job auch an die `ondemand` Warteschlange senden, indem Sie Folgendes angeben`-p ondemand`.
- `-C "[c5.xlarge*1&t2.micro*2]"`— gibt die spezifischen Knotenbeschränkungen für diesen Job an. Dies erfordert, dass ein (1) `c5.xlarge` Knoten und zwei (2) `t2.micro` Knoten für diesen Job verwendet werden.

Führen Sie den `sinfo` Befehl aus, um die Knoten und Warteschlangen anzuzeigen. (Warteschlangen in AWS ParallelCluster werden Partitionen genannt in Slurm.)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite      1   mix# spot-dy-c5xlarge-1
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      2  alloc spot-dy-t2micro-1,spot-st-t2micro-1
```

Die Knoten werden hochgefahren. Dies wird durch das # Suffix für den Knotenstatus angezeigt. Führen Sie das aus squeue Befehl, um Informationen über die Jobs im Cluster anzuzeigen.

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 3      spot     wrap ec2-user CF       0:04      3 spot-dy-
c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```

Ihr Job befindet sich in der `CF` (CONFIGURING) und wartet darauf, dass Instances hochskaliert werden und dem Cluster beitreten.

Nach etwa drei Minuten sollten die Knoten verfügbar sein und der Job geht in den `R` (RUNNING) Zustand.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      1    mix spot-dy-c5xlarge-1
spot*        up   infinite      2  alloc spot-dy-t2micro-1,spot-st-t2micro-1
```

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 3      spot     wrap ec2-user  R       0:04      3 spot-dy-
c5xlarge-1,spot-dy-t2micro-1,spot-st-t2micro-1
```

Der Job ist abgeschlossen und alle drei Knoten befinden sich im `idle` Status.

```
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      3   idle spot-dy-c5xlarge-1,spot-dy-t2micro-1,spot-st-
t2micro-1
```

Sobald keine Jobs mehr in der Warteschlange vorhanden sind, können Sie `slurm-3.out` in Ihrem lokalen Verzeichnis nach Jobs suchen.

```
$ cat slurm-3.out
Hello World from spot-dy-c5xlarge-1
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
```

Die Ausgabe zeigt auch, dass der Job auf den entsprechenden Knoten erfolgreich ausgeführt wurde.

Sie können den Prozess des Herunterskalierens beobachten. In Ihrer Clusterkonfiguration haben Sie eine benutzerdefinierte Dauer `scaledown_idletime` von 5 Minuten angegeben. Nach fünf Minuten im Ruhezustand werden Ihre dynamischen Knoten `spot-dy-c5xlarge-1` `spot-dy-t2micro-1` automatisch herunterskaliert und wechseln in den `POWER_DOWN` Modus. Beachten Sie, dass der statische Knoten `spot-st-t2micro-1` nicht herunterskaliert wird.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite      2  idle% spot-dy-c5xlarge-1,spot-dy-t2micro-1
spot*        up   infinite     17  idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*        up   infinite      1   idle spot-st-t2micro-1
```

Aus dem obigen Code können Sie das erkennen `spot-dy-c5xlarge-1` und `spot-dy-t2micro-1` befinden sich im `POWER_DOWN` Modus. Dies wird durch das `%` Suffix angezeigt. Die

entsprechenden Instanzen werden sofort beendet, aber die Knoten bleiben im `POWER_DOWN` Status und können 120 Sekunden (zwei Minuten) nicht verwendet werden. Nach Ablauf dieser Zeit kehren die Knoten in den Energiesparmodus zurück und können wieder verwendet werden. Weitere Informationen finden Sie unter SlurmLeitfaden für den Modus mit mehreren Warteschlangen.

Dies sollte der endgültige Status des Clusters sein:

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
ondemand     up   infinite     10  idle~ ondemand-dy-c52xlarge-[1-10]
spot*        up   infinite     19  idle~ spot-dy-c5xlarge-[1-10],spot-dy-t2micro-[1-9]
spot*        up   infinite      1   idle spot-st-t2micro-1
```

Nachdem Sie sich vom Cluster abgemeldet haben, können Sie den Vorgang aufräumen, indem Sie den Befehl ausführen`pcluster delete`. Weitere Informationen zu `pcluster list` und finden Sie `pcluster delete` unter `pcluster list` und`pcluster delete`.

```
$ pcluster list
multi-queue CREATE_COMPLETE 2.11.9
$ pcluster delete multi-queue
Deleting: multi-queue
...
```

Jobs auf einem Cluster mit EFA- und GPU-Instanzen ausführen

In diesem Teil des Tutorials wird beschrieben, wie Sie die Konfiguration ändern und einen Cluster mit mehreren Warteschlangen starten, der Instances mit EFA-Netzwerk- und GPU-Ressourcen enthält. Beachten Sie, dass es sich bei den in diesem Tutorial verwendeten Instances um teurere Instances handelt.

Überprüfe deine Kontolimits, um sicherzustellen, dass du berechtigt bist, diese Instances zu nutzen, bevor du mit den in diesem Tutorial beschriebenen Schritten fortfährst.

Ändern Sie die Konfigurationsdatei wie folgt.

```
[global]
update_check = true
sanity_check = true
cluster_template = multi-queue-special

[aws]
```

```
aws_region_name = <Your AWS-Region>

[scaling demo]
scaledown_idletime = 5

[cluster multi-queue-special]
key_name = <Your SSH key name>
base_os = alinux2                   # optional, defaults to alinux2
scheduler = slurm
master_instance_type = c5.xlarge    # optional, defaults to t2.micro
vpc_settings = <Your VPC section>
scaling_settings = demo
queue_settings = efa,gpu

[queue gpu]
compute_resource_settings = gpu_i1
disable_hyperthreading = true       # optional, defaults to false

[compute_resource gpu_i1]
instance_type = g3.8xlarge

[queue efa]
compute_resource_settings = efa_i1
enable_efa = true
placement_group = DYNAMIC           # optional, defaults to no placement group settings

[compute_resource efa_i1]
instance_type = c5n.18xlarge
max_count = 5
```

Den Cluster erstellen

```
$ pcluster create multi-queue-special -t multi-queue-special
```

Nachdem der Cluster erstellt wurde, verwenden Sie Ihre private SSH-Schlüsseldatei, um sich bei Ihrem Hauptknoten anzumelden.

```
$ pcluster ssh multi-queue-special -i ~/path/to/keyfile.pem
```

Dies sollte der Ausgangszustand des Clusters sein:

```
$ sinfo
```

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      5  idle~ efa-dy-c5n18xlarge-[1-5]
gpu          up   infinite     10  idle~ gpu-dy-g38xlarge-[1-10]
```

In diesem Abschnitt wird beschrieben, wie Sie einige Jobs einreichen, um zu überprüfen, ob Knoten über EFA- oder GPU-Ressourcen verfügen.

Schreiben Sie zunächst die Jobskripte. `efa_job.sh`wird 30 Sekunden lang schlafen. Suchen Sie danach in der Ausgabe des `lspci` Befehls nach EFA. `gpu_job.sh`wird 30 Sekunden lang schlafen. Führen Sie anschließend den Befehl aus, `nvidia-smi` um die GPU-Informationen über den Knoten anzuzeigen.

```
$ cat efa_job.sh
#!/bin/bash

sleep 30
lspci | grep "EFA"

$ cat gpu_job.sh
#!/bin/bash

sleep 30
nvidia-smi

$ chmod +x efa_job.sh
$ chmod +x gpu_job.sh
```

Reichen Sie den Job ein mit `sbatch`

```
$ sbatch -p efa --wrap "srun efa_job.sh"
Submitted batch job 2
$ sbatch -p gpu --wrap "srun gpu_job.sh" -G 1
Submitted batch job 3
$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 2       efa     wrap ec2-user CF       0:32      1 efa-dy-
c5n18xlarge-1
                 3       gpu     wrap ec2-user CF       0:20      1 gpu-dy-g38xlarge-1
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      1   mix# efa-dy-c5n18xlarge-1
efa*         up   infinite      4  idle~ efa-dy-c5n18xlarge-[2-5]
```

```
gpu          up   infinite      1   mix# gpu-dy-g38xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]
```

Nach ein paar Minuten sollten Sie sehen, dass die Knoten online sind und die Jobs ausgeführt werden.

```
[ec2-user@ip-172-31-15-251 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      4  idle~ efa-dy-c5n18xlarge-[2-5]
efa*         up   infinite      1    mix efa-dy-c5n18xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]
gpu          up   infinite      1    mix gpu-dy-g38xlarge-1
[ec2-user@ip-172-31-15-251 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                 4       gpu     wrap ec2-user  R       0:06      1 gpu-dy-g38xlarge-1
                 5       efa     wrap ec2-user  R       0:01      1 efa-dy-
c5n18xlarge-1
```

Überprüfen Sie nach Abschluss des Jobs die Ausgabe. Aus der Ausgabe in der `slurm-2.out` Datei können Sie erkennen, dass EFA auf dem `efa-dy-c5n18xlarge-1` Knoten vorhanden ist. Aus der Ausgabe in der `slurm-3.out` Datei können Sie ersehen, dass die `nvidia-smi` Ausgabe GPU-Informationen für den `gpu-dy-g38xlarge-1` Knoten enthält.

```
$ cat slurm-2.out
00:06.0 Ethernet controller: Amazon.com, Inc. Elastic Fabric Adapter (EFA)

$ cat slurm-3.out
Thu Oct  1 22:19:18 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.51.05    Driver Version: 450.51.05    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla M60           Off  | 00000000:00:1D.0 Off |                    0 |
| N/A   28C    P0    38W / 150W |      0MiB /  7618MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla M60           Off  | 00000000:00:1E.0 Off |                    0 |
| N/A   36C    P0    37W / 150W |      0MiB /  7618MiB |     98%      Default |
|                               |                      |                  N/A |
```

```
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Sie können den Prozess der Verkleinerung beobachten. In der Clusterkonfiguration haben Sie zuvor eine benutzerdefinierte Dauer `scaledown_idletime` von fünf Minuten angegeben. Das hat zur Folge, dass Ihre dynamischen Knoten nach fünf Minuten im Ruhezustand automatisch herunterskaliert werden und in den `POWER_DOWN` Modus wechseln. `spot-dy-c5xlarge-1` `spot-dy-t2micro-1` Schließlich wechseln die Knoten in den Energiesparmodus und können wieder verwendet werden.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      1  idle% efa-dy-c5n18xlarge-1
efa*         up   infinite      4  idle~ efa-dy-c5n18xlarge-[2-5]
gpu          up   infinite      1  idle% gpu-dy-g38xlarge-1
gpu          up   infinite      9  idle~ gpu-dy-g38xlarge-[2-10]

# After 120 seconds
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa*         up   infinite      5  idle~ efa-dy-c5n18xlarge-[1-5]
gpu          up   infinite     10  idle~ gpu-dy-g38xlarge-[1-10]
```

Nachdem Sie sich vom Cluster abgemeldet haben, können Sie den Vorgang aufräumen, indem Sie den Befehl ausführen`pcluster delete <cluster name>`.

```
$ pcluster list
multi-queue-special CREATE_COMPLETE 2.11.9
$ pcluster delete multi-queue-special
Deleting: multi-queue-special
...
```

Weitere Informationen finden Sie unter SlurmLeitfaden für den Modus mit mehreren Warteschlangen.

Entwicklung

Sie können die folgenden Abschnitte verwenden, um mit der Entwicklung von zu beginnen AWS ParallelCluster.

> **Important**
>
> Die folgenden Abschnitte sind für die Verwendung einer benutzerdefinierten Version von Rezeptbuchrezepten und eines benutzerdefinierten AWS ParallelCluster Node Package bestimmt. Diese Informationen behandeln eine fortgeschrittene Methode zur Anpassung mit potenziellen Problemen AWS ParallelCluster, die sich nur schwer debuggen lassen. Das AWS ParallelCluster Team empfiehlt dringend, zur Anpassung die Skripts in den benutzerdefinierten Bootstrap-Aktionen zu verwenden, da Hooks nach der Installation im Allgemeinen einfacher zu debuggen und zwischen Versionen von portabler sind. AWS ParallelCluster

Themen

- Ein benutzerdefiniertes AWS ParallelCluster Kochbuch einrichten
- Ein benutzerdefiniertes AWS ParallelCluster Knotenpaket einrichten

Ein benutzerdefiniertes AWS ParallelCluster Kochbuch einrichten

> **Important**
>
> Im Folgenden finden Sie Anweisungen zur Verwendung einer benutzerdefinierten Version der AWS ParallelCluster Kochbuchrezepte. Dies ist eine fortgeschrittene Methode zur Anpassung mit potenziellen Problemen AWS ParallelCluster, die sich nur schwer debuggen lassen. Das AWS ParallelCluster Team empfiehlt dringend, die Skripts in benutzerdefinierten Bootstrap-Aktionen zur Anpassung zu verwenden, da Hooks nach der Installation im Allgemeinen einfacher zu debuggen und zwischen Versionen von portabler sind. AWS ParallelCluster

Schritte

1. Identifizieren Sie das AWS ParallelCluster Cookbook-Arbeitsverzeichnis, in das Sie den Kochbuchcode geklont haben.AWS ParallelCluster

```
_cookbookDir=<path to cookbook>
```

2. Ermitteln Sie die aktuelle Version des Kochbuchs. AWS ParallelCluster

```
_version=$(grep version ${_cookbookDir}/metadata.rb|awk '{print $2}'| tr -d \')
```

3. Erstellen Sie ein Archiv des AWS ParallelCluster Kochbuches und berechnen Sie dessen MD5.

```
cd "${_cookbookDir}"
_stashName=$(git stash create)
git archive --format tar --prefix="aws-parallelcluster-cookbook-${_version}/"
 "${_stashName:-HEAD}" | gzip > "aws-parallelcluster-cookbook-${_version}.tgz"
md5sum "aws-parallelcluster-cookbook-${_version}.tgz" > "aws-parallelcluster-
cookbook-${_version}.md5"
```

4. Erstellen Sie einen Amazon S3 S3-Bucket und laden Sie das Archiv, seine MD5-Datei und das Datum der letzten Änderung in den Bucket hoch. Erteilen Sie öffentliche Leseberechtigung über eine „public-read“-ACL.

```
_bucket=<the bucket name>
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.tgz s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.tgz
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.md5 s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.md5
aws s3api head-object --bucket ${_bucket} --key cookbooks/aws-parallelcluster-
cookbook-${_version}.tgz --output text --query LastModified > aws-parallelcluster-
cookbook-${_version}.tgz.date
aws s3 cp --acl public-read aws-parallelcluster-cookbook-${_version}.tgz.date s3://
${_bucket}/cookbooks/aws-parallelcluster-cookbook-${_version}.tgz.date
```

5. Fügen Sie der AWS ParallelCluster Konfigurationsdatei unter dem `[cluster]`Abschnitt die folgenden Variablen hinzu.

```
custom_chef_cookbook = https://${_bucket}.s3.<the bucket region>.amazonaws.com/
cookbooks/aws-parallelcluster-cookbook-${_version}.tgz
extra_json = { "cluster" : { "skip_install_recipes" : "no" } }
```

> **Note**
>
> Ab AWS ParallelCluster Version 2.6.1 werden die meisten Installationsrezepte standardmäßig beim Starten von Knoten übersprungen, um die Startzeiten zu verkürzen. Um die meisten Installationsrezepte für bessere Startzeiten auf Kosten der Abwärtskompatibilität zu überspringen, `"skip_install_recipes" : "no"` entfernen Sie den `cluster` Schlüssel in der Einstellung. `extra_json`

Ein benutzerdefiniertes AWS ParallelCluster Knotenpaket einrichten

> **Warning**
>
> Im Folgenden finden Sie Anweisungen zur Verwendung einer benutzerdefinierten Version des AWS ParallelCluster Knotenpakets. Dies ist eine fortgeschrittene Methode zur Anpassung mit potenziellen Problemen AWS ParallelCluster, die sich nur schwer debuggen lassen. Das AWS ParallelCluster Team empfiehlt dringend, die Skripts in benutzerdefinierten Bootstrap-Aktionen zur Anpassung zu verwenden, da Hooks nach der Installation im Allgemeinen einfacher zu debuggen und zwischen Versionen von portabler sind. AWS ParallelCluster

Schritte

1. Identifizieren Sie das Arbeitsverzeichnis des AWS ParallelCluster Knotens, in das Sie den Knotencode geklont haben. AWS ParallelCluster

```
_nodeDir=<path to node package>
```

2. Ermitteln Sie die aktuelle Version des AWS ParallelCluster Knotens.

```
_version=$(grep "version = \"" ${_nodeDir}/setup.py |awk '{print $3}' | tr -d \")
```

3. Erstellen Sie ein Archiv des AWS ParallelCluster Knotens.

```
cd "${_nodeDir}"
```

```
_stashName=$(git stash create)
git archive --format tar --prefix="aws-parallelcluster-node-${_version}/"
 "${_stashName:-HEAD}" | gzip > "aws-parallelcluster-node-${_version}.tgz"
```

4. Erstellen Sie einen Amazon S3 S3-Bucket und laden Sie das Archiv in den Bucket hoch. Erteilen Sie öffentliche Leseberechtigung über eine „public-read“-ACL.

```
_bucket=<the bucket name>
aws s3 cp --acl public-read aws-parallelcluster-node-${_version}.tgz s3://${_bucket}/
node/aws-parallelcluster-node-${_version}.tgz
```

5. Fügen Sie der AWS ParallelCluster Konfigurationsdatei unter dem `[cluster]`Abschnitt die folgende Variable hinzu.

```
extra_json = { "cluster" : { "custom_node_package" : "https://${_bucket}.s3.<the
 bucket region>.amazonaws.com/node/aws-parallelcluster-node-${_version}.tgz",
 "skip_install_recipes" : "no" } }
```

> **Note**
>
> Ab AWS ParallelCluster Version 2.6.1 werden die meisten Installationsrezepte standardmäßig beim Starten von Knoten übersprungen, um die Startzeiten zu verkürzen. Um die meisten Installationsrezepte für bessere Startzeiten auf Kosten der Abwärtskompatibilität zu überspringen, `"skip_install_recipes" : "no"` entfernen Sie den `cluster` Schlüssel in der Einstellung. `extra_json`

AWS ParallelCluster Problembehebung

Die AWS ParallelCluster Community unterhält eine Wiki-Seite, die viele Tipps zur Fehlerbehebung im AWS ParallelCluster GitHub Wiki enthält. Eine Liste der bekannten Probleme finden Sie unter Bekannte Probleme.

Themen

- Protokolle abrufen und aufbewahren
- Behebung von Problemen bei der Stack-Bereitstellung
- Behebung von Problemen in Clustern mit mehreren Warteschlangenmodi
- Behebung von Problemen in Clustern im Single-Queue-Modus
- Probleme bei der Platzierung von Gruppen und beim Starten von Instances
- Verzeichnisse, die nicht ersetzt werden können
- Behebung von Problemen in Amazon DCV
- Behebung von Problemen in Clustern mit AWS Batch Integration
- Fehlerbehebung, wenn eine Ressource nicht erstellt werden kann
- Behebung von Problemen mit der Größe der IAM-Richtlinie
- Zusätzliche Unterstützung

Protokolle abrufen und aufbewahren

Protokolle sind eine nützliche Ressource zur Behebung von Problemen. Bevor Sie Protokolle zur Behebung von Problemen mit Ihren AWS ParallelCluster Ressourcen verwenden können, sollten Sie zunächst ein Archiv mit Clusterprotokollen erstellen. Folgen Sie den Schritten, die im AWS ParallelCluster GitHub Wiki im Thema Erstellen eines Clusterprotokollarchivs beschrieben sind, um diesen Vorgang zu starten.

Wenn bei einem Ihrer laufenden Cluster Probleme auftreten, sollten Sie den Cluster in einen `STOPPED` Zustand versetzen, indem Sie den `pcluster stop <cluster_name>` Befehl ausführen, bevor Sie mit der Fehlerbehebung beginnen. Dadurch werden unerwartete Kosten vermieden.

Wenn der `pcluster` Cluster nicht mehr funktioniert oder wenn Sie einen Cluster löschen und gleichzeitig seine Protokolle beibehalten möchten, führen Sie den `pcluster delete –keep-logs <cluster_name>` Befehl aus. Durch die Ausführung dieses Befehls wird der Cluster gelöscht, die in

Amazon CloudWatch gespeicherte Protokollgruppe wird jedoch beibehalten. Weitere Informationen zu diesem Befehl finden Sie in der `pcluster delete` Dokumentation.

Behebung von Problemen bei der Stack-Bereitstellung

Wenn Ihr Cluster nicht erstellt werden kann und die Stack-Erstellung rückgängig gemacht wird, können Sie die folgenden Protokolldateien durchsuchen, um das Problem zu diagnostizieren. Sie möchten `ROLLBACK_IN_PROGRESS` in diesen Protokollen nach der Ausgabe von suchen. Die Fehlermeldung sollte wie folgt aussehen:

```
$ pcluster create mycluster
Creating stack named: parallelcluster-mycluster
Status: parallelcluster-mycluster - ROLLBACK_IN_PROGRESS
Cluster creation failed.  Failed events:
  - AWS::EC2::Instance MasterServer Received FAILURE signal with UniqueId
 i-07af1cb218dd6a081
```

Um das Problem zu diagnostizieren, erstellen Sie den Cluster erneut`pcluster create`, einschließlich des `--norollback` Kennzeichens. Stellen Sie dann per SSH eine Verbindung zum Cluster her:

```
$ pcluster create mycluster --norollback
...
$ pcluster ssh mycluster
```

Nachdem Sie beim Hauptknoten angemeldet sind, sollten Sie drei primäre Protokolldateien finden, anhand derer Sie den Fehler lokalisieren können.

- `/var/log/cfn-init.log`ist das Protokoll für das `cfn-init` Skript. Überprüfen Sie zuerst dieses Protokoll. Sie werden wahrscheinlich einen Fehler wie `Command chef failed` in diesem Protokoll sehen. Sehen Sie sich die Zeilen unmittelbar vor dieser Zeile an, um weitere Einzelheiten im Zusammenhang mit der Fehlermeldung zu erfahren. Weitere Informationen finden Sie unter cfn-init.
- `/var/log/cloud-init.log`ist das Protokoll für Cloud-Init. Wenn Sie nichts darin sehen, versuchen Sie als `cfn-init.log` Nächstes, in diesem Protokoll nachzuschauen.
- `/var/log/cloud-init-output.log`ist die Ausgabe von Befehlen, die von cloud-init ausgeführt wurden. Dies beinhaltet die Ausgabe von. `cfn-init` In den meisten Fällen müssen Sie sich dieses Protokoll nicht ansehen, um diese Art von Problem zu beheben.

Behebung von Problemen in Clustern mit mehreren Warteschlangenmodi

Dieser Abschnitt ist relevant für Cluster, die mit AWS ParallelCluster Version 2.9.0 und höher mit dem installiert wurden Slurm Job-Scheduler. Weitere Hinweise zum Modus mit mehreren Warteschlangen finden Sie unterModus mit mehreren Warteschlangen.

Themen

- Wichtige Protokolle
- Behebung von Problemen mit der Knoteninitialisierung
- Behebung unerwarteter Knotenersetzungen und -beendigungen
- Probleminstanzen und Knoten ersetzen, beenden oder herunterfahren
- Behebung anderer bekannter Knoten- und Jobprobleme

Wichtige Protokolle

Die folgende Tabelle bietet einen Überblick über die Schlüsselprotokolle für den Hauptknoten:

`/var/log/cfn-init.log`

Dies ist das AWS CloudFormation Init-Protokoll. Es enthält alle Befehle, die beim Einrichten einer Instanz ausgeführt wurden. Es ist nützlich bei der Behebung von Initialisierungsproblemen.

`/var/log/chef-client.log`

Dies ist das Chef-Client-Protokoll. Es enthält alle Befehle, die über Chef/Cinc ausgeführt wurden. Es ist nützlich bei der Behebung von Initialisierungsproblemen.

`/var/log/parallelcluster/slurm_resume.log`

Das ist ein `ResumeProgram` Protokoll. Es startet Instances für dynamische Knoten und ist nützlich für die Behebung von Problemen beim Starten dynamischer Knoten.

`/var/log/parallelcluster/slurm_suspend.log`

Dies ist das `SuspendProgram` Protokoll. Es wird aufgerufen, wenn Instanzen für dynamische Knoten beendet werden, und ist nützlich, um Probleme mit der Terminierung dynamischer Knoten zu beheben. Wenn Sie dieses Protokoll überprüfen, sollten Sie auch das `clustermgtd` Protokoll überprüfen.

`/var/log/parallelcluster/clustermgtd`

Das ist das `clustermgtd` Protokoll. Es läuft als zentraler Daemon, der die meisten Clusteroperationen verwaltet. Er ist nützlich für die Behebung von Problemen beim Starten, Beenden oder Clusterbetrieb.

`/var/log/slurmctld.log`

Das ist der Slurm steuere das Daemon-Protokoll. AWS ParallelCluster trifft keine Skalierungsentscheidungen. Vielmehr versucht es nur, Ressourcen bereitzustellen, um die Anforderungen zu erfüllen Slurm Anforderungen. Dies ist nützlich bei Problemen mit der Skalierung und Zuweisung, bei Problemen mit dem Job und bei Problemen mit der Terminplanung.

Dies sind die wichtigsten Hinweise für die Compute-Knoten:

`/var/log/cloud-init-output.log`

Dies ist das Cloud-Init-Protokoll. Es enthält alle Befehle, die beim Einrichten einer Instanz ausgeführt wurden. Es ist nützlich bei der Behebung von Initialisierungsproblemen.

`/var/log/parallelcluster/computemgtd`

Dies ist das `computemgtd` Protokoll. Es wird auf jedem Rechenknoten ausgeführt, um den Knoten in dem seltenen Fall zu überwachen, dass der `clustermgtd` Daemon auf dem Hauptknoten offline ist. Es ist nützlich bei der Behebung unerwarteter Kündigungsprobleme.

`/var/log/slurmd.log`

Das ist der Slurm Daemon-Protokoll berechnen. Es ist nützlich, um Probleme mit der Initialisierung und Rechenfehlern zu beheben.

Behebung von Problemen mit der Knoteninitialisierung

In diesem Abschnitt wird beschrieben, wie Sie Probleme mit der Knoteninitialisierung beheben können. Dazu gehören Probleme, bei denen der Knoten nicht gestartet, eingeschaltet oder einem Cluster nicht beitreten kann.

Hauptknoten:

Anwendbare Protokolle:

- `/var/log/cfn-init.log`
- `/var/log/chef-client.log`
- `/var/log/parallelcluster/clustermgtd`
- `/var/log/parallelcluster/slurm_resume.log`
- `/var/log/slurmctld.log`

Überprüfen Sie die `/var/log/cfn-init.log` und `/var/log/chef-client.log` -Protokolle. Diese Protokolle sollten alle Aktionen enthalten, die bei der Einrichtung des Hauptknotens ausgeführt wurden. Bei den meisten Fehlern, die während der Installation auftreten, sollte eine Fehlermeldung im `/var/log/chef-client.log` Protokoll stehen. Wenn in der Konfiguration des Clusters Skripts vor oder nach der Installation angegeben sind, überprüfen Sie anhand der Protokollmeldungen, ob das Skript erfolgreich ausgeführt wird.

Wenn ein Cluster erstellt wird, muss der Hauptknoten warten, bis die Rechenknoten dem Cluster beitreten, bevor er dem Cluster beitreten kann. Wenn also die Rechenknoten dem Cluster nicht beitreten können, schlägt auch der Hauptknoten fehl. Je nachdem, welche Art von Compute Notes Sie verwenden, können Sie eines der folgenden Verfahren anwenden, um diese Art von Problem zu beheben:

Dynamische Rechenknoten:

- Suchen Sie in `ResumeProgram` log (`/var/log/parallelcluster/slurm_resume.log`) nach dem Namen Ihres Rechenknotens, um zu sehen, ob der Knoten jemals aufgerufen `ResumeProgram` wurde. (Falls `ResumeProgram` es noch nie aufgerufen wurde, können Sie anhand von `slurmctld` log (`/var/log/slurmctld.log`) überprüfen, ob Slurm hat schon einmal versucht, `ResumeProgram` mit dem Knoten anzurufen.)
- Beachten Sie, dass falsche Berechtigungen für `ResumeProgram` dazu führen `ResumeProgram` können, dass der Fehler automatisch fehlschlägt. Wenn Sie ein benutzerdefiniertes AMI mit Änderungen am `ResumeProgram` Setup verwenden, überprüfen Sie, ob das dem `slurm` Benutzer `ResumeProgram` gehört und über die `744` (`rwxr--r--`) -Berechtigung verfügt.
- Wenn aufgerufen `ResumeProgram` wird, überprüfen Sie, ob eine Instance für den Knoten gestartet wurde. Wenn keine Instance gestartet wurde, sollte Ihnen eine Fehlermeldung angezeigt werden, die den Startfehler beschreibt.
- Wenn die Instance gestartet wird, ist möglicherweise ein Problem während des Einrichtungsvorgangs aufgetreten. Sie sollten die entsprechende private IP-Adresse und Instanz-ID aus dem `ResumeProgram` Protokoll sehen. Darüber hinaus können Sie sich die entsprechenden

Setup-Protokolle für die jeweilige Instanz ansehen. Weitere Informationen zur Behebung eines Setup-Fehlers mit einem Compute-Knoten finden Sie im nächsten Abschnitt.

Statische Rechenknoten:

- Prüfen Sie im Protokoll `clustermgtd` (`/var/log/parallelcluster/clustermgtd`), ob Instanzen für den Knoten gestartet wurden. Wenn sie nicht gestartet wurden, sollte eine klare Fehlermeldung angezeigt werden, in der der Startfehler detailliert beschrieben wird.
- Wenn die Instanz gestartet wird, liegt während des Einrichtungsvorgangs ein Problem vor. Sie sollten die entsprechende private IP-Adresse und Instanz-ID aus dem `ResumeProgram` Protokoll sehen. Darüber hinaus können Sie sich die entsprechenden Setup-Protokolle für die jeweilige Instanz ansehen.

- Knoten berechnen:
  - Anwendbare Protokolle:
    - `/var/log/cloud-init-output.log`
    - `/var/log/slurmd.log`
  - Wenn der Rechenknoten gestartet wird, überprüfen Sie zunächst`/var/log/cloud-init-output.log`, ob dieser die Setup-Protokolle enthalten sollte, die dem `/var/log/chef-client.log` Protokoll auf dem Hauptknoten ähneln. Bei den meisten Fehlern, die während des Setups auftreten, sollten sich Fehlermeldungen im `/var/log/cloud-init-output.log` Protokoll befinden. Wenn in der Clusterkonfiguration Skripts vor oder nach der Installation angegeben sind, überprüfen Sie, ob sie erfolgreich ausgeführt wurden.
  - Wenn Sie ein benutzerdefiniertes AMI mit Änderung von verwenden Slurm Konfiguration, dann könnte es eine geben Slurm verwandter Fehler, der verhindert, dass der Compute-Knoten dem Cluster beitritt. Informationen zu Fehlern im Zusammenhang mit dem Scheduler finden Sie im `/var/log/slurmd.log` Protokoll.

Behebung unerwarteter Knotenersetzungen und -beendigungen

In diesem Abschnitt wird weiter untersucht, wie Sie Probleme im Zusammenhang mit Knoten beheben können, insbesondere wenn ein Knoten ersetzt oder unerwartet beendet wird.

- Anwendbare Protokolle:
  - `/var/log/parallelcluster/clustermgtd`(Kopfknoten)

  - `/var/log/slurmctld.log`(Kopfknoten)
  - `/var/log/parallelcluster/computemgtd`(Rechenknoten)
- Knoten wurden unerwartet ersetzt oder beendet
  - Prüfen Sie im `clustermgtd` Protokoll (`/var/log/parallelcluster/clustermgtd`), ob Sie die Aktion zum Ersetzen oder Beenden eines Knotens ergriffen `clustermgtd` haben. Beachten Sie, dass alle normalen Wartungsaktionen für Knoten `clustermgtd` behandelt werden.
  - Wenn der Knoten `clustermgtd` ersetzt oder beendet wurde, sollte eine Meldung erscheinen, in der detailliert beschrieben wird, warum diese Aktion auf dem Knoten ausgeführt wurde. Wenn der Grund mit dem Scheduler zusammenhängt (z. B. weil der Knoten aktiv ist`DOWN`), schauen Sie im `slurmctld` Protokoll nach, um weitere Informationen zu erhalten. Wenn der Grund auf Amazon EC2 zurückzuführen ist, sollte eine informative Nachricht angezeigt werden, in der das Problem EC2 im Zusammenhang mit Amazon beschrieben wird, für das der Ersatz erforderlich war.
  - Wenn der Knoten `clustermgtd` nicht beendet wurde, überprüfen Sie zunächst, ob es sich um eine erwartete Kündigung durch Amazon handelt EC2, genauer gesagt um eine Spot-Terminierung. `computemgtd`, der auf einem Compute-Knoten ausgeführt wird, kann auch Maßnahmen ergreifen, um einen Knoten zu beenden, wenn `clustermgtd` dieser als fehlerhaft eingestuft wird. Prüfen Sie `computemgtd` log (`/var/log/parallelcluster/computemgtd`), um zu sehen, ob der Knoten `computemgtd` beendet wurde.
- Knoten sind ausgefallen
  - Schauen Sie in `slurmctld` log (`/var/log/slurmctld.log`) nach, warum ein Job oder ein Knoten fehlgeschlagen ist. Beachten Sie, dass Jobs automatisch in die Warteschlange gestellt werden, wenn ein Knoten ausfällt.
  - Wenn `slurm_resume` gemeldet wird, dass der Knoten gestartet wurde, und nach einigen Minuten `clustermgtd` meldet, dass es keine entsprechende Instance in Amazon EC2 für diesen Knoten gibt, schlägt der Knoten möglicherweise während der Einrichtung fehl. Gehen Sie wie folgt vor, um das Protokoll von einem Compute (`/var/log/cloud-init-output.log`) abzurufen:
    - Reichen Sie einen Job zur Vermietung ein Slurm einen neuen Knoten einrichten.
    - Nachdem der Knoten gestartet wurde, aktivieren Sie den Kündigungsschutz mit diesem Befehl.

      ```
      aws ec2 modify-instance-attribute --instance-id i-xyz --disable-api-termination
      ```

    - Rufen Sie mit diesem Befehl die Konsolenausgabe vom Knoten ab.

```
aws ec2 get-console-output --instance-id i-xyz --output text
```

Probleminstanzen und Knoten ersetzen, beenden oder herunterfahren

- Anwendbare Protokolle:
  - `/var/log/parallelcluster/clustermgtd`(Kopfknoten)
  - `/var/log/parallelcluster/slurm_suspend.log`(Kopfknoten)
- Bearbeitet in den meisten Fällen `clustermgtd` alle erwarteten Aktionen zur Instanzbeendigung. Sehen Sie im `clustermgtd` Protokoll nach, warum ein Knoten nicht ersetzt oder beendet werden konnte.
- Wenn dynamische Knoten ausfallen`scaledown_idletime`, schauen Sie im `SuspendProgram` Protokoll nach, ob der spezifische Knoten als Argument aufgerufen `SuspendProgram` wurde. `slurmctld` Beachten Sie, dass tatsächlich `SuspendProgram` keine Aktion ausgeführt wird. Vielmehr protokolliert es nur, wenn es aufgerufen wird. Das Beenden und `NodeAddr` Zurücksetzen aller Instanzen erfolgt von`clustermgtd`. Slurm versetzt Knoten danach `SuspendTimeout` automatisch wieder in einen `POWER_SAVING` Zustand.

Behebung anderer bekannter Knoten- und Jobprobleme

Ein anderes bekanntes Problem besteht darin, dass AWS ParallelCluster möglicherweise keine Jobs zugewiesen oder Skalierungsentscheidungen getroffen werden können. Bei dieser Art von Problem werden Ressourcen AWS ParallelCluster nur entsprechend gestartet, beendet oder verwaltet Slurm Anweisungen. Überprüfen Sie bei diesen Problemen das `slurmctld` Protokoll, um diese Probleme zu beheben.

Behebung von Problemen in Clustern im Single-Queue-Modus

> **Note**
>
> Ab Version 2.11.5 wird die AWS ParallelCluster Verwendung von nicht unterstützt SGE or Torque Scheduler.

Dieser Abschnitt bezieht sich auf Cluster, die nicht über einen Modus mit mehreren Warteschlangen mit einer der folgenden beiden Konfigurationen verfügen:

- Mit einer AWS ParallelCluster Version vor 2.9.0 gestartet und SGE, Torque, oder Slurm Jobplaner.
- Mit AWS ParallelCluster Version 2.9.0 oder höher gestartet und SGE or Torque Job-Scheduler.

Themen

- Wichtige Protokolle
- Fehlerbehebung bei fehlgeschlagenen Start- und Verbindungsvorgängen
- Behebung von Skalierungsproblemen
- Behebung anderer Probleme im Zusammenhang mit Clustern

Wichtige Protokolle

Die folgenden Protokolldateien sind die Schlüsselprotokolle für den Hauptknoten.

Für AWS ParallelCluster Version 2.9.0 oder höher:

`/var/log/chef-client.log`

Dies ist das CINC (Chef) -Client-Protokoll. Es enthält alle Befehle, die über CINC ausgeführt wurden. Es ist nützlich bei der Behebung von Initialisierungsproblemen.

Für alle AWS ParallelCluster Versionen:

`/var/log/cfn-init.log`

Das ist das `cfn-init` Protokoll. Es enthält alle Befehle, die beim Einrichten einer Instanz ausgeführt wurden, und ist daher nützlich für die Behebung von Initialisierungsproblemen. Weitere Informationen finden Sie unter cfn-init.

`/var/log/clustermgtd.log`

Dies ist das Protokoll für `clustermgtd` Slurm Scheduler. `clustermgtd`wird als zentraler Daemon ausgeführt, der die meisten Clusteroperationen verwaltet. Er ist nützlich, um Probleme beim Starten, Beenden oder Clusterbetrieb zu beheben.

`/var/log/jobwatcher`

Dies ist das `jobwatcher` Protokoll für SGE and Torque Scheduler. `jobwatcher`überwacht die Scheduler-Warteschlange und aktualisiert die Auto Scaling Group. Dies ist nützlich bei der Behebung von Problemen im Zusammenhang mit der Skalierung von Knoten.

`/var/log/sqswatcher`

Dies ist das `sqswatcher` Protokoll für SGE and Torque Scheduler. `sqswatcher`verarbeitet das Instance-Ready-Ereignis, das von einer Recheninstanz nach erfolgreicher Initialisierung gesendet wird. Außerdem werden Rechenknoten zur Scheduler-Konfiguration hinzugefügt. Dieses Protokoll ist nützlich, um zu beheben, warum ein oder mehrere Knoten einem Cluster nicht beitreten konnten.

Im Folgenden sind die wichtigsten Protokolle für die Rechenknoten aufgeführt.

AWS ParallelCluster Version 2.9.0 oder höher

`/var/log/cloud-init-output.log`

Dies ist das Cloud-Init-Protokoll. Es enthält alle Befehle, die bei der Einrichtung einer Instanz ausgeführt wurden. Es ist nützlich bei der Behebung von Initialisierungsproblemen.

AWS ParallelCluster Versionen vor 2.9.0

`/var/log/cfn-init.log`

Dies ist das CloudFormation Init-Protokoll. Es enthält alle Befehle, die beim Einrichten einer Instanz ausgeführt wurden. Es ist nützlich bei der Behebung von Initialisierungsproblemen

Alle Versionen

`/var/log/nodewatcher`

Das ist das `nodewatcher` Protokoll. `nodewatcher`Daemons, die bei der Verwendung auf jedem Compute-Knoten ausgeführt werden SGE and Torque Scheduler. Sie skalieren einen Knoten herunter, wenn er inaktiv ist. Dieses Protokoll ist nützlich für alle Probleme im Zusammenhang mit der Reduzierung von Ressourcen.

Fehlerbehebung bei fehlgeschlagenen Start- und Verbindungsvorgängen

- Anwendbare Protokolle:
  - `/var/log/cfn-init-cmd.log`(Hauptknoten und Rechenknoten)
  - `/var/log/sqswatcher`(Hauptknoten)
- Wenn Knoten nicht gestartet werden konnten, schauen Sie im `/var/log/cfn-init-cmd.log` Protokoll nach, um die entsprechende Fehlermeldung zu finden. In den meisten Fällen sind Fehler beim Starten von Knoten auf einen Einrichtungsfehler zurückzuführen.
- Falls Compute-Knoten trotz erfolgreicher Einrichtung nicht an der Scheduler-Konfiguration teilnehmen konnten, überprüfen Sie im `/var/log/sqswatcher` Protokoll, ob das Ereignis `sqswatcher` verarbeitet wurde. Diese Probleme sind in den meisten Fällen darauf zurückzuführen, `sqswatcher` dass das Ereignis nicht verarbeitet wurde.

Behebung von Skalierungsproblemen

- Anwendbare Protokolle:
  - `/var/log/jobwatcher`(Kopfknoten)
  - `/var/log/nodewatcher`(Rechenknoten)
- Probleme mit der Skalierung: Überprüfen Sie für den Hauptknoten im `/var/log/jobwatcher` Protokoll, ob der `jobwatcher` Daemon die richtige Anzahl der erforderlichen Knoten berechnet und die Auto Scaling Scaling-Gruppe aktualisiert hat. Beachten Sie, dass die Scheduler-Warteschlange `jobwatcher` überwacht und die Auto Scaling Group aktualisiert wird.
- Probleme beim Herunterskalieren: Überprüfen Sie bei Rechenknoten das `/var/log/nodewatcher` Protokoll auf dem problematischen Knoten, um zu sehen, warum der Knoten herunterskaliert wurde. Beachten Sie, dass `nodewatcher` Daemons einen Rechenknoten herunterskalieren, wenn er inaktiv ist.

Behebung anderer Probleme im Zusammenhang mit Clustern

Ein bekanntes Problem sind zufällige Fehler bei Rechennotizen auf großen Clustern, insbesondere bei Clustern mit 500 oder mehr Rechenknoten. Dieses Problem steht im Zusammenhang mit einer Einschränkung der Skalierungsarchitektur eines Clusters mit einer Warteschlange. Wenn Sie einen großen Cluster verwenden möchten, verwenden Sie AWS ParallelCluster Version v2.9.0 oder höher Slurm, und dieses Problem vermeiden möchten, sollten Sie ein Upgrade durchführen und zu einem

Cluster wechseln, der den Modus mehrerer Warteschlangen unterstützt. Sie können dies tun, indem Sie Folgendes ausführen`pcluster-config convert`.

Bei ultra-large-scale Clustern ist möglicherweise eine zusätzliche Optimierung Ihres Systems erforderlich. Für weitere Informationen wenden Sie sich an Support.

Probleme bei der Platzierung von Gruppen und beim Starten von Instances

Verwenden Sie eine Platzierungsgruppe, um die niedrigste Latenz zwischen den Knoten zu erzielen. Eine Platzierungsgruppe garantiert, dass sich Ihre Instances auf demselben Netzwerk-Backbone befinden. Wenn bei einer Anfrage nicht genügend Instances verfügbar sind, wird ein `InsufficientInstanceCapacity` Fehler zurückgegeben. Um die Wahrscheinlichkeit zu verringern, dass dieser Fehler bei der Verwendung von Cluster-Placement-Gruppen auftritt, setzen Sie den `placement_group` Parameter auf `DYNAMIC` und setzen Sie den `placement` Parameter auf`compute`.

Wenn Sie ein leistungsstarkes gemeinsam genutztes Dateisystem benötigen, sollten Sie es FSx für Lustre verwenden.

Wenn sich der Hauptknoten in der Platzierungsgruppe befinden muss, verwenden Sie denselben Instanztyp und dasselbe Subnetz sowohl für den Kopf als auch für alle Rechenknoten.
Auf diese Weise hat der `compute_instance_type` Parameter denselben Wert wie der `master_instance_type` Parameter, der `placement` Parameter wird auf `cluster` gesetzt und der `compute_subnet_id` Parameter ist nicht angegeben. Bei dieser Konfiguration wird der Wert des `master_subnet_id` Parameters für die Rechenknoten verwendet.

Weitere Informationen finden Sie unter Problembehandlung bei Instance-Starts und Placement-Gruppen, Rollen und Einschränkungen im EC2 Amazon-Benutzerhandbuch

Verzeichnisse, die nicht ersetzt werden können

Die folgenden Verzeichnisse werden von den Knoten gemeinsam genutzt und können nicht ersetzt werden.

`/home`

Dazu gehört der Standard-Home-Ordner des Benutzers (`/home/ec2_user`auf Amazon Linux, `/home/centos` auf CentOS, und `/home/ubuntu` auf Ubuntu).

`/opt/intel`

Dazu gehören Intel MPI, Intel Parallel Studio und zugehörige Dateien.

`/opt/sge`

> **Note**
>
> Ab Version 2.11.5 wird die AWS ParallelCluster Verwendung von nicht unterstützt SGE or Torque Scheduler.

Das beinhaltet Son of Grid Engine und zugehörige Dateien. (Bedingt, nur wenn `scheduler = sge`.)

`/opt/slurm`

Das beinhaltet Slurm Workload Manager und zugehörige Dateien. (Bedingt, nur wenn `scheduler = slurm`.)

`/opt/torque`

> **Note**
>
> Ab Version 2.11.5 wird die AWS ParallelCluster Verwendung von nicht unterstützt SGE or Torque Scheduler.

Das beinhaltet Torque Resource Manager und zugehörige Dateien. (Bedingt, nur wenn `scheduler = torque`.)

Behebung von Problemen in Amazon DCV

Themen

- Protokolle für Amazon DCV
- Speicher vom Typ Amazon DCV-Instance
- Probleme mit Ubuntu Amazon DCV

Protokolle für Amazon DCV

Die Protokolle für Amazon DCV werden in Dateien im `/var/log/dcv/` Verzeichnis geschrieben. Die Überprüfung dieser Protokolle kann bei der Behebung von Problemen hilfreich sein.

Speicher vom Typ Amazon DCV-Instance

Der Instance-Typ sollte mindestens 1,7 Gibibyte (GiB) RAM haben, um Amazon DCV ausführen zu können. Nano and micro Instance-Typen haben nicht genug Speicher, um Amazon DCV auszuführen.

Probleme mit Ubuntu Amazon DCV

Wenn Sie Gnome Terminal über eine DCV-Sitzung auf Ubuntu ausführen, haben Sie möglicherweise nicht automatisch Zugriff auf die Benutzerumgebung, die über die AWS ParallelCluster Login-Shell verfügbar ist. Die Benutzerumgebung bietet Umgebungsmodule wie openmpi oder intelmpi und andere Benutzereinstellungen.

Die Standardeinstellungen von Gnome Terminal verhindern, dass die Shell als Login-Shell gestartet wird. Das bedeutet, dass Shell-Profile nicht automatisch bezogen werden und die AWS ParallelCluster Benutzerumgebung nicht geladen wird.

Gehen Sie wie folgt vor, um das Shell-Profil korrekt zu beziehen und auf die AWS ParallelCluster Benutzerumgebung zuzugreifen:

- Ändern Sie die standardmäßigen Terminaleinstellungen:
  1. Wählen Sie im Gnome-Terminal das Menü Bearbeiten.
  2. Wählen Sie Einstellungen und dann Profile aus.
  3. Wählen Sie „Befehl“ und anschließend „Befehl als Login-Shell ausführen“.
  4. Öffnen Sie ein neues Terminal.
- Verwenden Sie die Befehlszeile, um die verfügbaren Profile abzurufen:

```
$ source /etc/profile && source $HOME/.bashrc
```

Behebung von Problemen in Clustern mit AWS Batch Integration

Dieser Abschnitt ist relevant für Cluster mit AWS Batch Scheduler-Integration.

Probleme mit dem Hauptknoten

Einrichtungsprobleme im Zusammenhang mit dem Hauptknoten können auf die gleiche Weise wie bei Clustern mit einzelnen Warteschlangen behoben werden. Weitere Informationen zu diesen Problemen finden Sie unter Behebung von Problemen in Clustern im Single-Queue-Modus.

AWS Batch Probleme bei der Einreichung parallel Jobs mit mehreren Knoten

Wenn Sie bei der Verwendung AWS Batch als Job-Scheduler Probleme beim Senden von parallel Jobs mit mehreren Knoten haben, sollten Sie ein Upgrade auf AWS ParallelCluster Version 2.5.0 durchführen. Falls das nicht möglich ist, können Sie die Problemumgehung verwenden, die im Thema beschrieben wird: Selbstpatchen eines Clusters, über das parallel Aufträge mit mehreren Knoten gesendet werden. AWS Batch

Probleme mit der Datenverarbeitung

AWS Batch verwaltet die Skalierungs- und Rechenaspekte Ihrer Dienste. Wenn Sie auf Probleme im Zusammenhang mit der Datenverarbeitung stoßen, finden Sie in der Dokumentation AWS Batch zur Fehlerbehebung Hilfe.

Fehlschläge Job

Wenn ein Job fehlschlägt, können Sie den `awsbout` Befehl ausführen, um die Jobausgabe abzurufen. Sie können den `awsbstat -d` Befehl auch ausführen, um einen Link zu den von Amazon gespeicherten Jobprotokollen zu erhalten CloudWatch.

Fehlerbehebung, wenn eine Ressource nicht erstellt werden kann

Dieser Abschnitt ist relevant für Clusterressourcen, wenn sie nicht erstellt werden können.

Wenn eine Ressource nicht erstellt werden kann, wird eine Fehlermeldung wie die folgende ParallelCluster zurückgegeben.

```
pcluster create -c config my-cluster
Beginning cluster creation for cluster: my-cluster
WARNING: The instance type 'p4d.24xlarge' cannot take public IPs. Please make sure that
 the subnet with
id 'subnet-1234567890abcdef0' has the proper routing configuration to allow private IPs
 reaching the
```

```
Internet (e.g. a NAT Gateway and a valid route table).
WARNING: The instance type 'p4d.24xlarge' cannot take public IPs. Please make sure that
 the subnet with
id 'subnet-1234567890abcdef0' has the proper routing configuration to allow private IPs
 reaching the Internet
(e.g. a NAT Gateway and a valid route table).
Info: There is a newer version 3.0.3 of AWS ParallelCluster available.
Creating stack named: parallelcluster-my-cluster
Status: parallelcluster-my-cluster - ROLLBACK_IN_PROGRESS
Cluster creation failed.  Failed events:
- AWS::CloudFormation::Stack MasterServerSubstack Embedded stack
arn:aws:cloudformation:region-id:123456789012:stack/parallelcluster-my-cluster-
MasterServerSubstack-ABCDEFGHIJKL/a1234567-b321-c765-d432-dcba98766789
was not successfully created:
The following resource(s) failed to create: [MasterServer].
- AWS::CloudFormation::Stack parallelcluster-my-cluster-MasterServerSubstack-
ABCDEFGHIJKL The following resource(s) failed to create: [MasterServer].
- AWS::EC2::Instance MasterServer You have requested more vCPU capacity than your
 current vCPU limit of 0 allows for the instance bucket that the
specified instance type belongs to. Please visit http://aws.amazon.com/contact-us/ec2-
request to request an adjustment to this limit.
(Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request ID:
 a9876543-b321-c765-d432-dcba98766789; Proxy: null)
}
```

Wenn Sie beispielsweise die in der vorherigen Befehlsantwort angezeigte Statusmeldung sehen, müssen Sie Instanztypen verwenden, die Ihr aktuelles vCPU-Limit nicht überschreiten oder mehr vCPU-Kapazität anfordern.

Sie können die CloudFormation Konsole auch verwenden, um Informationen zum Status abzurufen. `"Cluster creation failed"`

CloudFormation Fehlermeldungen von der Konsole aus anzeigen.

1. Melden Sie sich bei der an AWS Management Console und navigieren Sie zu https://console.aws.amazon.com/cloudformation.
2. Wählen Sie den Stack mit dem Namen parallelcluster- aus. *`cluster_name`*
3. Wählen Sie die Registerkarte Ereignisse.
4. Überprüfen Sie den Status der Ressource, die nicht erstellt werden konnte, indem Sie die Liste der Ressourcenereignisse nach der logischen ID durchsuchen. Wenn eine Unteraufgabe nicht

erstellt werden konnte, gehen Sie rückwärts vor, um das fehlgeschlagene Ressourcenereignis zu finden.

5. Ein Beispiel für eine AWS CloudFormation Fehlermeldung:

```
2022-02-07 11:59:14 UTC-0800 MasterServerSubstack CREATE_FAILED Embedded stack
arn:aws:cloudformation:region-id:123456789012:stack/parallelcluster-my-cluster-
MasterServerSubstack-ABCDEFGHIJKL/a1234567-b321-c765-d432-dcba98766789
was not successfully created: The following resource(s) failed to create:
 [MasterServer].
```

Behebung von Problemen mit der Größe der IAM-Richtlinie

Informationen zur Überprüfung der AWS STS Kontingente für verwaltete Richtlinien, die Rollen zugeordnet sind, finden Sie unter IAM und Kontingente, Namensanforderungen und Zeichenbeschränkungen. Wenn die Größe einer verwalteten Richtlinie das Kontingent überschreitet, teilen Sie die Richtlinie in zwei oder mehr Richtlinien auf. Wenn Sie das Kontingent für die Anzahl der Richtlinien, die einer IAM-Rolle zugeordnet sind, überschreiten, erstellen Sie zusätzliche Rollen und verteilen Sie die Richtlinien auf diese, um das Kontingent zu erreichen.

Zusätzliche Unterstützung

Eine Liste der bekannten Probleme finden Sie auf der GitHubWiki-Hauptseite oder auf der Problemseite. Bei dringenderen Problemen wenden Sie sich an Support oder öffnen Sie ein neues GitHub Problem.

AWS ParallelCluster Unterstützungspolitik

AWS ParallelCluster unterstützt mehrere Versionen gleichzeitig. Jede AWS ParallelCluster Version hat ein geplantes Ende der Support-Laufzeit (EOSL). Nach dem EOSL-Datum wird für diese Version kein weiterer Support oder keine Wartung mehr angeboten.

AWS ParallelCluster verwendet ein `major.minor.`*`patch`* Versionsschema. Neue Funktionen, Leistungsverbesserungen, Sicherheitsupdates und Bugfixes sind in den neuen Nebenversionsversionen der neuesten Hauptversion enthalten. Nebenversionen sind innerhalb einer Hauptversion abwärtskompatibel. AWS Stellt für kritische Probleme Korrekturen durch Patch-Versionen bereit, jedoch nur für die neuesten Nebenversionen von Releases, die EOSL noch nicht erreicht haben. Wenn Sie die Updates einer neuen Version verwenden möchten, müssen Sie ein Upgrade auf die neue Neben- oder Patch-Version durchführen.

AWS ParallelCluster Versionen	Datum des Endes der unterstützten Nutzungsdauer (EOSL)
2.10.4 und früher	31.12.2021
2.11. *x*	31.12.2022

Sicherheit in AWS ParallelCluster

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud.

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS ParallelCluster, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem jeweiligen AWS Dienst oder den Diensten, die Sie nutzen. Sie sind auch für mehrere andere damit verbundene Faktoren verantwortlich, darunter die Sensibilität Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

In dieser Dokumentation wird beschrieben, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung anwenden sollten AWS ParallelCluster. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen AWS ParallelCluster , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Außerdem erfahren Sie, wie Sie Ihre Ressourcen auf eine AWS ParallelCluster Weise verwenden können, die Ihnen hilft, Ihre AWS Ressourcen zu überwachen und zu schützen.

Themen

- Sicherheitsinformationen für Dienste, die genutzt werden von AWS ParallelCluster
- Datenschutz in AWS ParallelCluster
- Identity and Access Management für AWS ParallelCluster
- Konformitätsvalidierung für AWS ParallelCluster
- Erzwingen einer Mindestversion von TLS 1.2

Sicherheitsinformationen für Dienste, die genutzt werden von AWS ParallelCluster

- Sicherheit bei Amazon EC2
- Sicherheit in Amazon API Gateway
- Sicherheit in AWS Batch
- Sicherheit in AWS CloudFormation
- Sicherheit bei Amazon CloudWatch
- Sicherheit in AWS CodeBuild
- Sicherheit in Amazon DynamoDB
- Sicherheit in Amazon ECR
- Sicherheit in Amazon ECS
- Sicherheit in Amazon EFS
- Sicherheit im Visier FSx von Lustre
- Sicherheit in AWS Identity and Access Management (IAM)
- Sicherheit in EC2 Image Builder
- Sicherheit in AWS Lambda
- Sicherheit in Amazon Route 53
- Sicherheit in Amazon SNS
- Sicherheit in Amazon SQS (für AWS ParallelCluster Version 2.x.)
- Sicherheit in Amazon S3
- Sicherheit in Amazon VPC

Datenschutz in AWS ParallelCluster

Das Modell der AWS gemeinsamen Verantwortung und geteilter Verantwortung gilt für den Datenschutz in AWS ParallelCluster. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig

gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der AWS ParallelCluster API oder auf andere AWS-Services Weise arbeiten oder diese verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

Ein wesentliches Merkmal eines sicheren Service ist, dass Informationen verschlüsselt werden, wenn sie nicht aktiv verwendet werden.

Verschlüsselung im Ruhezustand

AWS ParallelCluster speichert selbst keine anderen Kundendaten als die Anmeldeinformationen, die es benötigt, um im Namen des Benutzers mit den AWS Diensten zu interagieren.

Daten auf den Knoten im Cluster können im Ruhezustand verschlüsselt werden.

Für Amazon EBS-Volumes wird die Verschlüsselung anhand der `ebs_kms_key_id` Einstellungen im `[ebs]`Abschnitt für AWS ParallelCluster Version 2.x konfiguriert.) Weitere Informationen finden Sie unter Amazon EBS-Verschlüsselung im EC2 Amazon-Benutzerhandbuch.

Für Amazon EFS-Volumes wird die Verschlüsselung mithilfe der `efs_kms_key_id` Einstellungen `encrypted` und im `[efs]`Abschnitt in AWS ParallelCluster Version 2.x konfiguriert. Weitere Informationen finden Sie unter So funktioniert Verschlüsselung im Ruhezustand im Amazon Elastic File System-Benutzerhandbuch.

FSx Für Lustre-Dateisysteme wird die Verschlüsselung von Daten im Ruhezustand automatisch aktiviert, wenn ein FSx Amazon-Dateisystem erstellt wird. Weitere Informationen finden Sie unter Verschlüsseln ruhender Daten im Amazon FSx for Lustre-Benutzerhandbuch.

Bei Instance-Typen mit NVMe Volumes werden die Daten auf NVMe Instance-Speicher-Volumes mit einer XTS-AES-256-Verschlüsselung verschlüsselt, die auf einem Hardwaremodul auf der Instance implementiert ist. Die Verschlüsselungsschlüssel werden mithilfe des Hardwaremoduls generiert und sind für jedes Instance-Speichergerät einzigartig. NVMe Alle Verschlüsselungsschlüssel werden zerstört, wenn die Instance angehalten oder beendet wird, und können nicht wiederhergestellt werden. Sie können diese Verschlüsselung nicht deaktivieren und keine eigenen Verschlüsselungsschlüssel bereitstellen. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand im EC2 Amazon-Benutzerhandbuch.

Wenn Sie AWS ParallelCluster einen AWS Service aufrufen, der Kundendaten zur Speicherung auf Ihren lokalen Computer überträgt, finden Sie im Kapitel Sicherheit und Konformität im Benutzerhandbuch dieses Dienstes weitere Informationen darüber, wie diese Daten gespeichert, geschützt und verschlüsselt werden.

Verschlüsselung während der Übertragung

Standardmäßig werden alle Daten, die von den laufenden Client-Computern AWS ParallelCluster und den AWS Dienstendpunkten übertragen werden, verschlüsselt, indem alles über eine HTTPS/TLS-Verbindung gesendet wird. Der Datenverkehr zwischen den Knoten im Cluster kann je nach

den ausgewählten Instanztypen automatisch verschlüsselt werden. Weitere Informationen finden Sie unter Verschlüsselung bei der Übertragung im EC2 Amazon-Benutzerhandbuch.

Weitere Informationen finden Sie auch unter

- Datenschutz bei Amazon EC2
- Datenschutz in EC2 Image Builder
- Datenschutz in AWS CloudFormation
- Datenschutz in Amazon EFS
- Datenschutz in Amazon S3
- Datenschutz FSx für Lustre

Identity and Access Management für AWS ParallelCluster

AWS ParallelCluster verwendet Rollen für den Zugriff auf Ihre AWS Ressourcen und deren Dienste. Die Instanz- und Benutzerrichtlinien, die zur Gewährung von Berechtigungen AWS ParallelCluster verwendet werden, sind unter dokumentiertAWS Identity and Access Management Rollen in AWS ParallelCluster.

Der einzige wesentliche Unterschied besteht darin, wie Sie sich bei der Verwendung eines Standard--Benutzers und Langzeit-Anmeldeinformationen authentifizieren. Ein Benutzer benötigt zwar ein Passwort, um auf die Konsole eines AWS Dienstes zuzugreifen, aber derselbe Benutzer benötigt ein Zugriffsschlüsselpaar, um dieselben Operationen mit AWS ParallelCluster. Alle anderen Kurzzeit-Anmeldeinformationen werden auf die gleiche Weise verwendet, wie sie mit der Konsole verwendet werden.

Die von verwendeten Anmeldeinformationen AWS ParallelCluster werden in Klartextdateien gespeichert und sind nicht verschlüsselt.

- Die `$HOME/.aws/credentials`-Datei speichert Langzeit-Anmeldeinformationen, die für den Zugriff auf Ihre AWS -Ressourcen erforderlich sind. Diese beinhalten Ihre Zugriffsschlüssel-ID und Ihren geheimen Zugriffsschlüssel.
- Kurzfristige Anmeldeinformationen, z. B. für Rollen, die Sie übernehmen, oder die für AWS IAM Identity Center Dienste bestimmt sind, werden ebenfalls in den `$HOME/.aws/cli/cache` und $HOME/ gespeichert. aws/sso/cacheOrdner jeweils.

Risikominderung

- Wir empfehlen dringend, die Dateisystemberechtigungen für den Ordner `$HOME/.aws` und seine untergeordneten Ordner und Dateien so zu konfigurieren, dass der Zugriff ausschließlich auf autorisierte Benutzer beschränkt wird.
- Verwenden Sie möglichst Rollen mit temporären Anmeldeinformationen, um bei einer Gefährdung der Sicherheit von Anmeldeinformationen die Möglichkeit für Schäden zu reduzieren. Verwenden Sie Langzeit-Anmeldeinformationen nur zum Abfragen und Aktualisieren der Anmeldeinformationen von Kurzzeit-Rollen.

Konformitätsvalidierung für AWS ParallelCluster

Externe Prüfer bewerten die Sicherheit und Konformität von AWS Services im Rahmen mehrerer AWS Compliance-Programme. Die Nutzung des AWS ParallelCluster Zugriffs auf einen Dienst hat keinen Einfluss auf die Konformität dieses Dienstes.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter AWS Services im Umfang nach Compliance-Programm AWS . Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern über den herunterladen AWS Artifact. Weitere Informationen finden Sie unter Herunterladen von Berichten in AWS Artifact.

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS ParallelCluster hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance Schnellstartanleitungen zu — In diesen Bereitstellungshandbüchern werden architektonische Überlegungen erörtert und Schritte für die Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben. AWS
- Whitepaper Architecting for HIPAA Security and Compliance on Amazon Web Services — Dieses AWS Whitepaper beschreibt, wie Unternehmen HIPAA-konforme Anwendungen erstellen können AWS .
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.

- Bewertung von Ressourcen anhand von Regeln im AWS Config Developer Guide — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Erzwingen einer Mindestversion von TLS 1.2

Um die Sicherheit bei der Kommunikation mit AWS Diensten zu erhöhen, sollten Sie Ihr System so konfigurieren, AWS ParallelCluster dass es TLS 1.2 oder höher verwendet. Wenn Sie verwenden AWS ParallelCluster, wird Python verwendet, um die TLS-Version festzulegen.

Um sicherzustellen, dass keine TLS-Version vor TLS 1.2 AWS ParallelCluster verwendet wird, müssen Sie möglicherweise OpenSSL neu kompilieren, um dieses Minimum durchzusetzen, und dann Python neu kompilieren, um das neu erstellte OpenSSL zu verwenden.

Ermitteln Ihrer derzeit unterstützten Protokolle

Erstellen Sie zunächst mit OpenSSL ein selbstsigniertes Zertifikat, das für den Testserver und das Python-SDK verwendet werden soll.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out
 cert.pem -days 365
```

Starten Sie dann einen Testserver mit OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

Erstellen Sie in einem neuen Terminalfenster eine virtuelle Umgebung und installieren Sie das Python-SDK.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Erstellen Sie ein neues Python-Skript namens `check.py`, das die dem SDK zugrunde liegende HTTP-Bibliothek verwendet.

```
$ 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'))
```

Führen Sie Ihr neues Skript aus.

```
$ python check.py
```

Damit werden Details über die hergestellte Verbindung angezeigt. Suchen Sie in der Ausgabe nach „Protokoll:". Wenn die Ausgabe "TLSv1.2" oder höher ist, verwendet das SDK standardmäßig TLS v1.2 oder höher. Wenn es sich um eine frühere Version handelt, müssen Sie OpenSSL und Python neu kompilieren.

Auch wenn die Installation von Python auf TLS v1.2 oder höher voreingestellt ist, ist es jedoch weiterhin möglich, dass Python neu auf eine frühere Version als TLS v1.2 verhandelt, wenn der Server TLS v1.2 oder höher nicht unterstützt. Um zu überprüfen, ob Python nicht automatisch auf frühere Versionen neu verhandelt, starten Sie den Testserver erneut mit Folgendem.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Wenn Sie eine frühere Version von OpenSSL verwenden, steht Ihnen die `-no_tls_3`-Flag möglicherweise nicht zur Verfügung. Wenn dies der Fall ist, entfernen Sie das Flag, da die von Ihnen verwendete Version von OpenSSL TLS v1.3 nicht unterstützt. Führen Sie dann das Python-Skript aus.

```
$ python check.py
```

Wenn die Installation von Python korrekterweise für Versionen vor TLS 1.2 nicht neu verhandelt, sollten Sie einen SSL-Fehler erhalten.

```
$ 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)')))
```

Wenn Sie eine Verbindung herstellen können, müssen Sie OpenSSL und Python neu kompilieren, um das Aushandeln von Protokollen vor TLS v1.2 zu deaktivieren.

Kompilieren von OpenSSL und Python

Um sicherzustellen, dass AWS ParallelCluster nicht für etwas vor TLS 1.2 verhandelt wird, müssen Sie OpenSSL und Python neu kompilieren. Kopieren Sie dazu den folgenden Inhalt, um ein Skript zu erstellen und auszuführen.

```
#!/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
```

Dadurch wird eine Version von Python kompiliert, die über ein statisch verknüpftes OpenSSL verfügt, das nicht automatisch eine frühere Version als TLS 1.2 aushandelt. Dadurch werden auch OpenSSL im `/opt/openssl-with-min-tls1_2`-Verzeichnis und Python im `/opt/python-with-min-tls1_2`-Verzeichnis installiert. Nachdem Sie dieses Skript ausgeführt haben, bestätigen Sie die Installation der neuen Version von Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Dadurch sollte Folgendes ausgedruckt werden.

```
Python 3.8.1
```

Um zu bestätigen, dass diese neue Version von Python keine frühere Version als TLS 1.2 aushandelt, führen Sie die Schritte unter Ermitteln Ihrer derzeit unterstützten Protokolle mit der neu installierten Python-Version (also `/opt/python-with-min-tls1_2/bin/python3`) erneut aus.

Versionshinweise und Dokumentenverlauf

Die folgende Tabelle beschreibt die wichtigen Updates und neuen Funktionen für das AWS ParallelCluster -Benutzerhandbuch. Wir aktualisieren die Dokumentation regelmäßig, um das Feedback, das Sie uns senden, einzuarbeiten.

Änderung	Beschreibung	Datum
Veröffentlichung nur zur Dokumentation	AWS ParallelCluster Spezifisches Benutzerhandbuch für Version 2 veröffentlicht. Version nur zur Dokumentation: • AWS ParallelCluster Version 2 hat ein eigenes separates Benutzerhandbuch.	17. Juli 2023
AWS ParallelCluster Version 2.11.9 veröffentlicht	AWS ParallelCluster Version 2.11.9 veröffentlicht. Fehlerbehebungen: • Vermeiden Sie die Ersetzung von FSx für Lustre verwalteten Dateisystemen und den Verlust von Daten bei Cluster-Updates, die Änderungen an enthalten. `vpc_security_group_id`	2. Dezember 2022

Einzelheiten zu den Änderungen finden Sie in der `CHANGELOG` Datei für das aws-parallelcluster-Paket unter. GitHub

AWS ParallelCluster Version 2.11.8 veröffentlicht	AWS ParallelCluster Version 2.11.8 veröffentlicht.
	
	Änderungen:
	
	• Aktualisieren Sie die Intel MPI Library auf Version 2021 Update 6 (aktualisiert von Version 2021 Update 4). Weitere Informationen finden Sie unter Intel® MPI Library 2021 Update 6.
	• Aktualisieren Sie den EFA-Installer auf 1.19.0
	  • EFA-Treiber: `efa-1.16.0-1`
	  • EFA-Config: (von) `efa-config-1.11-1` `efa-config-1.9-1`
	  • EFA-Profil: `efa-profile-1.5-1` (keine Änderung)
	  • libFabric-aws: (von) `libfabric-aws-1.16.0-1` `libfabric-1.13.2`
	  • RDMA-Core: (von) `rdma-core-41.0-2` `rdma-core-37.0`
	  • MPI öffnen: `openmpi40-aws-4.1.4-3` (von) `openmpi40-aws-4.1.1-2`
		14. November 2022

	• Aktualisieren Sie die Python-Laufzeit, die von Lambda-Funktionen in der AWS Batch Integration verwendet wird, auf Python3.9. Fehlerbehebungen: • Verhindern Sie, dass Cluster-Tags während eines Updates geändert werden, da dies nicht unterstützt wird. Einzelheiten zu den Änderungen finden Sie in den `CHANGELOG` Dateien für das aws-parallelcluster-Paket unter. GitHub	
AWS ParallelCluster Version 2.11.7 veröffentlicht	AWS ParallelCluster Version 2.11.7 veröffentlicht. Änderungen: • Aktualisieren Sie Slurm auf Version 20.11.9. Einzelheiten zu den Änderungen finden Sie in den `CHANGELOG` Dateien für das aws-parallelcluster-Paket unter. GitHub	13. Mai 2022

AWS ParallelCluster Version 2.11.6 veröffentlicht	AWS ParallelCluster Version 2.11.6 veröffentlicht. Verbesserungen: • Verbessern Sie das Ausnahmemanagement bei fehlendem Netzwerk. Änderungen: • Betriebssystem-Paketaktualisierungen und Sicherheitskorrekturen. Einzelheiten zu den Änderungen finden Sie in den `CHANGELOG` Dateien für das aws-parallelcluster-Paket unter. GitHub	19. April 2022

AWS ParallelCluster Version 2.11.5 veröffentlicht	AWS ParallelCluster Version 2.11.5 veröffentlicht. Verbesserungen: • Unterstützung für `NEW_CHANGED_DELETED` als Wert für die Option FSx `AutoImportPolicy` Lustre hinzugefügt. • Entfernen Sie die Unterstützung für SGE- und Torque-Scheduler. • Deaktivieren Sie den `log4j-cve-2021-44228-hotpatch` Service auf Amazon Linux, um mögliche Leistungseinbußen zu vermeiden. Änderungen: • Aktualisieren Sie den NVIDIA-Treiber auf Version `470.103.01` (von`470.82.01`). • Aktualisieren Sie den NVIDIA Fabric Manager auf Version `470.103.01` (von`470.82.01`). • Aktualisieren Sie die CUDA-Bibliothek auf Version `11.4.4` (von`11.4.3`).	1. März 2022

- Intel MPI wurde auf Version 2021 Update 4 aktualisiert (aktualisiert von Version 2019 Update 8). Weitere Informationen finden Sie unter Intel® MPI Library 2021 Update 4.
- Verlängern Sie das Timeout für die Erstellung von Kopfknoten auf eine Stunde.

Fehlerbehebungen:

- Korrigieren Sie die DCV-Verbindung über Browser.
- Korrigiert die YAML-Anfü hrungszeichen, um zu verhindern, dass benutzerd efinierte Tags als Zahlen analysiert werden.

Einzelheiten zu den Änderungen finden Sie in den `CHANGELOG` Dateien für das aws-parallelcluster-Paket unter. GitHub

AWS ParallelCluster Version 2.11.4 veröffentlicht	AWS ParallelCluster Version 2.11.4 veröffentlicht. Zu den Änderungen gehören: • CentOSUnterstützung für 8 wurde entfernt. CentOS8 erreicht am 31. Dezember 2021 das Lebensende (EOL). • Führen Sie ein Upgrade Slurm Workload Manager auf Version 20.11.8 durch. • Aktualisieren Sie den Cinc Client auf. `17.2.29` • Amazon DCV wurde auf Amazon DCV 2021.2-11 190 aktualisiert. Weitere Informationen finden Sie unter DCV 2021.2-11 190 — 11. Oktober 2021 im Amazon DCV-Administratorhandbuch. • Aktualisieren Sie den NVIDIA-Treiber auf Version (von). `470.82.01` `460.73.01` • Aktualisieren Sie die CUDA-Bibliothek auf Version `11.4.3` (von`11.3.0`). • Aktualisieren Sie NVIDIA Fabric Manager auf`470.82.01` . • Deaktivieren Sie das Paket-Update beim Start der	20. Dezember 2021

Instance auf Amazon Linux 2.

- Deaktivieren Sie das unbeaufsichtigte Paket-Update auf Ubuntu Amazon Linux 2.
- Installieren Sie die Python 3-Version der AWS CloudFormation Hilfsskripte auf CentOS 7 und Ubuntu 18.04. (Diese wurden bereits auf Amazon Linux 2 und Ubuntu 20.04 verwendet.)

Zu den Korrekturen gehören:

- Deaktiviert die Aktualisierung des `ec2_iam_role` Parameters.
- Korrigieren Sie die `CpuOptions` Konfiguration in der Startvorlage für T2 Instances.

Einzelheiten zu den Änderungen finden Sie in den `CHANGELOG` Dateien für den aws-parallelcluster und in den Paketen unter. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub

AWS ParallelCluster Version 2.11.3 veröffentlicht	AWS ParallelCluster Version 2.11.3 veröffentlicht. • Behebung `pcluster createami` eines Fehlers, der darauf zurückzuführen ist, dass Son of Grid Engine Quellen unter nicht verfügbar sind. `arc.liv.ac.uk` Aktualisieren Sie das Elastic Fabric Adapter Installationsprogramm auf 1.14.1 (von 1.13.0) • EFA-Konfiguration: (von) `efa-config-1.9-1` `efa-config-1.9` • EFA-Profil: `efa-profile-1.5-1` (keine Änderung) • EFA-Kernel-Modul: `efa-1.14.2` (von) `efa-1.13.0` • RDMA-Kern: `rdma-core-37.0` (von) `rdma-core-35.0amzn` • Libfabric: `libfabric-1.13.2` (von) `libfabric-1.13.0amzn1.0`	3. November 2021

- MPI öffnen: `openmpi40-aws-4.1.1-2` (keine Änderung)

GPUDirect RDMA ist immer aktiviert, wenn es vom Instanztyp unterstützt wird.

- Die Optionen `enable_efa_gdr` und die `enable_efa_gdr` Konfiguration haben keine Auswirkung.

Einzelheiten zu den Änderungen finden Sie in den `CHANGELOG` Dateien für den aws-parallelcluster und in den Paketen unter. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub

AWS ParallelCluster Version 2.11.2 veröffentlicht	AWS ParallelCluster Version 2.11.2 veröffentlicht.
	
	Zu den Änderungen gehören:
	
	• Installieren Sie EFA nicht mit aktiviertem GPUDirect RDMA (GDR) beim Bootstrap, wenn EFA im Basis-AMI installiert ist.
	• Sperren Sie die Version des `nvidia-fabricmanager` Pakets, damit sie mit der NVIDIA-Treiberversion synchronisiert bleibt, die von installiert wurde. AWS ParallelCluster
	• Slurm: Behebt das Problem, dass der Cluster gestoppt und neu gestartet wurde, während ein Knoten hochgefahren wurde.
	• Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.13.0 aktualisiert:
	  • EFA-Konfiguration: `efa-config-1.9` (keine Änderung)
	  • EFA-Profil: `efa-profile-1.5-1` (keine Änderung)
	  • EFA-Kernel-Modul: `efa-1.13.0` (keine Änderung)
		27. August 2021

  - RDMA-Kern: `rdma-core-35.0amzn` (von) `rdma-core-32.1amzn`
  - Libfabric: `libfabric-1.13.0amzn1.0` (von) `libfabric-1.11.2amzn1.1`
  - MPI öffnen: `openmpi40-aws-4.1.1-2` (keine Änderung)
- Wenn Sie ein benutzerdefiniertes AMI mit einem vorinstallierten EFA-Paket verwenden, werden beim Bootstrap des Knotens keine Änderungen an EFA vorgenommen. Die ursprüngliche EFA-Paket bereitstellung wird beibehalten.

Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für den aws-parallelcluster und die Pakete auf. aws-parallelcluster-cookbook GitHub

AWS ParallelCluster Version 2.11.1 veröffentlicht	AWS ParallelCluster Version 2.11.1 veröffentlicht.
	
	Zu den Änderungen gehören:
	
	• Hängen Sie Dateisyst eme mithilfe der `noatime` Mount-Option ein, um die Aufzeichnung des letzten Zugriffs zu beenden, wenn eine Datei gelesen wird. Dies verbessert die Leistung des Remote-Dateisystems.
	• Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.12.3 aktualisiert:
	  • EFA-Konfiguration: `efa-config-1.9` (von) `efa-config-1.8-1`
	  • EFA-Profil: `efa-profile-1.5-1` (keine Änderung)
	  • EFA-Kernel-Modul: `efa-1.13.0` (von) `efa-1.12.3`
	  • RDMA-Kern: `rdma-core-32.1amzn` (keine Änderung)
	  • Libfabric: `libfabric-1.11.2amzn1.1` (keine Änderung)
	  • MPI öffnen: `openmpi40-aws-4.1.1-2` (keine Änderung)
		23. Juli 2021

- Versuchen Sie erneut, das `aws-parallelcluster` Paket auf dem Hauptknoten zu installieren, wenn Sie es AWS Batch als Scheduler verwenden.
- Vermeiden Sie Fehler beim Aufbau SGE auf einem Instanztyp mit mehr als 31 v. CPUs
- An Version 1.247347.6 des Amazon CloudWatch Agent angeheftet, um Probleme zu vermeiden, die in Version 1.247348.0 aufgetreten sind.

Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für den aws-parallelcluster und die Pakete auf. aws-parallelcluster-cookbook GitHub

AWS ParallelCluster Version 2.11.0 veröffentlicht	AWS ParallelCluster Version 2.11.0 veröffentlicht. Zu den Änderungen gehören: • Unterstützung für Ubuntu 20.04 (`ubuntu2004`) hinzugefügt und Unterstützung für Ubuntu 16.04 (`ubuntu1604`) und Amazon Linux () entfernt. `alinux` Amazon Linux 2 (`alinux2`) wird weiterhin vollständig unterstützt. Weitere Informationen finden Sie unter `base_os`. • Die Unterstützung für Python-Versionen unter 3.6 wurde entfernt. • Die Standardgröße des Root-Volumes wurde auf 35 Gibibyte (GiB) erhöht. Weitere Informationen erhalten Sie unter `compute_root_volume_size` und `master_root_volume_size` . • Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.12.2 aktualisiert: • EFA-Konfiguration: `efa-config-1.8-1` (von) `efa-config-1.7`	1. Juli 2021

  - EFA-Profil: `efa-profile-1.5-1` (von) `efa-profile-1.4`
  - EFA-Kernel-Modul: `efa-1.12.3` (von) `efa-1.10.2`
  - RDMA-Kern: `rdma-core-32.1amzn` (von) `rdma-core-31.2amzn`
  - Libfabric: `libfabric-1.11.2amzn1.1` (von) `libfabric-1.11.1amzn1.0`
  - MPI öffnen: `openmpi40-aws-4.1.1-2` (von) `openmpi40-aws-4.1.0`
- Auf Version `20.11.7` (von`20.02.7`) aktualisiertSlurm.
- Installieren Sie den SSM-Agent auf `centos7` und`centos8`. (Der SSM-Agent ist in `alinux2``ubuntu1804`, und vorinstalliert.) `ubuntu2004`
- SGE: Verwenden Sie immer den Kurznamen als Hostnamenfilter mit. `qstat`
- Verwenden Sie den Instanz-Metadatendienst Version 2 (IMDSv2) anstelle des

Instanz-Metadatendienstes Version 1 (IMDSv1), um Instanz-Metadaten abzurufen. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im EC2 Amazon-Benutzerhandbuch.

- Aktualisieren Sie den NVIDIA-Treiber auf Version `460.73.01` (von`450.80.02`).
- Aktualisieren Sie die CUDA-Bibliothek auf Version `11.3.0` (von`11.0`).
- Aktualisieren Sie NVIDIA Fabric Manager auf`nvidia-fabricmanager-460`.
- Aktualisieren Sie Python, das in AWS ParallelCluster virtualenvs verwendet wird, auf `3.7.10` (von). `3.6.13`
- Aktualisieren Sie den Cinc-Client auf. `16.13.16`
- Aktualisieren Sie die Abhängigkeiten von Drittanbietern von aws-parallelcluster-cookbook:
  - `apt-7.4.0` (von`apt-7.3.0`).
  - `iptables-8.0.0` (von`iptables-7.1.0`).

- `line-4.0.1` (von `line-2.9.0`).
- `openssh-2.9.1` (von `openssh-2.8.1`).
- `pyenv-3.4.2` (von `pyenv-3.1.1`).
- `selinux-3.1.1` (von `selinux-2.1.1`).
- `ulimit-1.1.1` (von `ulimit-1.0.0`).
- `yum-6.1.1` (von `yum-5.1.0`).
- `yum-epel-4.1.2` (von `yum-epel-3.3.0`).

Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für den aws-parallelcluster und in den Paketen unter. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub

AWS ParallelCluster Version 2.10.4 veröffentlicht	AWS ParallelCluster Version 2.10.4 veröffentlicht. Zu den Änderungen gehören: • Auf Version `20.02.7` (von`20.02.4`) aktualisiertSlurm. Weitere Informationen zu den Änderungen finden Sie in der CHANGELOG-Datei für das aws-parallelcluster-Paket unter. GitHub	15. Mai 2021

AWS ParallelCluster Version 2.10.3 veröffentlicht	AWS ParallelCluster Version 2.10.3 veröffentlicht.
	
	Zu den Änderungen gehören:
	
	• Unterstützung für Ubuntu 18.04 und Amazon Linux 2 auf ARM-basierten AWS Graviton-Instances in China und. AWS AWS GovCloud (US) AWS-Regionen
	• Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.11.2 aktualisiert:
	  • EFA-Konfiguration: `efa-config-1.7` (keine Änderung)
	  • EFA-Profil: `efa-profile-1.4` (von) `efa-profile-1.3`
	  • EFA-Kernel-Modul: `efa-1.10.2` (keine Änderung)
	  • RDMA-Kern: `rdma-core-31.2amzn` (keine Änderung)
	  • Libfabric: `libfabric-1.11.1amzn1.0` (keine Änderung)
	  • MPI öffnen: `openmpi40-aws-4.1.0` (keine Änderung)
		18. März 2021

Weitere Informationen zu den Änderungen finden Sie in der CHANGELOG-Datei für das aws-parallelcluster-Paket unter. GitHub

AWS ParallelCluster Version 2.10.2 veröffentlicht	AWS ParallelCluster Version 2.10.2 veröffentlicht. Zu den Änderungen gehören: • Verbessern Sie die Validierung der Cluster-Konfiguration, um das Cluster-Ziel-AMI beim Aufrufen des EC2 RunInstancesAmazon-API-Vorgangs im `--dry-run` Modus zu verwenden. • Aktualisieren Sie die in den AWS ParallelCluster virtuellen Umgebungen verwendete Python-Version auf 3.6.13. • Korrektur `sanity_check` für Arm-Instanztypen. • Behoben `enable_efa` bei Verwendung `centos8` mit den Instanztypen Slurm Scheduler oder Arm. • `apt update`Im nicht interaktiven Modus ausführen ()`-y`. • Fix `encrypted_ephemeral` = wahr mit `alinux2` und`centos8`. Weitere Informationen zu den Änderungen finden Sie in der CHANGELOG-Datei für das aws-parallelcluster-Paket unter. GitHub	2. März 2021

AWS ParallelCluster Version 2.10.1 veröffentlicht	AWS ParallelCluster Version 2.10.1 veröffentlicht.
	
	Zu den Änderungen gehören:
	
	• Unterstützung für Afrika (Kapstadt) (`af-south-1`), Europa (Mailand) (`me-south-1`) und den Nahen Osten (Bahrain) (`me-south-1`) hinzugefügt AWS-Regionen. Bei der Markteinführung ist der Support auf folgende Weise eingeschränkt:
	  • FSx für Lustre- und ARM-basierte Graviton-Instances werden in keiner dieser Versionen unterstützt. AWS-Regionen
	  • AWS Batch wird in Afrika (Kapstadt) nicht unterstützt.
	  • Amazon EBS `io2` und `gp3` Volumetypen werden in Afrika (Kapstadt) und Europa (Mailand) AWS-Regionen nicht unterstützt.
	• Unterstützung für Amazon EBS `io2` und `gp3` Volume-Typen hinzugefügt. Weitere Informationen finden Sie in `[ebs]`Abschnitt und `[raid]`Abschnitt.
		22. Dezember 2020

- Unterstützung für Elastic Fabric Adapter auf ARM basierende Graviton2-Instances hinzugefügt, auf denen, oder ausgeführt wird`alinux2`. `ubuntu1804` `ubuntu2004` Weitere Informationen finden Sie unter Elastic Fabric Adapter.
- Installieren Sie die Arm Performance Libraries 20.2.1 auf Arm AMIs (`alinux2`, und). `centos8` `ubuntu1804` Weitere Informationen finden Sie unter ARM-Leistungsbibliotheken.
- Intel MPI wurde auf Version 2019 Update 8 aktualisiert (aktualisiert von Version 2019 Update 7). Weitere Informationen finden Sie unter Intel® MPI Library 2019 Update 8.
- Der AWS CloudFormation `DescribeStacks` API-Operationsaufruf wurde vom AWS Batch Docker-Einstiegspunkt entfernt, um die Jobfehler zu beenden, die durch die Drosselung von verursacht wurden. AWS CloudFormation
- Die Aufrufe des Amazon EC2 `DescribeInstanceTypes` API-

Operationsaufrufs bei der Validierung einer Cluster-Konfiguration wurden verbessert.

- Amazon Linux 2-Docker-Images werden beim Erstellen des Docker-Images für den Scheduler aus Amazon ECR Public abgerufen. `awsbatch`
- Der Standard-Instance-Typ wurde vom hartcodierten `t2.micro` Instance-Typ in den Instance-Typ „Kostenloses Kontingent“ für AWS-Region (`t2.micro`oder`t3.micro`, abhängig von) geändert. AWS-Region AWS-Regionen für die kein kostenloses Kontingent gilt, ist standardmäßig der `t3.micro` Instance-Typ.
- Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.11.1 aktualisiert:
  - EFA-Konfiguration: `efa-config-1.7` (von) `efa-config-1.5`
  - EFA-Profil: `efa-profile-1.3` (von) `efa-profile-1.1`

  - EFA-Kernel-Modul: `efa-1.10.2` (keine Änderung)
  - RDMA-Kern: `rdma-core-31.2amzn` (von) `rdma-core-31.amzn0`
  - Libfabric: `libfabric-1.11.1amzn1.0` (von) `libfabric-1.10.1amzn1.1`
  - MPI öffnen: `openmpi40-aws-4.1.0` (von) `openmpi40-aws-4.0.5`
- Die `master_subnet_id` Parameter `vpc_settings vpc_id`, und sind jetzt erforderlich.
- Der `nfsd` Daemon im Hauptknoten ist jetzt so eingestellt, dass er mindestens 8 Threads verwendet. Wenn es mehr als 8 Kerne gibt, verwendet er so viele Threads, wie es Kerne gibt. Wenn verwendet `ubuntu1604` wird, ändert sich die Einstellung erst, nachdem der Knoten neu gestartet wurde.
- Amazon DCV wurde auf Amazon DCV 2020.2-9662 aktualisiert. Weitere Informationen finden Sie

unter DCV 2020.2-9662 — 04. Dezember 2020 im Amazon DCV-Administratorhandbuch.

- Die Intel MPI- und HPC-Pakete für AWS ParallelCluster stammen aus Amazon S3. Sie werden nicht mehr aus Intel Yum-Repos abgerufen.
- OSs Bei der Erstellung von official wurde der systemd Standard-Runlevel `multi-user.target` auf „Alle aktiviert“ geändert. AWS ParallelCluster AMIs Der Runlevel ist auf dem Hauptknoten nur dann `graphical.target` auf eingestellt, wenn DCV aktiviert ist. Dadurch wird verhindert, dass grafische Dienste (z. B.`x/gdm`) ausgeführt werden, wenn sie nicht benötigt werden.
- Die Unterstützung für `p4d.24xlarge` Instanzen auf dem Hauptknoten wurde aktiviert.
- Erhöhen Sie die maximale Anzahl von Wiederholungsversuchen bei der Registrierung von Slurm Knoten in Amazon Route 53.

Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für aws-parallelcluster, und packages on. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub

AWS ParallelCluster Version 2.10.0 veröffentlicht	AWS ParallelCluster Version 2.10.0 veröffentlicht.
	
	Zu den Änderungen gehören:
	
	• Unterstützung für CentOS 8 in allen Regionen AWS-Regionen (außerhalb der Regionen AWS China und AWS GovCloud (USA)) hinzugefügt. Die Unterstützung für CentOS 6 wurde entfernt.
	• Unterstützung für `p4d.24xlarge` Instanzen für Rechenknoten hinzugefügt.
	• Mithilfe der neuen `enable_efa_gdr` Einstellung wurde Unterstützung für NVIDIA GPUDirect RDMA auf EFA hinzugefügt.
	• Unterstützung für Amazon FSx for Lustre-Funktionen hinzugefügt.
	  • Konfigurieren Sie Ihr Amazon FSx for Lustre-Dateisystem so, dass Einstellungen mithilfe der `auto_import_policy` Einstellung importiert werden.
	  • Unterstützung für HDD-basierte Amazon FSx for Lustre-Dateisystem e mit den `storage_t`
		18. November 2020

ype Einstellungen und hinzugefügt. drive_cache_type

- Ein CloudWatch Amazon-Dashboard mit Headnode-Metriken und einfachem Zugriff auf Cluster-Logs wurde hinzugefügt. Weitere Informationen finden Sie unter CloudWatch Amazon-Dashboard.
- Unterstützung für die Verwendung eines vorhandenen Amazon S3 S3-Buckets zum Speichern von Cluster-Konfigurationsinformationen mithilfe der cluster_resource_bucket Einstellung hinzugefügt.
- Der pcluster createami Befehl wurde verbessert.
  - Es wurde ein `--post-install` Parameter hinzugefügt, um beim Erstellen eines AMI ein Skript nach der Installation zu verwenden.
  - Es wurde ein Validierungsschritt hinzugefügt, der fehlschlägt, wenn ein Basis-AMI verwendet wird, das mit einer anderen Version

von erstellt wurde AWS ParallelCluster.

- Es wurde ein Validierungsschritt hinzugefügt, der fehlschlägt, wenn sich das ausgewählte Betriebssystem von dem Betriebssystem im Basis-AMI unterscheidet.
- Unterstützung für die Verwendung eines AWS ParallelCluster Basis-AMI hinzugefügt.

- Der `pcluster update` Befehl wurde verbessert.
  - Die `tags` Einstellung kann jetzt während eines Updates geändert werden.
  - Die Größe von Warteschlangen kann jetzt während eines Updates geändert werden, ohne dass die Rechenflotte angehalten wird
- `all_or_nothing_batch` Konfigurationsparameter für `slurm_resume` das Skript hinzugefügt. `slurm_resume` Wann`True`, ist nur erfolgreich, wenn alle Instanzen verfügbar sind, die für alle ausstehenden Jobs in Slurm erforderlich

sind. Weitere Informationen finden Sie unter Introducing `all_or_nothing_batch` Launches im AWS ParallelCluster Wiki am GitHub.

- Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.10.1 aktualisiert:
  - EFA-Konfiguration: `efa-config-1.5` (von) `efa-config-1.4`
  - EFA-Profil: `efa-profile-1.1` (von) `efa-profile-1.0.0`
  - EFA-Kernel-Modul: `efa-1.10.2` (von) `efa-1.6.0`
  - RDMA-Kern: `rdma-core-31.amzn0` (von) `rdma-core-28.amzn0`
  - Libfabric: `libfabric-1.11.1amzn1.0` (von) `libfabric-1.10.1amzn1.1`
  - MPI öffnen: `openmpi40-aws-4.0.5` (von) `openmpi40-aws-4.0.3`
- Aktivieren Sie in den AWS GovCloud (US) Regionen die Unterstützung für

Amazon DCV und AWS Batch.

- Aktivieren Sie in den Regionen AWS China die Unterstützung für Amazon FSx for Lustre.
- Aktualisieren Sie den NVIDIA-Treiber auf Version 450.80.02 (von 450.51.05).
- Installieren Sie NVIDIA Fabric Manager, um NVIDIA auf unterstützten Plattform en zu aktivieren. NVSwitch
- Die Standardeinstellung AWS-Region von wurde entferntus-east-1 . Die Standardeinstellung verwendet diese Suchreihe nfolge.
  - AWS-Region in unserem -r --region Argument angegeben.
  - AWS_DEFAULT_REGION -Umgebungsvariable.
  - aws_regio n_name Einstellung im [aws]Abschnitt der AWS ParallelCluster Konfigura tionsdatei (standard mäßig~/.parall elcluster/config).
  - regionEinstellung im [default] Abschnitt der AWS CLI Konfigura

tionsdatei (standard mäßig). `~/aws/config`

Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für aws-parallelcluster und packages on. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub

AWS ParallelCluster Version 2.9.0 veröffentlicht	AWS ParallelCluster Version 2.9.0 veröffentlicht. Zu den Änderungen gehören: • Es wurde Unterstützung für mehrere Warteschlangen und mehrere Instanzty pen in der Rechenflotte hinzugefügt, wenn sie mit Slurm Workload Manager verwendet werden. Bei der Verwendung von Warteschl angen werden Auto Scaling Scaling-Gruppen nicht mehr verwendet. Slurm Eine von Amazon Route 53 gehostete Zone wird jetzt mit dem Cluster erstellt und für die DNS-Auflösung von Rechenknoten verwendet, wenn der Slurm Scheduler verwendet wird. Weitere Informationen finden Sie unter Modus mit mehreren Warteschlangen. • Unterstützung für Amazon DCV auf ARM-basierten Graviton-basierten AWS Instances hinzugefügt. • Unterstützung für die Deaktivierung von Hyperthre ading für Instance-Typen hinzugefügt, die CPU-Optio nen in Startvorlagen nicht	11. September 2020

unterstützen (z. B. Instance-Typen). `*.metal`

- Unterstützung für NFS 4 für Dateisysteme hinzugefügt, die vom Hauptknoten aus gemeinsam genutzt werden.
- Die Abhängigkeit von cfn-init beim Bootstrapping von Rechenknoten wurde entfernt, um eine Drosselung zu vermeiden, AWS CloudFormation wenn eine große Anzahl von Knoten dem Cluster beitritt.
- Elastic Fabric AdapterDas Installationsprogramm wurde auf Version 1.9.5 aktualisiert:
  - EFA-Konfiguration: `efa-config-1.4` (von) `efa-config-1.3`
  - EFA-Profil: `efa-profile-1.0.0` (neu)
  - Kernel-Modul: `efa-1.6.0` (keine Änderung)
  - RDMA-Kern: `rdma-core-28.amzn0` (keine Änderung)
  - Libfabric: `libfabric-1.10.1amzn1.1` (keine Änderung)
  - MPI öffnen: `openmpi40-aws-4.0.3` (keine Änderung)

- Auf Version `20.02.4` (von`19.05.5`) aktualisiertSlurm.
- Amazon DCV wurde auf Amazon DCV 2020.1-9012 aktualisiert. Weitere Informationen finden Sie in den Versionshinweisen zu DCV 2020.1-9012 — 24. August 2020 im Amazon DCV-Administratorhandbuch.
- Verwenden Sie beim Mounten gemeinsam genutzter NFS-Laufwerke die private IP-Adresse des Hauptknotens anstelle des Hostnamens.
- Neue Log-Streams wurden zu CloudWatch Logs:`chef-client` ,, `clustermgtd` `computemgtd` `slurm_resume` , und hinzugefügt. `slurm_suspend`
- Unterstützung für Warteschlangennamen in Skripten vor und nach der Installation hinzugefügt.
- Verwenden Sie in der die AWS GovCloud (US) AWS-Regionen On-Demand-Abrechnungsoption von Amazon DynamoDB. Weitere Informationen finden Sie unter On-

	Demand-Modus im Amazon DynamoDB DynamoDB-Entwicklerhandbuch. Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für den aws-parallelcluster und in den Paketen unter. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub	
AWS ParallelCluster Version 2.8.1 veröffentlicht	AWS ParallelCluster Version 2.8.1 veröffentlicht. Zu den Änderungen gehören: • Deaktivieren Sie die Bildschirmsperre für Amazon DCV-Sitzungen, um zu verhindern, dass Benutzer gesperrt werden. • Dieser Fehler wurde behoben`pcluster configure` , wenn ein ARM-basierter AWS Graviton-basierter Instance-Typ hinzugefügt wurde. Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für aws-parallelcluster und in den Paketen unter. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub	4. August 2020

AWS ParallelCluster Version 2.8.0 veröffentlicht	AWS ParallelCluster Version 2.8.0 veröffentlicht. Zu den Änderungen gehören: • Unterstützung für ARM-basierte AWS Graviton-basierte Instanzen (wie und) hinzugefügt. `A1` `C6g` • Unterstützung für die automatischen täglichen Backup-Funktionen von Amazon FSx for Lustre hinzugefügt. Weitere Informationen dazu finden Sie unter `automatic_backup_retention_days`, `copy_tags_to_backups`, `daily_automatic_backup_start_time` und `fsx_backup_id`. • Die Abhängigkeit von Berkshelf wurde von entfernt. `pcluster createami` • Die Robustheit und Benutzererfahrung von wurden verbessert. `pcluster update` Weitere Informationen finden Sie unter Verwenden von `pcluster update`. • Elastic Fabric AdapterDas Installationsprogramm wurde auf 1.9.4 aktualisiert:	23. Juli 2020

  - Kernel-Modul: `efa-1.6.0` (aktualisiert von `efa-1.5.1`)
  - RDMA-Kern: `rdma-core-28.amzn0` (aktualisiert von `rdma-core-25.0`)
  - Libfabric: `libfabric-1.10.1amzn1.1` (aktualisiert von) `libfabric-aws-1.9.0amzn1.1`
  - MPI öffnen: `openmpi40-aws-4.0.3` (keine Änderung)
- Aktualisieren Sie den NVIDIA-Treiber auf Tesla-Version 440.95.01 auf CentOS 6 und Version 450.51.05 auf allen anderen Distributionen.
- Aktualisieren Sie die CUDA-Bibliothek auf allen Distributionen außer 6 auf Version 11.0. CentOS

Weitere Informationen zu den Änderungen finden Sie in den CHANGELOG-Dateien für den aws-parallelcluster und in den Paketen unter. aws-parallelcluster-cookbookaws-parallelcluster-node GitHub

AWS ParallelCluster Version 2.7.0 veröffentlicht	AWS ParallelCluster Version 2.7.0 veröffentlicht. Zu den Änderungen gehören: • `base_os` ist jetzt ein erforderlicher Parameter. • `scheduler` ist jetzt ein erforderlicher Parameter. • Amazon DCV wurde auf Amazon DCV 2020.0 aktualisiert. Weitere Informationen finden Sie unter Amazon DCV veröffentlicht Version 2020.0 mit Surround-Sound 7.1 und Stylus-Unterstützung. Intel MPI wurde auf Version 2019 Update 7 aktualisiert (aktualisiert von Version 2019 Update 6). Weitere Informationen finden Sie in der Intel® MPI-Bibliothek 2019 Update 7. Elastic Fabric Adapter Installer aktualisiert auf 1.8.4: • Kernel-Modul: `efa-1.5.1` (keine Änderung) • RDMA-Kern: `rdma-core-25.0` (keine Änderung)	19. Mai 2020

	• Libfabric: `libfabric-aws-1.9.0amzn1.1` (keine Änderung) • Open MPI: `openmpi40-aws-4.0.3` (aktualisiert von `openmpi40-aws-4.0.2`) • Aktualisieren Sie CentOS 7 AMI auf Version 7.8-2003 (aktualisiert von 7.7-1908) . Weitere Informationen finden Sie in den Versionshinweisen zu CentOS-7 (2003).	
AWS ParallelCluster Version 2.6.1 veröffentlicht	AWS ParallelCluster Version 2.6.1 veröffentlicht. Zu den Änderungen gehören: • Entfernt `cfn-init-cmd` und `cfn-wire` aus den in Amazon Logs gespeicherten CloudWatch Protokollen. Weitere Informationen finden Sie unter Integration mit Amazon CloudWatch Logs.	17. April 2020

AWS ParallelCluster Version 2.6.0 veröffentlicht	AWS ParallelCluster Version 2.6.0 veröffentlicht. Zu den Änderungen gehören: • Support für Amazon Linux 2 hinzugefügt. • Jetzt wird Amazon CloudWatch Logs zum Sammeln von Cluster- und Scheduler-Protokollen verwendet. Weitere Informationen finden Sie unter Integration mit Amazon CloudWatch Logs. • Unterstützung für neue Bereitstellungstypen von Amazon FSx for Lustre hinzugefügt `SCRATCH_2` und `PERSISTENT_1`. Support FSx für Lustre am Ubuntu 18.04 und 16.04. Ubuntu Weitere Informationen finden Sie unter fsx. • Unterstützung für Amazon DCV wurde am Ubuntu 18.04 hinzugefügt. Weitere Informationen finden Sie unter Stellen Sie über Amazon DCV eine Connect zum Hauptknoten her.	27. Februar 2020
AWS ParallelCluster Version 2.5.1 veröffentlicht	AWS ParallelCluster Version 2.5.1 veröffentlicht.	13. Dezember 2019
AWS ParallelCluster Version 2.5.0 veröffentlicht	AWS ParallelCluster Version 2.5.0 veröffentlicht.	18. November 2019

AWS ParallelCluster führt Unterstützung für Intel MPI ein	AWS ParallelCluster Version 2.4.1 führt Unterstützung für Intel MPI ein.	29. Juli 2019
AWS ParallelCluster führt Unterstützung für EFA ein	AWS ParallelCluster Version 2.4.0 führt Unterstützung für Elastic Fabric Adapter (EFA) ein.	11. Juni 2019
AWS ParallelCluster Die Dokumentation wurde auf der Dokumentationsseite AWS veröffentlicht	Die AWS ParallelCluster Dokumentation ist jetzt in 10 Sprachen und sowohl im HTML- als auch im PDF-Format verfügbar.	24. Mai 2018

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.