



AWS ParallelCluster 사용자 가이드 (v3)

# AWS ParallelCluster



# AWS ParallelCluster: AWS ParallelCluster 사용자 가이드 (v3)

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

# Table of Contents

AWS ParallelCluster이란 무엇입니까? .....	1
요금 .....	1
설 AWS ParallelCluster정 .....	2
설정하기 AWS 계정 .....	2
가입하세요. AWS 계정 .....	2
관리자 액세스 권한이 있는 사용자 생성 .....	3
키 페어 생성 .....	4
AWS ParallelCluster CLI 설치 .....	4
가상 환경에 설치 AWS ParallelCluster (권장) .....	5
AWS ParallelCluster pip를 사용하여 비가상 환경에 설치 .....	6
독립 실행형 AWS ParallelCluster 애플리케이션으로 설치 .....	8
설치 후 수행할 단계 .....	9
AWS ParallelCluster UI 설치하기 .....	10
AWS ParallelCluster UI 설치 .....	10
사용자 지정 도메인 생성 .....	13
Amazon Cognito 사용자 풀 옵션 .....	15
AWS ParallelCluster 및 AWS ParallelCluster UI 버전을 식별합니다. ....	18
AWS ParallelCluster UI를 새 AWS ParallelCluster 버전으로 업데이트 .....	18
AWS ParallelCluster UI 비용 .....	18
시작하기 .....	19
AWS ParallelCluster CLI를 사용하여 클러스터 구성 및 생성 .....	19
AWS ParallelCluster UI를 사용하여 클러스터 구성 및 생성 .....	29
클러스터에 연결 .....	30
클러스터에 대한 다중 사용자 액세스 .....	31
Active Directory 생성 .....	32
AD 도메인을 사용하여 클러스터를 생성합니다. ....	32
AD 도메인과 통합된 클러스터에 로그인합니다. ....	35
MPI 작업 실행 .....	36
LDAP (S) 클러스터 구성의 예 AWS Managed Microsoft AD .....	37
모범 사례 .....	41
모범 사례: 헤드 노드 인스턴스 유형 선택 .....	41
모범 사례: 네트워크 성능 .....	41
모범 사례: 예산 알림 .....	43
모범 사례: 클러스터를 새 AWS ParallelCluster 마이너 또는 패치 버전으로 이동 .....	43

AWS ParallelCluster 2.x에서 3.x로 이동 .....	44
사용자 지정 부트스트랩 작업 .....	44
AWS ParallelCluster 2.x와 3.x는 다른 구성 파일 구문을 사용합니다. ....	45
포괄적인 언어 .....	51
스케줄러 지원 .....	51
AWS ParallelCluster CLI .....	51
IMDS 구성 업데이트 .....	54
AWS ParallelCluster 지원 리전 .....	54
사용 AWS ParallelCluster .....	56
AWS ParallelCluster UI .....	57
AWS Lambda 에서 VPC 구성 AWS ParallelCluster .....	58
AWS Identity and Access Management 권한 입력 AWS ParallelCluster .....	60
AWS ParallelCluster Amazon EC2 인스턴스 역할 .....	61
AWS ParallelCluster 사용자 정책 예시 <code>pcluster</code> .....	61
AWS ParallelCluster IAM 리소스 관리를 위한 사용자 예제 정책 .....	76
AWS ParallelCluster IAM 권한을 관리하기 위한 구성 파라미터 .....	82
네트워크 구성 .....	96
AWS ParallelCluster 단일 퍼블릭 서브넷에서 .....	98
AWS ParallelCluster 서브넷 2개 사용 .....	99
AWS ParallelCluster 를 사용하여 연결된 단일 프라이빗 서브넷에서 AWS Direct Connect ...	101
AWS ParallelCluster 스케줄러 포함 AWS Batch .....	102
AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서 .....	104
로그인 노드 .....	110
사용자 지정 부트스트랩 작업 .....	113
구성 .....	115
인수 .....	119
사용자 지정 부트스트랩 작업이 포함된 예제 클러스터 .....	119
IMDSv2의 사용자 지정 부트스트랩 스크립트 업데이트 예제 .....	121
IMDSv1의 구성 업데이트 예제 .....	121
Amazon S3 작업 .....	122
예제 .....	122
스팟 인스턴스 작업 .....	123
시나리오 1: 실행 중인 작업이 없는 스팟 인스턴스가 중단됨 .....	124
시나리오 2: 단일 노드 작업을 실행하는 스팟 인스턴스가 중단됨 .....	124
시나리오 3: 다중 노드 작업을 실행하는 스팟 인스턴스가 중단됨 .....	124
에서 지원하는 스케줄러 AWS ParallelCluster .....	125

Slurm Workload Manager .....	125
AWS Batch .....	186
공유 스토리지 .....	193
공유된 스토리지 구성 .....	196
공유된 스토리지 작업 .....	199
할당량 .....	203
태그 지정 .....	203
모니터링 AWS ParallelCluster 및 로그 .....	207
Amazon CloudWatch Logs와 통합 .....	208
아마존 CloudWatch 대시보드 .....	211
클러스터 지표에 대한 Amazon CloudWatch 경보 .....	213
AWS ParallelCluster 로그 교체 구성 .....	216
pcluster CLI 로그 .....	217
Amazon EC2 콘솔 출력 로그 .....	217
AWS ParallelCluster UI 및 AWS ParallelCluster 런타임 로그 검색 .....	219
로그 검색 및 보존 .....	220
AWS CloudFormation 사용자 지정 리소스 .....	223
에서 호스팅하는 제공자 스택 AWS ParallelCluster .....	224
클러스터 리소스 .....	226
클러스터 작업 .....	229
사용자 지정 리소스가 AWS ParallelCluster 포함된 문제 해결 스택 .....	229
Elastic Fabric Adapter .....	230
인텔 활성화 MPI .....	230
AWS ParallelCluster API .....	232
AWS ParallelCluster API 설명서 .....	232
를 사용하여 배포하십시오. AWS CLI .....	233
API 업데이트 .....	235
API 호출 AWS ParallelCluster .....	236
API 로그 및 지표에 액세스 .....	238
AWS ParallelCluster 테라폼용 .....	239
NICE DCV를 통해 헤드 노드에 연결합니다. ....	239
NICE DCV HTTPS 인증서 .....	240
NICE DCV 라이선스 .....	240
pcluster update-cluster 사용하기 .....	240
업데이트 정책: 정의 .....	241
pcluster update-cluster 예제 .....	244

AWS ParallelCluster AMI 사용자 지정 .....	246
AWS ParallelCluster AMI 사용자 지정 고려 사항 .....	247
사용자 지정 구성 요소 검증 테스트 수행 .....	247
디버깅을 지원하는 pcluster 명령을 사용하여 Image Builder 프로세스를 모니터링합니 다. ....	248
기타 고려 사항 .....	249
온디맨드 용량 예약 (ODCR) 으로 인스턴스 시작 .....	249
ODCR을 다음과 같이 사용할 수 있습니다. AWS ParallelCluster .....	250
용량 블록 (CB) 으로 인스턴스 시작 .....	258
CB 사용: AWS ParallelCluster .....	258
AMI 패치 적용 및 Amazon EC2 인스턴스 교체 .....	260
헤드 노드 인스턴스 업데이트 또는 교체 .....	261
휘발성 드라이브의 데이터를 저장합니다. ....	261
클러스터의 헤드 노드 중지 및 시작 .....	261
운영 체제 .....	263
운영 체제 고려 사항 .....	263
에 대한 참조 AWS ParallelCluster .....	266
AWS ParallelCluster 버전 3 CLI 명령 .....	266
pcluster .....	267
pcluster3-config-converter .....	310
구성 파일 .....	311
클러스터 구성 파일 .....	311
빌드 이미지 구성 파일 .....	436
AWS ParallelCluster API 레퍼런스 .....	445
buildImage .....	446
createCluster .....	451
deleteCluster .....	456
deleteClusterInstances .....	459
deleteImage .....	461
describeCluster .....	464
describeClusterInstances .....	472
describeComputeFleet .....	476
describeImage .....	477
getClusterLog이벤트 .....	484
getClusterStack이벤트 .....	488
getImageLog이벤트 .....	492

getImageStack이벤트 .....	496
listClusters .....	500
listClusterLog스트림 .....	504
listImageLog스트림 .....	508
listImages .....	512
listOfficialImages .....	515
updateCluster .....	518
updateComputeFleet .....	524
AWS ParallelCluster Python 라이브러리 API .....	526
AWS ParallelCluster Python 라이브러리 권한 부여 .....	527
AWS ParallelCluster Python 라이브러리를 설치합니다. ....	527
클러스터 API 작업 .....	527
컴퓨팅 플릿 API 작업 .....	531
클러스터 및 스택 로그 작업 .....	533
이미지 API 작업 .....	536
이미지 및 스택 로그 작업 .....	538
예제 .....	541
AWS ParallelCluster Python 라이브러리를 AWS Lambda .....	542
AWS ParallelCluster 작동 방식 .....	544
AWS ParallelCluster 프로세스 .....	544
clustermgtd .....	544
clusterstatusmgtd .....	545
computemgtd .....	545
AWS 에서 사용하는 서비스 AWS ParallelCluster .....	545
Amazon API Gateway .....	546
AWS Batch .....	547
AWS CloudFormation .....	547
아마존 CloudWatch .....	547
아마존 CloudWatch 이벤트 .....	548
아마존 CloudWatch 로그 .....	548
AWS CodeBuild .....	548
Amazon DynamoDB .....	548
Amazon Elastic Block Store .....	549
Amazon Elastic Compute Cloud .....	549
Amazon Elastic 컨테이너 레지스트리 .....	549
Amazon EFS .....	549

Amazon FSx for Lustre .....	550
ONTAP용 아마존 NetApp FSx .....	550
Amazon FSx for OpenZFS .....	550
AWS Identity and Access Management .....	550
AWS Lambda .....	551
Amazon RDS .....	551
Amazon Route 53 .....	551
Amazon Simple Notification Service .....	551
Amazon Simple Storage Service(S3) .....	552
Amazon VPC .....	552
Elastic Fabric Adapter .....	552
EC2 Image Builder .....	552
NICE DCV .....	553
AWS ParallelCluster내부 디렉터리 .....	553
자습서 .....	554
에서 첫 작업 실행 AWS ParallelCluster .....	554
설치 확인 .....	555
첫 번째 클러스터 생성 .....	555
헤드 노드에 로그인 .....	556
Slurm을 사용하여 첫 번째 작업 실행 .....	557
사용자 지정 AWS ParallelCluster AMI 구축 .....	558
AWS ParallelCluster AMI를 사용자 지정하는 방법 .....	559
사용자 지정 AWS ParallelCluster AMI 구축 .....	559
AWS ParallelCluster AMI 수정 .....	566
Active Directory 통합 .....	568
AWS KMS 키를 사용한 공유 스토리지 암호화 구성 .....	599
정책을 생성합니다. ....	600
클러스터 구성 및 생성 .....	600
다중 대기열 모드 클러스터에서 작업 실행 .....	602
클러스터 구성 .....	603
클러스터 생성 .....	604
헤드 노드에 로그인 .....	605
다중 대기열 모드에서 작업 실행 .....	606
AWS ParallelCluster API 사용하기 .....	609
Slurm 화계를 사용하여 클러스터 생성 .....	624
1단계: VPC 및 서브넷 생성 AWS ParallelCluster .....	625



2단계: 데이터베이스 스택 생성 .....	625
3단계: Slurm 회계가 활성화된 클러스터 생성 .....	625
외부 Slurmdbd 어카운팅을 사용하여 클러스터 생성 .....	626
1단계: Slurmdbd 스택 생성 .....	627
2단계: 외부 Slurmdbd 기능이 활성화된 클러스터 생성 .....	628
이전 AWS Systems Manager 문서 버전으로 되돌리기 .....	629
이전 SSM 문서 버전으로 되돌리기 .....	629
를 사용하여 클러스터 생성 AWS CloudFormation .....	632
CloudFormation 빠른 생성 스택을 통한 클러스터 생성 .....	632
AWS CloudFormation 명령줄 인터페이스 (CLI) 를 사용한 클러스터 생성 .....	634
CloudFormation 클러스터 출력 보기 .....	636
클러스터에 액세스 .....	637
정리 .....	637
테라폼으로 ParallelCluster API 배포하기 .....	638
테라폼 프로젝트 정의 .....	638
API 배포 .....	640
필요한 권한 .....	640
테라폼으로 클러스터 만들기 .....	643
테라폼 프로젝트 정의 .....	644
클러스터 배포 .....	650
필요한 권한 .....	651
테라폼으로 사용자 지정 AMI 생성 .....	652
테라폼 프로젝트 정의 .....	652
AMI 배포하기 .....	655
필요한 권한 .....	656
AWS ParallelClusterID 센터와의 UI 통합 .....	657
IAM Identity Center 활성화 .....	657
IAM ID 센터에 애플리케이션 추가 .....	660
AWS ParallelCluster 문제 해결 .....	668
클러스터를 생성하려는 경우 .....	669
failureCode가 OnNodeConfiguredExecutionFailure일 시 .....	669
failureCode가 OnNodeConfiguredDownloadFailure일 시 .....	669
failureCode가 OnNodeConfiguredFailure일 시 .....	670
failureCode가 OnNodeStartExecutionFailure일 시 .....	670
failureCode가 OnNodeStartDownloadFailure일 시 .....	670
failureCode가 OnNodeStartFailure일 시 .....	671

failureCode가 EbsMountFailure일 시 .....	671
failureCode가 EfsMountFailure일 시 .....	671
failureCode가 FsxMountFailure일 시 .....	672
failureCode가 RaidMountFailure일 시 .....	672
failureCode가 AmiVersionMismatch일 시 .....	672
failureCode가 InvalidAmi일 시 .....	673
failureCode가 HeadNodeBootstrapFailure이며 failureReason이 헤드 노드 설정 에 실패했습니다. ....	673
failureCode가 HeadNodeBootstrapFailure이며 failureReason이 클러스터 생성 시간이 초과되었습니다. ....	674
failureCode가 HeadNodeBootstrapFailure이며 failureReason이 헤드 노드 부트 스트랩에 실패했습니다. ....	675
failureCode가 ResourceCreationFailure일 시 .....	675
failureCode가 ClusterCreationFailure일 시 .....	675
WaitCondition timed out... CloudFormation 스택에서 보기 .....	676
Resource creation cancelled CloudFormation 스택에서 보기 .....	676
AWS CloudFormation 스택의 오류 확인 Failed to run cfn-init... 또는 기타 오류 ..	676
INFO: Waiting for static fleet capacity provisioning로 끝나는 chef- client.log이 표시되는 경우 .....	676
Failed to run preinstall or postinstall in cfn-init.log가 표시되는 경 우 .....	676
This AMI was created with xxx, but is trying to be used with xxx... CloudFormation 스택에서 보기 .....	676
This AMI was not baked by AWS ParallelCluster... CloudFormation 스택에 서 보기 .....	676
pcluster create-cluster 명령이 로컬에서 실행되지 않는 경우 .....	677
추가 지원 .....	677
작업을 실행하려는 경우 .....	677
srun 대화형 작업이 srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf 오류가 발생하여 실패합니다. ....	677
작업이 squeue 명령을 실행한 CF 상태에서 멈췄습니다. ....	678
대규모 작업을 실행한 후 nfsd: too many open connections, consider increasing the number of threads in /var/log/messages가 표시된 경우 .....	678
MPI 작업 실행 .....	678
클러스터를 업데이트하려는 경우 .....	679
pcluster update-cluster 명령이 로컬에서 실행되지 않습니다. ....	679

pcluster describe-cluster 명령으로 clusterStatus가 UPDATE_FAILED로 표시되는 경우 .....	679
클러스터 업데이트 제한 시간이 초과되었습니다. ....	679
스토리지에 액세스하려는 경우 .....	680
외부 Amazon FSx for Lustre 파일 시스템 사용 .....	680
외부 Amazon Elastic File System 파일 시스템 사용 .....	680
클러스터를 삭제하려는 경우 .....	680
pcluster delete-cluster 명령이 로컬에서 실행되지 않습니다. ....	680
클러스터 스택 삭제에 실패했습니다. ....	680
AWS ParallelCluster API 스택 업그레이드 시도 중 .....	680
컴퓨팅 노드 초기화 오류가 표시되는 경우 .....	680
clustermgtd.log에서 Node bootstrap error가 표시되는 경우 .....	680
온디맨드 용량 예약(ODCR) 또는 영역별 예약 인스턴스를 구성했습니다. ....	681
작업 실행 실패 시 slurm_resume.log에서 또는 클러스터 실행 실패 시 clustermgtd.log에서 An error occurred (VcpuLimitExceeded)이 표시되는 경우 .....	682
작업 실행 실패 시 slurm_resume.log에서 또는 클러스터 실행 실패 시 clustermgtd.log에서 An error occurred (InsufficientInstanceCapacity)이 표시되는 경우 .....	682
노드가 Reason (Code:InsufficientInstanceCapacity)...으로 DOWN 상태로 표시되는 경우 .....	682
slurm_resume.log에서 cannot change locale (en_US.utf-8) because it has an invalid name가 표시되는 경우 .....	683
이전 시나리오 중 어느 것도 제 상황에 적용되지 않습니다. ....	683
클러스터 상태 지표 문제 해결 .....	683
인스턴스 프로비저닝 오류 그래프 참조 .....	684
비정상 인스턴스 오류 그래프 보기 .....	686
컴퓨팅 플릿 유휴 시간 그래프 보기 .....	687
클러스터 배포 문제 해결 .....	688
에서 AWS CloudFormation 이벤트 보기 CREATE_FAILED .....	689
CLI를 사용하여 로그 스트림을 볼 수 있습니다. ....	691
rollback-on-failure을 사용하여 실패한 클러스터를 다시 생성합니다. ....	693
Terraform을 사용한 클러스터 배포 문제 해결 .....	694
ParallelCluster API를 찾을 수 없습니다. ....	694
사용자는 ParallelCluster API를 호출할 권한이 없습니다. ....	695
규모 조정 문제 해결 .....	695

디버깅을 위한 키 로그 .....	696
작업 실행 실패 시 <code>slurm_resume.log</code> 에서 또는 클러스터 실행 실패 시 <code>clustermgtd.log</code> 에서 <code>InsufficientInstanceCapacity</code> 오류가 표시되는 경우 .....	682
노드 초기화 문제 해결 .....	699
예상치 못한 노드 교체 및 종료 문제 해결 .....	701
문제가 있는 인스턴스 및 노드 교체, 종료 또는 전원 끄기 .....	703
대기열(파티션) <code>Inactive</code> 상태 .....	703
기타 알려진 노드 및 작업 문제 해결 .....	703
배치 그룹 및 인스턴스 시작 문제 .....	704
교체할 수 없는 디렉터리 .....	704
NICE DCV 문제 해결 .....	704
NICE DCV에 대한 로그 .....	704
Ubuntu NICE DCV 문제 .....	705
AWS Batch 통합을 통해 클러스터의 문제 해결 .....	705
헤드 노드 문제 .....	706
컴퓨팅 문제 .....	706
작업 실패 .....	706
엔드포인트 URL의 연결 시간 초과 오류 .....	706
Active Directory와의 다중 사용자 통합 문제 해결 .....	706
Active Directory 관련 문제 해결 .....	707
디버그 모드 활성화 .....	708
LDAPS에서 LDAP로 이동하는 방법 .....	708
LDAPS 서버 인증서 검증을 비활성화하는 방법 .....	708
암호 대신 SSH 키를 사용하여 로그인하는 방법 .....	709
사용자 암호 및 만료된 암호를 재설정하는 방법 .....	709
조인한 도메인을 확인하는 방법 .....	710
인증서 문제를 해결하는 방법 .....	710
Active Directory와의 통합이 제대로 작동하는지 확인하는 방법 .....	712
컴퓨팅 노드 로그인 문제를 해결하는 방법 .....	713
다중 사용자 환경에서 SimCenter StarCCM+ 작업의 알려진 문제 .....	713
사용자 이름 확인과 관련된 알려진 문제 .....	714
홈 디렉터리 생성 문제를 해결하는 방법 .....	714
사용자 지정 AMI 문제 해결 .....	715
<code>cfn-hup</code> 이 실행 중이 아닐 때의 클러스터 업데이트 제한 시간 문제 해결 .....	716
네트워크 문제 해결 .....	717
단일 퍼블릭 서브넷 안의 클러스터 문제 .....	717

onNodeUpdated 사용자 지정 작업에서 클러스터 업데이트가 실패한 경우 .....	717
사용자 지정 Slurm 구성에서 오류가 표시되는 경우 .....	717
클러스터 알람 .....	718
추가 지원 .....	718
AWS ParallelCluster 지원 정책 .....	719
보안 .....	720
에서 사용하는 서비스의 보안 정보 AWS ParallelCluster .....	720
데이터 보호 .....	721
데이터 암호화 .....	722
다음 사항도 참조하세요. ....	723
ID 및 액세스 관리 .....	723
규정 준수 확인 .....	724
TLS 1.2 적용 .....	725
현재 지원되는 프로토콜 확인 .....	725
OpenSSL 및 Python 컴파일 .....	726
릴리스 정보 및 문서 기록 .....	728
.....	dcccxxii

# AWS ParallelCluster이란 무엇입니까?

AWS ParallelCluster는 AWS 클라우드에서 고성능 컴퓨팅(HPC) 클러스터를 배포하고 관리할 수 있는 AWS 지원 오픈 소스 클러스터 관리 도구입니다. 필요한 컴퓨팅 리소스, 스케줄러 및 공유 파일 시스템을 자동으로 설정합니다. AWS ParallelCluster는 AWS Batch 및 Slurm 스케줄러와 함께 사용할 수 있습니다.

AWS ParallelCluster를 사용하면 개념 증명 및 프로덕션 HPC 컴퓨팅 환경을 신속하게 구축하고 배포할 수 있습니다. AWS ParallelCluster를 기반으로, 전체 DNA 시퀀싱 워크플로우를 자동화하는 Genomics 포털과 같은 더 높은 수준의 워크플로우를 빌드하고 배포할 수 있습니다.

다음 방법을 사용하여 AWS ParallelCluster에 액세스할 수 있습니다.

- [AWS ParallelCluster 명령줄 인터페이스\(CLI\)](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster UI](#)(릴리스 3.5.0부터 추가됨)
- [AWS ParallelCluster Python 라이브러리 API](#)(릴리스 3.5.0부터 추가됨)
- [AWS CloudFormation 사용자 지정 리소스](#)로서(릴리스 3.6.0부터 추가됨)

## 요금

AWS ParallelCluster 명령줄 인터페이스(CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 내용은 [AWS에서 사용하는 서비스 AWS ParallelCluster](#) 항목을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 내용은 [AWS ParallelCluster UI 비용](#) 항목을 참조하세요.

# 설 AWS ParallelCluster정

## 주제

- [설정하기 AWS 계정](#)
- [키 페어 생성](#)
- [AWS ParallelCluster 명령줄 인터페이스 \(CLI\) 설치](#)
- [설치 후 수행할 단계](#)
- [AWS ParallelCluster UI 설치하기](#)
- [시작하기 AWS ParallelCluster](#)
- [클러스터에 대한 다중 사용자 액세스](#)
- [모범 사례](#)
- [AWS ParallelCluster 2.x에서 3.x로 이동](#)
- [AWS ParallelCluster 지원 리전](#)

## 설정하기 AWS 계정

사용할 AWS 계정을 설정합니다 AWS ParallelCluster.

### 가입하세요. AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. <https://portal.aws.amazon.com/billing/signup>을 여세요.
2. 온라인 지시 사항을 따르세요.

등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니](#)다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

## 관리자 액세스 권한이 있는 사용자 생성

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

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 [AWS Management Console](#)소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM [사용 설명서의 AWS 계정 루트 사용자 \(콘솔\)에 대한 가상 MFA 디바이스 활성화를 참조](#)하십시오.

### 관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리자 액세스 권한을 부여합니다.

를 ID 소스로 사용하는 방법에 대한 자습서는 사용 [설명서의 기본값으로 IAM Identity Center 디렉터리사용자 액세스 구성](#)을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

### 관리 액세스 권한이 있는 사용자 로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM IDentity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자를 사용하여 [로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인](#)을 참조하십시오.AWS 로그인



## 추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Create a permission set](#)를 참조하세요.

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

지침은 AWS IAM Identity Center 사용 설명서의 [Add groups](#)를 참조하세요.

## 키 페어 생성

클러스터를 배포하려면 Amazon EC2 인스턴스를 AWS ParallelCluster 시작하여 클러스터 헤드 노드와 컴퓨팅 노드를 생성합니다. 작업 실행 및 모니터링, 사용자 관리와 같은 클러스터 작업을 수행하려면 클러스터 헤드 노드에 액세스할 수 있어야 합니다. SSH를 사용하여 헤드 노드 인스턴스에 액세스할 수 있는지 확인하려면 Amazon EC2 키 쌍을 사용해야 합니다. 키 페어를 생성하는 방법을 알아보려면 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 [키 페어 생성](#)을 참조하세요.

## AWS ParallelCluster 명령줄 인터페이스 (CLI) 설치

AWS ParallelCluster Python 패키지로 배포되며 Python pip 패키지 관리자를 사용하여 설치됩니다. Python 패키지를 설치하는 방법에 대한 지침은 Python 패키징 사용 설명서의 [패키지 설치](#)를 참조하세요.

설치 방법 AWS ParallelCluster:

- [가상 환경에 설치 AWS ParallelCluster \(권장\)](#)
- [AWS ParallelCluster pip를 사용하여 비가상 환경에 설치](#)
- [독립 실행형 AWS ParallelCluster 애플리케이션으로 설치](#)

최신 CLI의 버전 번호는 [릴리스 페이지에서](#) 확인할 수 있습니다. GitHub 이 안내서의 명령 예제는 버전 3.6 이상의 Python 버전을 설치했다고 가정합니다. pip 명령 예제에는 pip3 버전이 사용됩니다.

AWS ParallelCluster 2와 AWS ParallelCluster 3을 모두 관리합니다.

AWS ParallelCluster 2와 AWS ParallelCluster 3을 모두 사용하고 두 패키지의 CLI를 관리하려는 고객의 경우 AWS ParallelCluster 2와 AWS ParallelCluster 3을 서로 다른 [가상 환경에](#) 설치하는 것이 좋습니다. 이렇게 하면 AWS ParallelCluster의 각 버전과 모든 관련 클러스터 리소스를 계속 사용할 수 있습니다.

## 가상 환경에 설치 AWS ParallelCluster (권장)

요구 사항 버전이 다른 pip 패키지와 충돌하지 않도록 가상 환경에 설치하는 AWS ParallelCluster 것이 좋습니다.

### 사전 조건

- AWS ParallelCluster Python 3.7 이상이 필요합니다. 아직 설치하지 않았다면 [python.org](https://python.org)에서 [플랫폼과 호환되는 버전을 다운로드](#)하세요.

### 가상 AWS ParallelCluster 환경에 설치하려면

1. `virtualenv`를 설치하지 않은 경우 `pip3`을 사용하여 `virtualenv`를 설치합니다. `python3 -m virtualenv help`에 도움말 정보가 표시되면 2단계로 이동합니다.

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

`exit`를 실행하여 현재 터미널 창에서 나가서 새 터미널 창을 열어 환경에 변경 사항을 적용합니다.

2. 가상 환경을 생성하고 이름을 지정합니다.

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

또는 `-p` 옵션을 사용하여 특정 Python 버전을 지정할 수 있습니다.

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

3. 새 가상 환경을 활성화합니다.

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

4. 가상 AWS ParallelCluster 환경에 설치하세요.

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

5. 노드 버전 관리자와 최신 장기 지원 (LTS) Node.js 버전을 설치합니다. AWS Cloud Development Kit (AWS CDK) (AWS CDK) 템플릿 생성을 CloudFormation 위해서는 Node.js 가 필요합니다.

**Note**

플랫폼에서 Node.js 설치가 작동하지 않는 경우 최신 버전의 LTS가 아닌 이전 버전의 LTS를 설치해 보세요. 자세한 내용은 [Node.js 출시 일정](#) 및 [AWS CDK 사전 요구 사항](#)을 참조하세요.

Node.js 설치 명령 예시:

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

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

6. 제대로 AWS ParallelCluster 설치되었는지 확인하십시오.

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

deactivate 명령을 사용하여 가상 환경을 종료할 수 있습니다. 새 세션을 시작할 때마다 [환경을 다시 활성화](#)해야 합니다.

최신 버전으로 업그레이드하려면 설치 명령을 다시 실행합니다. AWS ParallelCluster

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

## AWS ParallelCluster pip를 사용하여 비가상 환경에 설치

### 사전 조건

- AWS ParallelCluster Python 3.7 이상이 필요합니다. 아직 설치하지 않았다면 [python.org](#)에서 플랫폼과 호환되는 버전을 다운로드하세요.

## 설치 AWS ParallelCluster

1. pip설치에 사용합니다 AWS ParallelCluster.

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

--user스위치를 사용하면 AWS ParallelCluster 에 pip ~/.local/bin 설치합니다.

2. 노드 버전 관리자와 최신 장기 지원 (LTS) Node.js 버전을 설치합니다. AWS Cloud Development Kit (AWS CDK) (AWS CDK) 템플릿 생성을 CloudFormation 위해서는 Node.js 가 필요합니다.

### Note

플랫폼에서 Node.js 설치가 작동하지 않는 경우 최신 버전의 LTS가 아닌 이전 버전의 LTS를 설치해 보세요. 자세한 내용은 [Node.js 출시 일정](#) 및 [AWS CDK](#) 사전 요구 사항을 참조하세요.

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

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

3. 제대로 AWS ParallelCluster 설치되었는지 확인하십시오.

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

4. 최신 버전으로 업그레이드하려면 설치 명령을 다시 실행합니다.

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

## 독립 실행형 AWS ParallelCluster 애플리케이션으로 설치

사용자 환경에 독립 실행형 AWS ParallelCluster 애플리케이션으로 설치합니다. 다음 섹션의 사용 가능한 AWS ParallelCluster OS에 설치하기 위한 지침을 따르십시오.

### 사전 조건

- 사용 가능한 설치 프로그램 버전과 호환되는 운영 체제가 있는 환경

#### Note

AWS ParallelCluster NodeJS가 필요합니다. AWS ParallelCluster 설치 프로그램에는 NodeJS (v18) 의 번들 버전이 포함되어 있으며, 아직 없을 경우 설치됩니다. 시스템이 NodeJS v18과 호환되지 않는 경우 설치하기 전에 NodeJS를 설치해야 합니다. AWS ParallelCluster

### Linux

#### Linux x86 (64-bit)

사용자 환경에 설치하십시오. AWS ParallelCluster

1. 최신 [pcluster 설치 프로그램](#)을 다운로드하세요.
2. 다음 명령을 사용하여 설치 프로그램 번들의 압축을 풀고 설치합니다 AWS ParallelCluster .

```
$ unzip pcluster-installer-bundle-3.10.1.713-node-v18.20.3-Linux_x86_64-signed.zip
-d pcluster-installer-bundle
$ cd pcluster-installer-bundle
$ chmod +x install_pcluster.sh
```

3. 다음 설치 스크립트를 실행합니다.

```
$ bash install_pcluster.sh
```

4. 제대로 AWS ParallelCluster 설치되었는지 확인하십시오.

```
$ pcluster version
{
  "version": "3.10.1"
```

```
}
```

## pcluster 설치 오류 문제 해결

- 4단계에서 AWS ParallelCluster 버전이 반환되지 않으면 다음 예와 같이 터미널이나 `source bash_profile` 를 다시 시작하여 새 바이너리 디렉토리를 포함하도록 PATH 변수를 업데이트하십시오.

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

- `pcluster` 설치를 사용하여 S3 URI가 아닌 HTTPS 리소스로 CustomActions가 지정된 클러스터를 생성하는 경우 이러한 리소스가 검증되지 않을 수 있다는 WARNING 메시지([SSL: CERTIFICATE\_VERIFY\_FAILED])가 표시될 수 있습니다. 이는 알려진 문제로 인해 발생하며 지정된 리소스의 신뢰성을 신뢰할 수 있는 경우 이 경고를 무시해도 됩니다.

## 이전 설치 프로그램 번들 버전

- None

## 설치 후 수행할 단계

를 실행하여 제대로 AWS ParallelCluster 설치되었는지 확인할 수 [pcluster version](#) 있습니다.

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

AWS ParallelCluster 정기적으로 업데이트됩니다. 최신 버전으로 업데이트하려면 설치 명령을 다시 실행합니다. AWS ParallelCluster 최신 버전에 대한 AWS ParallelCluster 자세한 내용은 [AWS ParallelCluster 릴리스 노트](#)를 참조하십시오.

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

제거하려면 AWS ParallelCluster를 사용하십시오 `pip3 uninstall`.

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

Python 및 pip3가 없는 경우 사용 환경에 해당하는 절차를 사용합니다.

## AWS ParallelCluster UI 설치하기

AWS ParallelCluster UI는 콘솔과 유사한 경험을 제공하면서 AWS ParallelCluster `pccluster` CLI를 미러링하는 웹 기반 사용자 인터페이스입니다. AWS 계정에 AWS ParallelCluster UI를 설치하고 액세스할 수 있습니다. 실행하면 AWS ParallelCluster UI가 사용자의 AWS 계정에 있는 Amazon API Gateway에 호스팅된 AWS ParallelCluster API의 인스턴스에 액세스합니다. AWS ParallelCluster UI에 대한 자세한 내용은 [AWS ParallelCluster UI](#)를 참조하세요.

사전 조건:

- AWS 계정
- AWS Management Console에 대한 액세스

주제

- [AWS ParallelCluster UI 설치](#)
- [사용자 지정 도메인 생성](#)
- [Amazon Cognito 사용자 풀 옵션](#)
- [AWS ParallelCluster 및 AWS ParallelCluster UI 버전을 식별합니다.](#)
- [AWS ParallelCluster UI를 새 AWS ParallelCluster 버전으로 업데이트](#)
- [AWS ParallelCluster UI 비용](#)

## AWS ParallelCluster UI 설치

AWS ParallelCluster UI 인스턴스를 설치하려면 AWS 리전에 클러스터를 생성할 때 사용할 AWS CloudFormation 빠른 생성 링크를 선택합니다. 빠른 생성 URL을 사용하면 스택 생성 마법사로 이동하여 빠른 생성 스택 템플릿 입력을 제공하고 스택을 배포할 수 있습니다. 스택 CloudFormation 빠른 생성에 대한 자세한 내용은 사용 설명서의 스택에 [대한 빠른 생성 링크 만들기를](#) 참조하십시오. AWS CloudFormation

### Note

클러스터를 생성 및 편집하거나 AWS ParallelCluster UI를 설치하는 데 사용한 것과 동일한 AWS ParallelCluster 버전으로만 이미지를 빌드할 수 있습니다.

## 리전별 AWS ParallelCluster UI 빠른 생성 링크

### UI 빠른 생성 링크

[us-east-1](#)

[us-east-2](#)

[us-west-1](#)

[us-west-2](#)

[eu-west-1](#)

[eu-west-2](#)

[eu-west-3](#)

[eu-central-1](#)

[eu-north-1](#)

[me-south-1](#)

[sa-east-1](#)

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

[ap-southeast-1](#)

[ap-southeast-2](#)

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



AWS CloudFormation 빠른 생성 링크를 사용하여 중첩된 Amazon Cognito, API Gateway 및 Amazon EC2 Systems Manager 스택이 포함된 AWS ParallelCluster UI 스택을 배포할 수 있습니다.

1. AWS Management Console에 로그인합니다.
2. 이 섹션의 시작 부분에 있는 표에서 AWS 리전 빠른 생성 링크를 선택하여 AWS ParallelCluster UI를 배포하세요. 그러면 콘솔의 스택 CloudFormation 생성 마법사로 이동합니다.
3. 관리자 이메일에 사용할 유효한 이메일 주소를 입력합니다.

배포가 성공적으로 완료되면, AWS ParallelCluster UI에서 이 이메일 주소로 임시 암호를 보냅니다. 임시 암호를 사용하여 AWS ParallelCluster UI에 액세스할 수 있습니다. 임시 암호를 저장하거나 사용하기 전에 이메일을 삭제한 경우, 스택을 삭제하고 AWS ParallelCluster UI를 다시 설치해야 합니다.

4. 양식의 나머지 부분은 비워 두거나 (선택 사항) 파라미터 값을 입력하여 AWS ParallelCluster UI 빌드를 사용자 지정합니다.
5. 스택 이름은 이후 단계에서 사용할 수 있도록 기록해 둡니다.
6. 기능으로 이동합니다. CloudFormation 기능에 동의하세요.
7. 생성을 선택하세요. AWS ParallelCluster API 및 AWS ParallelCluster UI의 배포를 완료하는 데 약 15분 정도 걸립니다.
8. 스택이 생성되면 스택 세부 정보를 볼 수 있습니다.
9. 배포가 완료되면 입력한 주소로 전송된 관리자 이메일을 엽니다. 여기에는 AWS ParallelCluster UI에 액세스하는 데 사용하는 임시 암호가 들어 있습니다. 이메일을 영구 삭제하고 AWS ParallelCluster UI에 아직 로그인하지 않은 경우 생성한 AWS ParallelCluster UI 스택을 삭제하고 AWS ParallelCluster UI를 다시 설치해야 합니다.
10. AWS CloudFormation 콘솔 스택 목록에서, 전 단계에서 적어 놓은 스택 이름 링크를 선택합니다.
11. 스택 세부 정보에서 출력을 선택하고 **Stackname** URL이라는 키의 링크를 선택하여 AWS ParallelCluster UI를 엽니다. **Stackname**은 이전 단계에서 기록해 둔 이름입니다.
12. 임시 암호를 입력합니다. 단계에 따라 암호를 직접 만들고 로그인합니다.
13. 이제 사용자가 선택한 AWS 리전에 있는 AWS ParallelCluster UI의 홈 페이지에 있습니다.
14. AWS ParallelCluster UI를 사용하여 시작하려면 [AWS ParallelCluster UI를 사용하여 클러스터 구성 및 생성](#) 섹션을 참조하세요.

**Note**

PCUI 세션의 기본 지속 시간은 5분으로, 이는 PCUI 2023.12.0을 기준으로 Cognito에서 제공하는 최소값입니다. 따라서 Cognito 사용자 풀에서 제거된 사용자는 세션이 만료될 때까지 시스템에 계속 액세스할 수 있을 것으로 예상됩니다.

## 사용자 지정 도메인 생성

AWS ParallelCluster UI에 대한 사용자 지정 도메인을 생성하는 방법을 알아봅니다. UI는 사용자의 AWS 계정에 있는 Amazon API Gateway에서 호스팅됩니다. API Gateway 콘솔을 사용하여 사용자 지정 도메인 이름을 생성할 수 있습니다.

사전 조건:

- AWS 계정이 있습니다.
- 액세스할 수 있는 AWS ParallelCluster UI 인스턴스가 있습니다.
- 도메인을 소유하고 있습니다.
- 기본 도메인 이름 시스템(DNS) 설정을 변경할 수 있습니다.

### 1단계: Amazon API Gateway에 새 도메인 생성

1. AWS Management Console에서 [API Gateway](#)로 이동하여 AWS ParallelCluster UI API 목록을 확인할 수 있습니다.
2. 탐색 창에서 사용자 지정 도메인 이름을 선택합니다.
3. 생성을 선택하세요.
4. 도메인 세부 정보에 도메인 이름을 입력합니다.
5. 엔드포인트 구성에서 기존 ACM 인증서를 선택하거나 새 ACM 인증서 생성을 선택합니다.

(선택 사항)인증서 생성

- a. ACM 콘솔에서 요청을 선택합니다.
- b. 도메인 이름에 사용자의 도메인 이름을 입력합니다.
- c. 검증 방법에서 검증 방법을 선택합니다.

이메일 검증을 선택하면 도메인 등록기관에 등록된 이메일 주소로 이메일이 전송됩니다.

- d. 승인을 선택하여 인증서를 활성화합니다.

## 2단계: API 매핑 설정

1. [API Gateway](#)의 사용자 지정 도메인 이름에서 API 매핑 구성을 선택합니다. `your-domain-name`
2. 사용자 지정 도메인 이름을 선택합니다.
3. 새 매핑 추가를 선택합니다.
4. AWS ParallelCluster UI API, \$default 스테이지 및 저장을 선택합니다.
5. API Gateway 도메인 이름에서 다음 단계에서 사용할 값을 복사합니다.

## 3단계: DNS 설정

- 도메인이 API Gateway 도메인을 가리키는 DNS CNAME 규칙을 생성합니다. 도메인만 입력합니다. 예를 들어, `beta` 또는 `prod`와 같이 스테이지를 추가하지 마세요. `abcde12345`를 API Gateway API ID로 바꾸고 `us-east-2`를 API AWS 리전으로 바꾸세요.

규칙	소스(Source)	대상
CNAME	<code>example.com</code>	<code>d-abcde12345 .execute-api.us-east-2 .amazonaws.com</code>

## 4단계: Amazon Cognito 사용자 풀에 도메인 추가

1. [Amazon Cognito 콘솔](#)로 이동합니다.
2. 사용자 풀 링크를 선택합니다.
3. 앱 통합을 선택합니다.
4. 도메인에서 작업을 선택한 다음, 사용자 지정 도메인 생성을 선택합니다.
5. 사용자 지정 도메인을 입력하고 ACM 인증서를 선택합니다.
6. 그런 다음 사용자 지정 도메인 생성을 선택합니다.

## 5단계: API Gateway 콜백 URL 구성

1. [Amazon Cognito 콘솔](#)로 이동합니다.
2. Amazon Cognito 사용자 풀 앱 통합, 앱 클라이언트 및 분석에서 애플리케이션 링크를 선택합니다.
3. 호스팅된 UI에서 편집을 선택합니다.
4. 허용된 콜백 URL에서 다른 URL 추가를 선택하고 `example.com/login`과 같은 콜백 URL을 입력합니다.

## 6단계: Lambda 함수 구성

1. [Lambda 콘솔](#)로 이동합니다.
2. 탐색 창에서 함수를 선택합니다.
3. 함수 목록을 필터링하여 `ParallelClusterUIFunction`을 찾아 링크를 선택합니다.
4. 구성, 환경 변수를 선택합니다.
5. 편집을 선택합니다.
6. `SITE_URL` 값에는 사용자 지정 도메인을 입력합니다.
7. 도메인(예: `example.com`)으로 이동하고, 인증하여 AWS ParallelCluster UI에 연결합니다.

## Amazon Cognito 사용자 풀 옵션

다음 섹션은 빠른 CloudFormation 생성 링크 또는 빠른 생성 URL을 참조합니다. 빠른 생성 URL을 사용하면 스택 생성 마법사로 이동하여 빠른 생성 스택 템플릿 입력을 제공하고 스택을 배포할 수 있습니다. 스택을 CloudFormation 빠르게 생성하는 방법에 대한 자세한 내용은 사용 설명서의 스택용 [빠른 생성 링크 만들기를](#) 참조하십시오. AWS CloudFormation

여러 AWS ParallelCluster UI 인스턴스에서 사용할 수 있는 Amazon Cognito 사용자 풀을 유지 관리하려면 다음 옵션을 고려해 보세요.

- CloudFormation 중첩된 스택에서 생성된 Amazon Cognito 사용자 풀에 연결되는 기존 AWS ParallelCluster UI 인스턴스를 사용합니다. 이는 빠른 생성 링크를 사용하여 AWS ParallelCluster UI를 배포하고 모든 Amazon Cognito 파라미터를 비워 둘 때 생성되는 것입니다.
- AWS ParallelCluster UI가 배포되기 전에 배포된 독립형 Amazon Cognito 사용자 풀을 사용하세요. 그런 다음 이미 배포한 독립형 Amazon Cognito 사용자 풀에 연결된 새 AWS ParallelCluster UI 인스턴스를 배포합니다. 이렇게 하면 Amazon Cognito 배포를 AWS ParallelCluster UI 배포와 분리할 수

있습니다. 또한 중첩되지 않은 AWS ParallelCluster UI CloudFormation 스택은 업데이트하기가 더 쉽습니다.

## 기존 Amazon Cognito 사용자 풀을 새 AWS ParallelCluster UI 인스턴스와 함께 사용

1. CloudFormation 콘솔에서 여러 AWS ParallelCluster UI 인스턴스와 함께 사용하려는 Amazon Cognito 사용자 풀이 포함된 AWS ParallelCluster UI 스택을 선택합니다.
2. Amazon Cognito 사용자 풀을 생성한 중첩된 스택으로 이동합니다.
3. 출력 탭을 선택합니다.
4. 다음 파라미터의 값을 복사합니다.
  - UserPoolId
  - UserPoolAuthDomain
  - SNSRole
5. 빠른 생성 링크를 사용하여 새 AWS ParallelCluster UI 인스턴스를 배포하고, 복사한 출력값으로 모든 External AWS ParallelCluster UI Amazon Cognito 파라미터를 채웁니다. 이렇게 하면 새 AWS ParallelCluster UI 스택이 새 풀을 생성할 수 없으며 이를 중첩된 스택에서 생성된 기존 Amazon Cognito 사용자 풀에 연결합니다. 파라미터 값이 동일한 후속 새 AWS ParallelCluster UI 인스턴스를 배포하고 이를 Amazon Cognito 사용자 풀에 연결할 수 있습니다.

## 독립형 Amazon Cognito 사용자 풀 생성

AWS ParallelCluster UI Amazon Cognito 리전별 빠른 생성 링크

UI Amazon Cognito 빠른 생성 링크

[us-east-1](#)

[us-east-2](#)

[us-west-1](#)

[us-west-2](#)

[eu-west-1](#)

[eu-west-2](#)

## UI Amazon Cognito 빠른 생성 링크

[eu-west-3](#)

[eu-central-1](#)

[eu-north-1](#)

[me-south-1](#)

[sa-east-1](#)

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

[ap-southeast-1](#)

[ap-southeast-2](#)

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

1. AWS ParallelCluster UI 인스턴스를 배포할 때 사용한 것과 동일한 AWS 리전 레이블이 붙은 빠른 생성 링크를 선택하여 Amazon Cognito 전용 스택을 시작하세요. 이 섹션의 시작 부분에 나오는 빠른 생성 링크를 참조하세요.
2. 스택 생성이 완료되면 출력 탭을 선택하고 다음 파라미터의 값을 복사합니다.
  - UserPoolId
  - UserPoolAuthDomain
  - SNSRole
3. AWS ParallelCluster UI 빠른 시작 링크를 선택하고 모든 External AWS ParallelCluster UI Amazon Cognito 파라미터를 복사한 값으로 채워 새 AWS ParallelCluster UI 인스턴스를 배포합니다. 새 AWS ParallelCluster UI 인스턴스는 독립형 Amazon Cognito 사용자 풀에 연결되며 중첩된 스택이나 새 사용자 풀을 생성하지 않습니다. 파라미터 값이 동일한 후속 새 AWS

ParallelCluster UI 인스턴스를 배포하고 이를 독립형 Amazon Cognito 사용자 풀에 연결할 수 있습니다.

## AWS ParallelCluster 및 AWS ParallelCluster UI 버전을 식별합니다.

1. CloudFormation 콘솔에서 AWS ParallelCluster UI 스택을 선택합니다.
2. 파라미터 탭을 선택합니다.
3. AWS ParallelCluster 버전은 파라미터 버전의 값입니다.
4. AWS ParallelClusterUI 버전은 PublicEcrImageUri값의 끝에 있습니다. 예를 들어 값이 `public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02`이면 버전도 `2023.02`입니다.

## AWS ParallelCluster UI를 새 AWS ParallelCluster 버전으로 업데이트

AWS ParallelCluster UI를 최신 AWS ParallelCluster 버전으로 업데이트하려면 [빠른 생성 링크](#)를 선택하여 새 스택을 시작합니다.

## AWS ParallelCluster UI 비용

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 다음 표에는 AWS ParallelCluster UI가 의존하는 AWS 서비스 및 프리 티어 한도가 나와 있습니다. 일반적인 사용 비용은 매월 1달러 미만인 것으로 추정됩니다.

Service	AWS 프리 티어
Amazon Cognito	월별 활성 사용자 50,000명
Amazon API Gateway	1백만 건의 REST API 직접 호출
AWS Lambda	매월 1백만 건의 무료 요청 및 매월 400,000GB-초의 컴퓨팅 시간
EC2 Image Builder	비용 없음(EC2 제외)
Amazon Elastic Compute Cloud	15분 분량의 일회성 컨테이너 이미지 빌드

Service	AWS 프리 티어
AWS CloudFormation	5GB 데이터(수집, 아카이브 스토리지 및 Logs Insights 쿼리로 스캔한 데이터)

## 시작하기 AWS ParallelCluster

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 웹 기반 사용자 인터페이스 (UI) 를 사용하여 클러스터를 구성하고 생성하여 시작하십시오. AWS ParallelCluster UI는 릴리스 3.5.0에 추가되었습니다.

주제

- [AWS ParallelCluster 명령줄 인터페이스를 사용하여 클러스터를 구성하고 생성합니다.](#)
- [AWS ParallelCluster UI를 사용하여 클러스터 구성 및 생성](#)
- [클러스터에 연결](#)

AWS ParallelCluster 명령줄 인터페이스를 사용하여 클러스터를 구성하고 생성합니다.

설치 AWS ParallelCluster후 다음 구성 단계를 완료하십시오.

AWS 계정에 [pcluster](#) CLI를 실행하는 데 필요한 권한이 포함된 역할이 있는지 확인하십시오. 자세한 정보는 [AWS ParallelCluster 사용자 정책 예시 pcluster](#)을 참조하세요.

AWS 자격 증명을 설정합니다. 자세한 내용은AWS CLI 사용 설명서의 [AWS CLI구성](#)을 참조하세요.

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

AWS 리전 클러스터가 시작되는 위치에는 Amazon EC2 키 페어가 하나 이상 있어야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute 클라우드 사용 설명서의 Amazon Elastic Compute Cloud [키 페어](#)를 참조하십시오.



AWS ParallelCluster 명령줄 인터페이스 (CLI) 를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#) 을 참조하세요.

## 첫 번째 클러스터 구성 및 생성

`pcluster configure` CLI 명령을 사용하여 클러스터를 구성하고 생성하는 데 필요한 모든 정보를 입력하라는 메시지를 표시하는 마법사를 시작하여 첫 번째 클러스터를 생성합니다. 시퀀스의 세부 사항은 AWS Batch 스케줄러로 사용할 때와 사용할 때 다릅니다. Slurm

### Slurm

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

유효한 AWS 리전 식별자 목록에서 클러스터를 AWS 리전 실행할 위치를 선택합니다.

#### Note

AWS 리전 표시된 목록은 계정 파티션을 기반으로 하며 계정에 활성화된 파티션만 포함합니다. AWS 리전 . 계정 활성화에 AWS 리전 대한 자세한 내용은 AWS 리전에서 [관리를](#) 참조하십시오. 표시된 예는 AWS 글로벌 파티션에서 가져온 것입니다. 계정이 이 파티션에 있는 AWS 리전 경우 해당 AWS GovCloud (US) 파티션의 계정만 나열됩니다 (gov-us-east-1 및 gov-us-west-1). 마찬가지로 계정이 AWS 중국 파티션에 있는 경우에도, AN만 cn-north-1 cn-northwest-1 표시됩니다. 에서 AWS 리전 지원하는 전체 목록은 AWS ParallelCluster을 참조하십시오 [AWS ParallelCluster 지원 리전](#).

Allowed values for AWS ## ID:

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

```

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

```

키 페어는 선택 항목의 Amazon Elastic Compute Cloud에 등록된 키 페어에서 선택됩니다 AWS 리전. 키 페어 선택:

```

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

```

클러스터와 함께 사용할 스케줄러를 선택합니다.

```

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

```

운영 체제를 선택합니다.

```

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

```

헤드 노드 인스턴스 유형 선택:

```

Head node instance type [t2.micro]:

```

대기열 구성을 선택합니다. 참고: 동일한 대기열에 있는 여러 컴퓨팅 리소스에 대해서는 인스턴스 유형을 지정할 수 없습니다.

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

EFA가 대규모 인스턴스 간 통신이 필요한 애플리케이션을 추가 비용 없이 실행할 수 있도록 지원합니다.

- [Elastic Fabric Adapter\(EFA\)](#)를 지원하는 인스턴스 유형을 선택합니다.
- [EFA](#)를 활성화합니다.
- 기존의 [배치 그룹](#) 이름을 지정합니다. 이 필드를 비워 두면 AWS ParallelCluster 자동으로 생성됩니다.

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

이전 단계를 완료한 후 기존 VPC를 사용할지 아니면 VPC를 AWS ParallelCluster 생성할지 결정하십시오. 제대로 구성된 VPC가 없는 경우 새 VPC를 만들 수 있습니다. 동일한 퍼블릭 서브넷의 헤드 및 컴퓨팅 노드를 모두 배치하거나 헤드 노드만 퍼블릭 서브넷에 배치하고 모든 컴퓨팅 노드는 프라이빗 서브넷에 둡니다. VPC를 AWS ParallelCluster 생성할 경우 모든 노드를 퍼블릭 서브넷에 포함할지 여부를 결정해야 합니다. 자세한 정보는 [네트워크 구성](#)을 참조하세요.

여러 네트워크 인터페이스 또는 네트워크 카드가 있는 인스턴스 유형을 사용하도록 클러스터를 구성하는 경우 추가 네트워킹 요구 사항은 [네트워크 구성](#)을 참조하세요.

AWS 리전에 허용되는 VPC 수의 할당량에 도달할 수 있습니다. 기본 할당량은 AWS 리전 1개당 VPC 5개입니다. 이 할당량 및 증가 요청 방법에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 및 서브넷](#)을 참조하세요.

#### Important

에서 생성된 VPC는 기본적으로 VPC 흐름 로그를 AWS ParallelCluster 활성화하지 않습니다. VPC 흐름 로그를 사용하여 VPC의 네트워크 인터페이스에서 송수신되는 IP 트래픽에

대한 정보를 캡처할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 흐름 로](#)  
[그](#)를 참조하세요.

VPC를 AWS ParallelCluster 만들 수 있는 경우 모든 노드를 퍼블릭 서브넷에 포함할지 여부를 결정해야 합니다.

### Note

1. Head node in a public subnet and compute fleet in a private subnet 항목을 선택하면 AWS ParallelCluster 항목은 프리 티어 리소스를 지정하더라도 추가 비용이 발생하는 NAT 게이트웨이를 생성합니다.

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

새 VPC를 생성하지 않는 경우 기존 VPC를 선택해야 합니다.

VPC를 AWS ParallelCluster 생성하기로 선택한 경우 나중에 삭제할 수 있도록 VPC ID를 AWS CLI 기록해 두십시오.

```
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

```

VPC를 선택한 후 기존 서브넷을 사용할지 아니면 새 서브넷을 생성할지를 결정합니다.

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

```

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

```

## AWS Batch

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

유효한 AWS 리전 식별자 목록에서 클러스터를 실행할 AWS 리전 위치를 선택합니다.

### Note

AWS 리전 표시된 목록은 계정 파티션을 기반으로 합니다. 계정에서 AWS 리전 활성화된 항목만 포함됩니다. 계정 활성화에 AWS 리전 대한 자세한 내용은 AWS 리전의 [관리를](#) 참조하십시오 AWS 일반 참조. 표시된 예는 AWS 글로벌 파티션에서 가져온 것입니다. 계정이 파티션에 있는 AWS 리전 경우 해당 AWS GovCloud (US) 파티션의 계정만 나열됩니다 (gov-us-east-1 및 gov-us-west-1). 마찬가지로 계정이 AWS 중국 파티션에 있는 경우에도, AN만 cn-north-1 cn-northwest-1 표시됩니다. 에서 AWS 리전 지원하는 전체 목록은 AWS ParallelCluster을 참조하십시오 [AWS ParallelCluster 지원 리전](#).

Allowed values for AWS ## ID:

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

```

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

```

키 페어는 선택한 AWS 리전에서 Amazon EC2로 등록된 키 페어 중에 선택됩니다. 키 페어 선택:

```

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

```

클러스터와 함께 사용할 스케줄러를 선택합니다.

```

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

```

awsbatch을 스케줄러로 선택한 경우 alinux2가 운영 체제로 사용됩니다. 헤드 노드 인스턴스 유형이 입력됩니다.

```

Head node instance type [t2.micro]:

```

대기열 구성을 선택합니다. AWS Batch 스케줄러에는 대기열이 하나만 포함됩니다. 컴퓨팅 노드 클러스터의 최대 크기가 입력됩니다. 이 값은 vCPU에서 측정됩니다.

```

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

```

기존 VPC를 사용할지 아니면 VPC를 AWS ParallelCluster 생성하도록 할지를 결정하세요. 제대로 구성된 VPC가 없는 경우 AWS ParallelCluster 에서 새 VPC를 생성할 수 있습니다. 동일한 퍼블릭

서브넷의 헤드 및 컴퓨팅 노드를 모두 사용하거나 모든 노드가 프라이빗 서브넷에 있는 퍼블릭 서브넷의 헤드 노드만 사용합니다. 하나의 리전에 허용되는 VPC 수의 할당량에 도달할 수 있습니다. 기본 VPC 수는 5개입니다. 이 할당량 및 증가 요청 방법에 대한 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 및 서브넷](#)을 참조하세요.

#### Important

에서 생성된 VPC는 기본적으로 VPC 흐름 로그를 AWS ParallelCluster 활성화하지 않습니다. VPC 흐름 로그를 사용하여 VPC의 네트워크 인터페이스에서 송수신되는 IP 트래픽에 대한 정보를 캡처할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [VPC 흐름 로그](#)를 참조하세요.

VPC를 AWS ParallelCluster 만들 수 있는 경우 모든 노드를 퍼블릭 서브넷에 포함할지 여부를 결정해야 합니다.

#### Note

1. Head node in a public subnet and compute fleet in a private subnet 항목을 선택하면 AWS ParallelCluster 항목은 프리 티어 리소스를 지정하더라도 추가 비용이 발생하는 NAT 게이트웨이를 생성합니다.

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

새 VPC를 생성하지 않는 경우 기존 VPC를 선택해야 합니다.

VPC를 AWS ParallelCluster 생성하기로 선택한 경우 나중에 AWS CLI OR를 사용하여 삭제할 수 있도록 VPC ID를 AWS Management Console 기록해 두십시오.

```
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
```

VPC를 선택한 후 기존 서브넷을 사용할지 아니면 새 서브넷을 생성할지를 결정해야 합니다.

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

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

이전 단계를 완료하면 간단한 클러스터가 VPC로 시작됩니다. VPC는 퍼블릭 IP 주소를 지원하는 기존 서브넷을 사용합니다. 서브넷의 라우팅 테이블은  $0.0.0.0/0 \Rightarrow igw-xxxxxx$ 입니다. 다음 조건을 알아 두세요.

- VPC에는 DNS Resolution = yes 및 DNS Hostnames = yes가 있어야 합니다.
- 또한 VPC에는 AWS 리전에 대한 올바른 domain-name가 있는 DHCP 옵션이 있어야 합니다. 기본 DHCP 옵션 세트는 이미 필수 AmazonProvidedDNS를 지정합니다. 도메인 이름 서버를 두 개 이상 지정하는 경우 Amazon VPC 사용 설명서의 [DHCP 옵션 세트](#)를 참조하세요. 프라이빗 서브넷을 사용하는 경우 NAT 게이트웨이 또는 내부 프록시를 사용하여 컴퓨팅 노드에 대한 웹 액세스를 활성화하세요. 자세한 내용은 [네트워크 구성](#) 항목을 참조하세요.

모든 설정에 올바른 값이 포함되면 생성 명령을 실행하여 클러스터를 시작할 수 있습니다.

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



```

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

```

클러스터 진행 상황 확인:

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

또는

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

클러스터가 "clusterStatus": "CREATE\_COMPLETE" 상태에 도달하면 정상 SSH 클라이언트 설정을 사용하여 연결할 수 있습니다. Amazon EC2 인스턴스 연결에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EC2 사용](#) 설명서를 참조하십시오. 또는 다음을 통해 클러스터를 연결할 수 있습니다.

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

다음 명령을 실행하여 클러스터를 삭제합니다.

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

클러스터를 삭제한 후 CloudFormation 네트워킹 스택을 삭제하여 VPC의 네트워크 리소스를 삭제할 수 있습니다. 스택 이름은 "parallelclusternetworking-"으로 시작하고 "YYYYMMDDHHMMSS" 형식의 생성 시간을 포함합니다. [list-stacks](#) 명령을 사용하여 스택을 나열할 수 있습니다.

```

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

```

스택은 [delete-stack](#) 명령을 사용하여 삭제할 수 있습니다.

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

사용자를 위해 [pcluster configure](#) 생성하는 VPC는 CloudFormation 네트워킹 스택에서 생성되지 않습니다. 콘솔에서 또는 AWS CLI를 사용하여 해당 VPC를 수동으로 삭제할 수 있습니다.

```
$ aws --region us-east-1 Amazon EC2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

## AWS ParallelCluster UI를 사용하여 클러스터 구성 및 생성

AWS ParallelCluster UI는 콘솔과 유사한 경험을 제공하면서 AWS ParallelCluster pcluster CLI를 미러링하는 웹 기반 사용자 인터페이스입니다. 에서 UI를 설치하고 액세스할 수 있습니다. AWS ParallelCluster AWS 계정실행하면 AWS ParallelCluster UI가 사용자의 AWS 계정 Amazon API Gateway에 호스팅된 AWS ParallelCluster API의 인스턴스에 액세스합니다.

### Note

지원되는 최신 AWS ParallelCluster 버전에서 지원되는 모든 기능에 대한 UI 옵션이 UI 마법사에 없을 수 있습니다. AWS ParallelCluster 필요에 따라 구성 파일을 수동으로 편집하거나 AWS ParallelCluster CLI를 사용할 수 있습니다.

이 섹션은 AWS ParallelCluster UI를 사용하여 클러스터를 구성하고 생성하는 방법을 안내합니다.

사전 조건:

- 실행 중인 AWS ParallelCluster UI 인스턴스에 대한 액세스 자세한 정보는 [AWS ParallelCluster UI 설치하기](#)을 참조하세요.

### 클러스터 구성 및 생성

1. AWS ParallelCluster UI 클러스터 보기에서 클러스터 만들기, 단계별을 선택합니다.
2. 클러스터, 이름에 클러스터의 이름을 입력합니다.
3. 클러스터용 퍼블릭 서브넷이 있는 VPC를 선택하고 다음을 선택합니다.
4. 헤드 노드에서 SSM 세션 추가를 선택하고 다음을 선택합니다.

5. 대기열의 컴퓨팅 리소스에서 정적 노드에 1을 선택합니다.
6. 인스턴스 유형에서 선택한 기본 인스턴스 유형을 제거하고 t2.micro를 선택한 후 다음을 선택합니다.
7. 스토리지에서 다음을 선택합니다.
8. 클러스터 구성에서 클러스터 구성 YAML을 검토하고 모의 실행을 선택하여 유효성을 검사합니다.
9. 생성을 선택하여 검증된 구성을 기반으로 클러스터를 생성합니다.
10. 몇 초 후 AWS ParallelCluster UI가 자동으로 클러스터로 돌아가서 클러스터 생성 상태 및 스택 이벤트를 모니터링할 수 있습니다.
11. 세부 정보를 선택하여 버전 및 상태와 같은 클러스터 세부 정보를 확인할 수 있습니다.
12. Amazon EC2 인스턴스 및 상태 목록을 보려면 [인스턴스]를 선택합니다.
13. 클러스터 스택 이벤트를 보려면 Stack events를 선택하고 클러스터를 생성하는 CloudFormation 스택에 대한 AWS Management Console 링크를 볼 수 있습니다.
14. 세부 정보에서 클러스터 생성이 완료된 후 YAML 보기를 선택하여 클러스터 구성 YAML 파일을 보거나 다운로드할 수 있습니다.
15. 클러스터 생성이 완료되면 셸을 선택하여 클러스터 헤드 노드에 액세스합니다.

#### Note

셸을 선택하면 Amazon EC2 Systems Manager 세션을 AWS ParallelCluster 열고 세션을 추가합니다. `ssm-user`. `/etc/sudoers` 자세한 내용은 Amazon EC2 Systems Manager 사용 설명서에서 [ssm-user 계정 관리 권한 활성화 또는 해제](#)를 참조하세요.

16. 정리하려면 클러스터 보기에서 클러스터를 선택하고 작업, 클러스터 삭제를 선택합니다.

## 클러스터에 연결

를 사용하는 경우 클러스터 헤드 노드에 연결하여 작업을 실행하고 AWS ParallelCluster, 결과를 보고, 사용자를 관리하고, 클러스터 및 작업 상태를 모니터링할 수 있습니다. 다음 방법을 사용하여 클러스터 헤드 노드 인스턴스에 연결합니다.

- [키 페어](#)가 있는 ssh를 사용하여 로그인합니다. 클러스터 구성의 [HeadNode/KeyName](#)에서 프라이빗 키를 지정합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [SSH를 사용하여 Linux 인스턴스에 연결](#)을 참조하세요.
- `pcluster ssh` 명령줄 인터페이스(CLI) 명령을 사용하여 로그인합니다. 클러스터 구성 [HeadNode/KeyName](#)에서 프라이빗 키를 지정합니다. 자세한 정보는 [pcluster ssh](#)을 참조하세요.

- SSM 세션을 사용하여 클러스터 헤드 노드에 연결합니다. SSM 세션을 사용하여 연결한 클러스터 구성의 [HeadNode/AdditionalIamPolicies](#)에 AmazonSSMManagedInstanceCore 관리형 정책을 추가해야 합니다. 자세한 내용은 SSM 사용 설명서의 [SSM 세션 관리자](#)를 참조하세요.
- NICE DCV를 사용하여 클러스터 헤드 노드에 연결합니다. 자세한 정보는 [NICE DCV를 통해 헤드 노드에 연결합니다](#)을 참조하세요.
- UI를 사용할 때 AWS ParallelCluster UI에서 제공하는 Amazon EC2 Connect 명령을 사용하여 클러스터 헤드 노드에 연결할 수도 있습니다.

## 클러스터에 대한 다중 사용자 액세스

단일 클러스터에 대한 다중 사용자 액세스를 구현하고 관리하는 방법을 알아봅니다.

이 항목에서 AWS ParallelCluster 사용자는 컴퓨팅 인스턴스의 시스템 사용자를 말합니다. Amazon EC2 인스턴스를 `ec2-user` 예로 들 수 있습니다.

AWS ParallelCluster 다중 사용자 액세스 AWS ParallelCluster 지원은 현재 가능한 모든 AWS 리전 곳에서 제공됩니다. [Amazon FSx for Lustre와 아마존 Elastic File System을 비롯한 다른 AWS 서비스 시스템과도 호환됩니다](#).

[AWS Directory Service for Microsoft Active Directory](#) 또는 [Simple AD](#)를 사용하여 클러스터 액세스를 관리할 수 있습니다. 이러한 서비스의 [AWS 리전 가용성](#)을 확인하세요. 클러스터를 설정하려면 구성을 지정하십시오. [AWS ParallelCluster DirectoryService](#) AWS Directory Service 디렉토리를 여러 클러스터에 연결할 수 있습니다. 이를 통해 여러 환경에서 자격 증명을 중앙 집중식으로 관리하고 통합 로그인 경험을 제공할 수 있습니다.

를 AWS ParallelCluster 여러 사용자 AWS Directory Service 액세스에 사용하는 경우 디렉터리에 정의된 사용자 자격 증명으로 클러스터에 로그인할 수 있습니다. 이러한 자격 증명은 사용자 이름과 암호로 구성됩니다. 클러스터에 처음 로그인하면 사용자 SSH 키가 자동으로 생성됩니다. 이를 사용하여 암호 없이 로그인할 수 있습니다.

디렉터리 서비스를 배포한 후 클러스터의 사용자 또는 그룹을 생성, 삭제 및 수정할 수 있습니다. 를 사용하면 AWS Management Console 또는 Active Directory 사용자 및 컴퓨터 도구를 사용하여 이 작업을 수행할 수 있습니다. AWS Directory Service이 도구는 Active Directory에 연결된 모든 Amazon EC2 인스턴스에서 액세스할 수 있습니다. 자세한 내용은 [Active Directory 관리 도구 설치](#)를 참조하세요.

인터넷에 액세스할 수 없는 단일 AWS ParallelCluster 서브넷에서 사용하려는 경우 추가 요구 사항을 [AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서](#) 참조하십시오.

주제

- [Active Directory 생성](#)
- [AD 도메인을 사용하여 클러스터를 생성합니다.](#)
- [AD 도메인과 통합된 클러스터에 로그인합니다.](#)
- [MPI 작업 실행](#)
- [LDAP \(S\) 클러스터 구성의 예 AWS Managed Microsoft AD](#)

## Active Directory 생성

클러스터를 생성하기 전에 반드시 Active Directory(AD)를 생성해야 합니다. 클러스터의 Active Directory 유형을 선택하는 방법에 대한 자세한 내용은 AWS Directory Service 관리 가이드의 [선택 항목](#)을 참조하세요.

디렉터리가 비어 있는 경우 사용자 이름과 암호를 사용하여 사용자를 추가하세요. 자세한 내용은 [AWS Directory Service for Microsoft Active Directory](#) 또는 [Simple AD](#) 관련 설명서를 참조하세요.

### Note

AWS ParallelCluster 모든 Active Directory 사용자 디렉터리가 `/home/$user` 디렉터리에 있어야 합니다.

## AD 도메인을 사용하여 클러스터를 생성합니다.

### Warning

이 소개 섹션에서는 경량 디렉터리 액세스 프로토콜 (LDAP) 을 통해 관리형 Active Directory (AD) 서버를 설정하는 AWS ParallelCluster 방법을 설명합니다. LDAP는 안전하지 않은 프로토콜입니다. 프로덕션 시스템의 경우 다음 [LDAP \(S\) 클러스터 구성의 예 AWS Managed Microsoft AD](#) 섹션에 설명된 대로 TLS 인증서(LDAPS)를 사용하는 것이 좋습니다.

클러스터 구성 파일의 `DirectoryService` 섹션에 관련 정보를 지정하여 디렉터리와 통합되도록 클러스터를 구성하세요. 자세한 내용은 [DirectoryService](#) 구성 섹션을 참조하세요.

다음 예제를 사용하여 클러스터를 경량 디렉터리 액세스 프로토콜 (LDAP) 을 AWS Managed Microsoft AD 통한 경량 디렉터리 액세스 프로토콜 (LDAP) 과 통합할 수 있습니다.

## AWS Managed Microsoft AD OverLDAP 구성에 필요한 특정 정의:

- `ldap_auth_disable_tls_never_use_in_production` 파라미터를 [DirectoryService/AdditionalSssdConfigs](#) 아래의 True로 설정해야 합니다.
- [DirectoryService/DomainAddr](#)에 컨트롤러 호스트 이름 또는 IP 주소를 지정할 수 있습니다.
- [DirectoryService/DomainReadOnlyUser](#) 구문은 다음과 같아야 합니다.

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

## AWS Managed Microsoft AD 구성 데이터 가져오기:

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

```
{
  "DirectoryDescriptions": [
    {
      "DirectoryId": "d-abcdef01234567890",
      "Name": "corp.example.com",
      "DnsIpAddrs": [
        "203.0.113.225",
        "192.0.2.254"
      ],
      "VpcSettings": {
        "VpcId": "vpc-021345abcdef6789",
        "SubnetIds": [
          "subnet-1234567890abcdef0",
          "subnet-abcdef01234567890"
        ],
        "AvailabilityZones": [
          "region-idb",
          "region-idd"
        ]
      }
    }
  ]
}
```

## AWS Managed Microsoft AD에 대한 클러스터 구성:

```
Region: region-id
```

```

Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True

```

Simple AD에 이 구성을 사용하려면 DirectoryService 섹션에서 DomainReadOnlyUser 속성 값을 변경하세요.

```

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

```

## 고려 사항:

- LDAP만 사용하는 대신 TLS/SSL(또는 LDAPS)을 통한 LDAP를 사용하는 것이 좋습니다. TLS/SSL은 연결이 암호화되도록 합니다.
- [DirectoryService/DomainAddr](#) 속성 값은 describe-directories 출력의 DnsIpAddrs 목록에 있는 항목과 일치합니다.
- 클러스터는 [DirectoryService/DomainAddr](#)이 가리키는 동일한 가용 영역에 있는 서브넷을 사용하는 것이 좋습니다. 디렉터리 VPC에 권장되는 [사용자 지정 Dynamic Host Configuration Protocol\(DHCP\) 구성](#)을 사용하고 서브넷이 [DirectoryService/DomainAddr](#)가용 영역에 있지 않은 가용 영역 간 교차 트래픽이 발생할 수 있습니다. 다중 사용자 AD 통합 기능을 사용하기 위해 사용자 지정 DHCP 구성을 사용할 필요는 없습니다.
- [DirectoryService/DomainReadOnlyUser](#) 속성 값은 디렉터리에 만들어야 하는 사용자를 지정합니다. 이 사용자는 기본적으로 생성되지 않습니다. 이 사용자에게 디렉터리 데이터를 수정할 수 있는 권한을 부여하지 않는 것이 좋습니다.
- [DirectoryService/PasswordSecretArn](#) 속성 값은 [DirectoryService/DomainReadOnlyUser](#) 속성에 지정한 사용자의 암호가 포함된 AWS Secrets Manager 암호를 가리킵니다. 이 사용자의 암호가 변경되면 보안 암호 값을 업데이트하고 클러스터를 업데이트하세요. 새 보안 암호 값에 맞게 클러스터를 업데이트하려면 `pcluster update-compute-fleet` 명령을 사용하여 컴퓨팅 플릿을 중지해야 합니다. 클러스터가 [LoginNodes](#)를 사용하도록 구성된 경우 [LoginNodes/Pool](#)을 [LoginNodes/Pool/Count](#)를 0으로 설정한 후 클러스터를 업데이트하세요. 그리고 나서 클러스터 헤드 노드 내에서 다음 명령을 실행합니다.

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

다른 예제는 [Active Directory 통합](#)을 참조하세요.

## AD 도메인과 통합된 클러스터에 로그인합니다.

Active Directory(AD) 도메인 통합 기능을 활성화한 경우 클러스터 헤드 노드에서 암호를 통한 인증이 활성화됩니다. AD 사용자의 홈 디렉터리는 헤드 노드에 처음 로그인하거나 `sudo-user`가 헤드 노드의 AD 사용자로 처음 전환할 때 생성됩니다.

클러스터 컴퓨팅 노드에는 암호 인증이 활성화되지 않습니다. AD 사용자는 SSH 키를 사용하여 컴퓨팅 노드에 로그인해야 합니다.



기본적으로 SSH 키는 헤드 노드에 처음 SSH 로그인할 때 AD 사용자 `/${HOME}/.ssh` 디렉터리에 설정됩니다. 클러스터 구성에서 [DirectoryService/GenerateSshKeysForUsers](#) 부울 속성을 `false`로 설정하여 이 동작을 비활성화할 수 있습니다. 기본적으로 [DirectoryService/GenerateSshKeysForUsers](#)는 `true`로 설정됩니다.

AWS ParallelCluster 응용 프로그램에 클러스터 노드 간에 암호 없는 SSH가 필요한 경우 SSH 키가 사용자의 홈 디렉터리에 올바르게 설정되어 있는지 확인하십시오.

AWS Managed Microsoft AD 암호는 42일 후에 만료됩니다. 자세한 내용은 AWS Directory Service 관리 가이드의 [AWS Managed Microsoft AD를 위한 암호 정책 관리](#)를 참조하세요. 암호가 만료된 경우 클러스터 액세스를 복원하려면 암호를 재설정해야 합니다. 자세한 정보는 [사용자 암호 및 만료된 암호를 재설정하는 방법](#)을 참조하세요.

### Note

AD 통합 기능이 예상대로 작동하지 않는 경우 SSSD 로그는 문제 해결에 유용한 진단 정보를 제공할 수 있습니다. 이러한 로그는 클러스터 노드의 `/var/log/sss` 디렉터리에 있습니다. 기본적으로 클러스터의 Amazon CloudWatch 로그 그룹에도 저장됩니다. 자세한 정보는 [Active Directory와의 다중 사용자 통합 문제 해결](#)을 참조하세요.

## MPI 작업 실행

SchedMD에서 제안한 대로 Slurm을 사용한 MPI 작업을 MPI 부트스트래핑 방법으로 사용하여 부트스트랩하세요. 자세한 내용은 공식 [Slurm설명서](#) 또는 MPI 라이브러리의 공식 설명서를 참조하십시오.

예를 들어 [IntelMPI 공식 설명서](#)에서는 StarCCM 작업을 실행할 때 환경 변수 `I_MPI_HYDRA_BOOTSTRAP=slurm`을 내보내 Slurm을 프로세스 오케스트레이터로 설정해야 한다는 내용을 알 수 있습니다.

### Note

#### 알려진 문제

MPI 애플리케이션이 MPI 작업을 생성하는 메커니즘으로 SSH를 사용하는 경우 [Slurm에서 알려진 버그](#)가 발생하여 디렉터리 사용자 이름을 “nobody”로 잘못 해석할 수 있습니다. Slurm을 MPI 부트스트래핑 방법으로 사용하도록 애플리케이션을 구성하거나 문제 해결 섹션의 [사용자 이름 확인과 관련된 알려진 문제](#)에서 자세한 내용 및 가능한 해결 방법을 참조하세요.

## LDAP (S) 클러스터 구성의 예 AWS Managed Microsoft AD

AWS ParallelCluster LDAP (경량 디렉터리 액세스 프로토콜) 를 AWS Directory Service 통한 LDAP 또는 TLS/SSL을 통한 LDAP (LDAPS) 와 통합하여 다중 사용자 액세스를 지원합니다.

다음 예제에서는 LDAP(S)를 통한 AWS Managed Microsoft AD 와 통합할 클러스터 구성을 생성하는 방법을 보여 줍니다.

AWS Managed Microsoft AD 인증서 검증이 포함된 LDAPS를 통해

이 예제를 사용하여 인증서 검증을 통해 클러스터를 AWS Managed Microsoft AD over LDAPS와 통합할 수 있습니다.

인증서를 사용하는 AWS Managed Microsoft AD Over LDAPS 구성에 대한 특정 정의:

- [DirectoryService/LdapTlsReqCert](#)는 인증서 검증이 포함된 LDAPS의 경우 hard(기본값)로 설정해야 합니다.
- [DirectoryService/LdapTlsCaCert](#)는 CA(인증 기관) 인증서의 경로를 지정해야 합니다.

CA 인증서는 AD 도메인 컨트롤러용 인증서를 발급한 전체 CA 체인의 인증서를 포함하는 인증서 번들입니다.

CA 인증서와 인증서를 클러스터 노드에 설치해야 합니다.

- 컨트롤러 호스트 이름은 IP 주소가 아니라 [DirectoryService/DomainAddr](#)에 대해 지정해야 합니다.
- [DirectoryService/DomainReadOnlyUser](#) 구문은 다음과 같아야 합니다.

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

LDAPS를 통한 AD를 사용하기 위한 예제 클러스터 구성 파일:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
```

```

KeyName: pcluster
Iam:
  AdditionalIamPolicies:
    - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
CustomActions:
  OnNodeConfigured:
    Script: s3://aws-parallelcluster/scripts/pcluster-dub-msad-ldaps.post.sh
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
        CustomActions:
          OnNodeConfigured:
            Script: s3://aws-parallelcluster-pcluster/scripts/pcluster-dub-msad-
ldaps.post.sh
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://win-abcdef01234567890.corp.example.com,ldaps://win-
abcdef01234567890.corp.example.com
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsCaCert: /etc/openldap/cacerts/corp.example.com.bundleca.cer
  LdapTlsReqCert: hard

```

설치 후 스크립트에서 인증서를 추가하고 도메인 컨트롤러를 구성합니다.

```

*#!/bin/bash*
set -e

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

```

```

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

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

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

# Configure domain controllers reachability
echo "${AD_IP_1} ${AD_HOSTNAME_1}" >> /etc/hosts
echo "${AD_IP_2} ${AD_HOSTNAME_2}" >> /etc/hosts

```

다음 예와 같이 도메인에 가입된 인스턴스에서 도메인 컨트롤러 호스트 이름을 검색할 수 있습니다.

Windows 인스턴스에서

```
$ nslookup 192.0.2.254
```

```

Server: corp.example.com
Address: 192.0.2.254

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

```

Linux 인스턴스에서

```
$ nslookup 192.0.2.254
```

```

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

```

AWS Managed Microsoft AD 인증서 검증이 필요 없는 LDAPS를 통한 사용

이 예제를 사용하여 인증서 확인 없이 클러스터를 AWS Managed Microsoft AD over LDAPS와 통합할 수 있습니다.

인증서 확인을 사용하지 않는 AWS Managed Microsoft AD over LDAPS 컨피그레이션에 대한 특정 정의:

- [DirectoryService/LdapTlsReqCert](#)를 never로 설정해야 합니다.
- [DirectoryService/DomainAddr](#)에 대해 컨트롤러 호스트 이름 또는 IP 주소를 지정할 수 있습니다.
- [DirectoryService/DomainReadOnlyUser](#) 구문은 다음과 같아야 합니다.

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

인증서 확인 없이 AWS Managed Microsoft AD OverLDAPS를 사용하기 위한 예제 클러스터 구성 파일:

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://203.0.113.225,ldaps://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

```
LdapTlsReqCert: never
```

## 모범 사례

### 모범 사례: 헤드 노드 인스턴스 유형 선택

헤드 노드가 작업을 실행하지는 않지만 헤드 노드의 기능과 크기 조정은 클러스터의 전체 성능에 매우 중요합니다. 헤드 노드에 사용할 인스턴스 유형을 선택할 때는 다음 특성을 고려하세요.

**클러스터 크기:** 헤드 노드는 클러스터의 규모 조정 로직을 관리하고 새 노드를 스케줄러에 연결하는 역할을 합니다. 노드 수가 많은 클러스터를 스케일 업하거나 스케일 다운하려면 헤드 노드에 추가 컴퓨팅 파워를 제공하세요.

**공유 파일 시스템:** 공유 파일 시스템을 사용할 때는 워크플로를 처리하기에 충분한 네트워크 대역폭과 Amazon EBS 대역폭이 충분한 인스턴스 유형을 선택하세요. 헤드 노드가 클러스터에 충분한 NFS 서버 디렉터리를 노출하고 컴퓨팅 노드와 헤드 노드 간에 공유해야 하는 아티팩트를 처리할 수 있는지 확인하세요.

### 모범 사례: 네트워크 성능

고성능 컴퓨팅(HPC) 애플리케이션에는 네트워크 성능이 중요합니다. 신뢰성 있는 네트워크 성능이 없으면 이러한 애플리케이션이 예상대로 작동할 수 없습니다. 네트워크 성능을 최적화하려면 다음 모범 사례를 고려하세요.

- **배치 그룹:** Slurm을 사용하는 경우 클러스터 배치 그룹을 사용하도록 각 Slurm 대기열을 구성하는 것이 좋습니다. 클러스터의 배치 그룹은 단일 가용 영역 내에 있는 인스턴스의 논리적 그룹입니다. 자세한 내용은 Amazon EC2 사용 설명서의 [배치 그룹](#)을 참조하십시오. 대기열의 [Networking](#) 섹션에서 [PlacementGroup](#)을 지정할 수 있습니다. 각 컴퓨팅 리소스는 대기열의 배치 그룹에 할당됩니다. 컴퓨팅 리소스의 [Networking](#) 섹션에서 [PlacementGroup](#)를 지정하면 해당 특정 컴퓨팅 리소스가 해당 배치 그룹에 할당됩니다. 컴퓨팅 리소스 배치 그룹 사양은 컴퓨팅 리소스의 대기열 사양보다 우선합니다. 자세한 내용은 [SlurmQueues/Networking/PlacementGroup](#) 및 [SlurmQueues/ComputeResources/Networking/PlacementGroup](#) 섹션을 참조하세요.

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

대신 배치 그룹을 AWS ParallelCluster 만들어 둘 수도 있습니다.

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

AWS ParallelCluster 버전 3.3.0부터 배치 그룹 생성 및 관리가 수정되었습니다. 대기열에 name 또는 Id 없이 배치 그룹을 활성화하도록 지정하면 대기열 전체에 대해 하나의 관리형 그룹이 할당되는 대신 각 컴퓨팅 리소스에 고유한 관리형 배치 그룹이 할당됩니다. 이렇게 하면 용량 부족 오류를 줄이는 데 도움이 됩니다. 전체 대기열에 하나의 배치 그룹이 필요한 경우 이름이 지정된 배치 그룹을 사용할 수 있습니다.

[SlurmQueues/Networking/PlacementGroup/Name](#)이 [SlurmQueues/Networking/PlacementGroup/Id](#)의 기본 대안으로 추가되었습니다.

자세한 정보는 [Networking](#)을 참조하세요.

- 향상된 네트워킹: 향상된 네트워킹을 지원하는 인스턴스 유형을 선택하는 것이 좋습니다. 이 권장 사항은 모든 [현재 세대 인스턴스](#)에 적용됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Linux에서의 향상된 네트워킹](#)을 참조하십시오.
- Elastic Fabric Adapter: 높은 수준의 규모 조정 가능한 인스턴스 간 통신을 지원하려면 네트워크에 맞는 EFA 네트워크 인터페이스를 선택하는 것이 좋습니다. EFA의 맞춤형 운영 체제(OS) 바이패스 하드웨어는 AWS 클라우드의 온디맨드 탄력성과 유연성을 통해 인스턴스 간 통신을 향상합니다. 각 Slurm 대기열 [ComputeResource](#)이 [Efa](#)를 사용하도록 구성할 수 있습니다. EFA를 사용하는 방법에 대한 자세한 내용은 [AWS ParallelCluster](#)참조하십시오. [Elastic Fabric Adapter](#)

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

EFA에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용자 설명서의 [Elastic Fabric Adapter](#)를 참조하세요.

- 인스턴스 대역폭: 대역폭은 인스턴스 크기에 따라 조정됩니다. 다양한 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EBS 최적화 인스턴스](#) 및 [Amazon EBS 볼륨 유형](#)을 참조하십시오.

## 모범 사례: 예산 알림

에서 리소스 비용을 관리하려면 AWS ParallelCluster 예산을 생성하는 AWS Budgets 작업을 사용하는 것이 좋습니다. 선택한 AWS 리소스에 대해 정의된 예산 임계값 알림을 생성할 수도 있습니다. 자세한 내용은 AWS Budgets 사용 설명서의 [예산 작업 구성](#)을 참조하세요. 마찬가지로 CloudWatch Amazon을 사용하여 결제 경보를 생성할 수도 있습니다. 자세한 내용은 [예산 AWS 요금을 모니터링하기 위한 결제 경보 생성](#)을 참조하세요.

## 모범 사례: 클러스터를 새 AWS ParallelCluster 마이너 또는 패치 버전으로 이동

현재 각 AWS ParallelCluster 마이너 버전은 CLI와 함께 독립적입니다. pcluster 클러스터를 새 마이너 또는 패치 버전으로 옮기려면 새 버전의 CLI를 사용하여 클러스터를 다시 생성해야 합니다.

클러스터를 새 마이너 또는 패치 버전으로 이동하는 프로세스를 최적화하려면 다음을 수행하는 것이 좋습니다.

- Amazon EFS 및 FSx for Lustre와 같이 클러스터 외부에서 생성된 외부 볼륨에 개인 데이터를 저장합니다. 이렇게 하면 나중에 한 클러스터에서 다른 클러스터로 데이터를 쉽게 이동할 수 있습니다.
- 다음 유형을 사용하여 공유 스토리지 시스템을 생성합니다. OR를 사용하여 이러한 시스템을 만들 수 있습니다. AWS CLI AWS Management Console
  - [SharedStorage](#) / [EbsSettings](#) / [VolumeId](#)
  - [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)
  - [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)

클러스터 구성의 파일 시스템 또는 볼륨을 기존 파일 시스템 또는 볼륨으로 정의합니다. 이렇게 하면 클러스터를 삭제해도 파일이 보존되고 새 클러스터에 연결할 수 있습니다.

Amazon EFS 또는 FSx for Lustre 파일 시스템을 사용하는 것이 좋습니다. 두 시스템 모두 동시에 여러 클러스터에 연결할 수 있습니다. 또한 기존 클러스터를 삭제하기 전에 두 시스템 중 하나를 새 클러스터에 연결할 수 있습니다.

- 사용자 지정 AMI 대신 [사용자 지정 부트스트랩 작업](#)을 사용하여 인스턴스를 사용자 지정합니다. 대신 사용자 지정 AMI를 사용하는 경우 새 버전이 릴리스될 때마다 해당 AMI를 삭제하고 다시 생성해야 합니다.
- 다음 순서대로 이전 권장 사항을 적용하는 것이 좋습니다.
  1. 기존 파일 시스템 정의를 사용하도록 기존 클러스터 구성을 업데이트하세요.



2. pcluster 버전을 확인하고 필요한 경우 업데이트하세요.
3. 새 클러스터를 만들고 테스트합니다. 새 클러스터를 테스트할 때는 다음을 확인하세요.
  - 새 클러스터에서 데이터를 사용할 수 있는지 확인합니다.
  - 애플리케이션이 새 클러스터에서 작동하는지 확인합니다.
4. 새 클러스터를 완전히 테스트하고 운영하여 기존 클러스터가 더 이상 필요하지 않은 후에는 삭제하세요.

## AWS ParallelCluster 2.x에서 3.x로 이동

### 사용자 지정 부트스트랩 작업

AWS ParallelCluster pre\_install3에서는 [미션/섹션의 \(버전 2\)](#) 및 OnNodeStart (AWS ParallelCluster AWS ParallelCluster 버전 2) 파라미터를 사용하여 헤드 노드와 컴퓨팅 노드에 대해 서로 다른 사용자 지정 부트스트랩 작업 스크립트를 지정할 수 있습니다. OnNodeConfigured post\_install [HeadNodeSchedulingSlurmQueues](#) 자세한 정보는 [사용자 지정 부트스트랩 작업을 참조하세요](#).

AWS ParallelCluster 2용으로 개발된 사용자 지정 부트스트랩 액션 스크립트는 3에서 AWS ParallelCluster 사용할 수 있도록 조정해야 합니다.

- 헤드 노드와 컴퓨팅 노드를 구분하기 위해 /etc/parallelcluster/cfnconfig 및 cfn\_node\_type을 사용하지 않는 것이 좋습니다. 대신, [HeadNode](#) 및 [Scheduling/SlurmQueues](#)에 서로 다른 두 스크립트를 지정하는 것이 좋습니다.
- 부트스트랩 액션 스크립트에서 사용하기 /etc/parallelcluster/cfnconfig 위해 계속 로드하려면 의 cfn\_node\_type 값이 ""에서MasterServer" "로 변경되었다는 점에 유의하십시오 (참조:[포괄적인 언어](#)). HeadNode
- AWS ParallelCluster 2에서 부트스트랩 작업 스크립트의 첫 번째 입력 인수는 스크립트의 S3 URL이 있으며 예약되었습니다. AWS ParallelCluster 3에서는 구성에 구성된 인수만 스크립트에 전달됩니다.

#### Warning

/etc/parallelcluster/cfnconfig 파일을 통해 제공된 내부 변수를 사용하는 것은 공식적으로 지원되지 않습니다. 이 파일은 향후 릴리스에서 제거될 수 있습니다.

## AWS ParallelCluster 2.x와 3.x는 다른 구성 파일 구문을 사용합니다.

AWS ParallelCluster 3.x 구성에서는 YAML 구문을 사용합니다. 전체 참조는 [구성 파일](#)에서 찾을 수 있습니다.

YAML 파일 형식을 요구하는 것 외에도 여러 구성 섹션, 설정 및 파라미터 값이 AWS ParallelCluster 3.x에서 업데이트되었습니다. 이 섹션에서는 구성의 주요 변경 사항과 각 버전의 차이점을 보여주는 side-by-side 예제와 함께 AWS ParallelCluster 구성에 대한 주요 변경 사항을 살펴봅니다. AWS ParallelCluster

하이퍼스레딩을 활성화하거나 비활성화한 다중 스케줄러 대기열 구성의 예

### AWS ParallelCluster 2:

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

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

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

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

### AWS ParallelCluster 3:

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: ht-enabled
      Networking:
        SubnetIds:
          - compute_subnet_id
```

```

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

```

## 새로운 FSx for Lustre 파일 시스템 구성의 예

### AWS ParallelCluster 2:

```

[cluster default]
fsx_settings = fsx
...

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

```

### AWS ParallelCluster 3:

```

...
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      ImportedFileChunkSize: 1024
      ImportPath: s3://bucket
      ExportPath: s3://bucket/export_dir

```

```
WeeklyMaintenanceStartTime: "3:02:30"
DeploymentType: PERSISTENT_1
DataCompressionType: LZ4
```

기존 FSx for Lustre 파일 시스템을 탑재하는 클러스터 구성의 예

AWS ParallelCluster 2:

```
[cluster default]
fsx_settings = fsx
...

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

AWS ParallelCluster 3:

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

인텔 HPC 플랫폼 사양 소프트웨어 스택이 포함된 클러스터의 예

AWS ParallelCluster 2:

```
[cluster default]
enable_intel_hpc_platform = true
...
```

AWS ParallelCluster 3:

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

**참고:**

- Intel HPC 플랫폼 사양 소프트웨어의 설치에는 해당 [Intel 최종 사용자 사용권 계약](#)의 약관이 적용됩니다.

사용자 지정 IAM 구성의 예: 인스턴스 프로파일, 인스턴스 역할, 인스턴스에 대한 추가 정책, 클러스터와 관련된 Lambda 함수의 역할 등

**AWS ParallelCluster 2:**

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

**AWS ParallelCluster 3:**

```
...
Iam:
  Roles:
    CustomLambdaResources: lambda_iam_role
HeadNode:
  ...
  Iam:
    InstanceRole: ec2_iam_role
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ...
      Iam:
        InstanceProfile: iam_instance_profile
    - Name: queue2
      ...
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
          - Policy: arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
```

**참고:**

- AWS ParallelCluster 2의 경우 IAM 설정은 클러스터의 모든 인스턴스에 적용되며 클러스터와 함께 `ec2_iam_role` 사용할 `additional_iam_policies` 수 없습니다.
- AWS ParallelCluster 3의 경우 헤드 노드와 컴퓨팅 노드에 대해 서로 다른 IAM 설정을 지정할 수 있으며 컴퓨팅 대기열마다 다른 IAM 설정을 지정할 수도 있습니다.
- AWS ParallelCluster 3의 경우 IAM 역할 대신 IAM 인스턴스 프로필을 사용할 수 있습니다. `InstanceProfileInstanceRole` 또는 함께 구성할 `AdditionalIamPolicies` 수 없습니다.

## 사용자 지정 부트스트랩 작업의 예

### AWS ParallelCluster 2:

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

### AWS ParallelCluster 3:

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

```

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

```

S3 버킷 리소스에 대한 읽기 및 쓰기 액세스 권한이 있는 클러스터의 예

AWS ParallelCluster 2:

```

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

```

AWS ParallelCluster 3:

```

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

```

```

EnableWriteAccess: False
- BucketName: bucket_name
  KeyName: read_and_write/
  EnableWriteAccess: True

```

## 포괄적인 언어

AWS ParallelCluster 3은 AWS ParallelCluster 2에서 “마스터”가 사용된 곳에서 “헤드 노드”라는 단어를 사용합니다. 다음 내용이 포함됩니다:

- AWS Batch 작업 환경에서 내보낸 변수가 에서 MASTER\_IP 로 변경되었습니다다PCLUSTER\_HEAD\_NODE\_IP.
- 모든 AWS CloudFormation 출력이 에서 Master\* 로 변경되었습니다다HeadNode\*.
- 모두 NodeType 및 태그가 에서 Master 로 변경되었습니다다HeadNode.

## 스케줄러 지원

AWS ParallelCluster 3.x는 썬 오브 그리드 엔진 (SGE) 및 토크 스케줄러를 지원하지 않습니다.

awsbhosts,, awskill awsbout awsbqueuesawsbstat, 및 AWS Batch 명령은 별도의 aws-parallelcluster-awsbatch-cli PyPI 패키지로 awbsub 배포됩니다. 헤드 노드에는 AWS ParallelCluster 에 의해 이 패키지가 설치됩니다. 클러스터의 헤드 노드에서 이러한 AWS Batch 명령을 계속 사용할 수 있습니다. 그러나 헤드 노드가 아닌 다른 위치에서 AWS Batch 명령을 사용하려면 먼저 aws-parallelcluster-awsbatch-cli PyPI 패키지를 설치해야 합니다.

## AWS ParallelCluster CLI

AWS ParallelCluster 명령줄 인터페이스 (CLI) 가 변경되었습니다. 새 구문은 [AWS ParallelCluster CLI 명령](#)에 설명되어 있습니다. CLI의 출력은 [JSON](#) 문자열 형식입니다.

### 새 클러스터 구성

이 pcluster configure 명령에는 AWS ParallelCluster 2에 비해 AWS ParallelCluster 3에는 다른 파라미터가 포함되어 있습니다. 자세한 정보는 [pcluster configure](#)을 참조하세요.

또한 구성 파일 구문이 AWS ParallelCluster 2에서 변경되었다는 점도 참고하십시오. 클러스터 구성 설정에 대한 전체 참조는 [클러스터 구성 파일](#)를 참조하세요.

### 새 클러스터 생성



AWS ParallelCluster 2의 `pcluster create` 명령이 [pcluster create-cluster](#) 명령으로 대체되었습니다.

참고로 AWS ParallelCluster 2.x에서 `-nw` 옵션이 없는 경우 기본 동작은 클러스터 생성 이벤트를 기다리는 것이고 AWS ParallelCluster 3.x 명령은 즉시 반환됩니다. [pcluster describe-cluster](#)를 사용하여 클러스터 생성 진행 상황을 모니터링할 수 있습니다.

AWS ParallelCluster 3 구성 파일에는 단일 클러스터 정의가 포함되므로 `-t` 매개 변수가 더 이상 필요하지 않습니다.

다음은 구성 파일의 예입니다.

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

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

## 클러스터 나열

`pcluster list` AWS ParallelCluster 2.x 명령은 명령으로 [pcluster list-clusters](#) 대체해야 합니다.

참고: 2.x 버전으로 생성된 클러스터를 나열하려면 AWS ParallelCluster v2 CLI가 필요합니다. AWS ParallelCluster 가상 환경을 사용하는 AWS ParallelCluster 의 여러 버전을 설치하는 방법은 [가상 환경에 설치 AWS ParallelCluster \(권장\)](#)을 참조하세요.

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

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

## 클러스터 시작 및 중지

`pcluster start` 및 `pcluster stop` AWS ParallelCluster 2.x 명령을 명령으로 바꾸어야 합니다.

[pcluster update-compute-fleet](#)

컴퓨팅 플릿 시작:

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

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

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

컴퓨팅 플릿 중지:

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

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

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

클러스터에 연결

3.x에서는 `pcluster ssh` AWS ParallelCluster AWS ParallelCluster 2.x 명령의 매개변수 이름이 다릅니다. [pcluster ssh](#) 섹션을 참조하십시오.

클러스터에 연결:

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

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

## IMDS 구성 업데이트

버전 3.0.0부터 헤드 노드의 IMDS (및 인스턴스 프로파일 자격 증명)에 대한 액세스를 기본적으로 일부 수퍼유저로 제한하는 지원이 AWS ParallelCluster 도입되었습니다. 자세한 내용은 [Imds 속성](#)(를) 참조하세요.

## AWS ParallelCluster 지원 리전

AWS ParallelCluster 버전 3은 다음 AWS 리전에서 사용 가능합니다.

리전 이름	리전
미국 동부(오하이오)	us-east-2
미국 동부(버지니아 북부)	us-east-1
미국 서부(캘리포니아 북부)	us-west-1
미국 서부(오레곤)	us-west-2
아프리카(케이프타운)	af-south-1
아시아 태평양(홍콩)	ap-east-1

리전 이름	리전
아시아 태평양(뭄바이)	ap-south-1
아시아 태평양(서울)	ap-northeast-2
아시아 태평양(싱가포르)	ap-southeast-1
아시아 태평양(시드니)	ap-southeast-2
아시아 태평양(도쿄)	ap-northeast-1
캐나다(중부)	ca-central-1
중국(베이징)	cn-north-1
중국(닝샤)	cn-northwest-1
유럽(프랑크푸르트)	eu-central-1
유럽(아일랜드)	eu-west-1
유럽(런던)	eu-west-2
유럽(밀라노)	eu-south-1
유럽(파리)	eu-west-3
유럽(스톡홀름)	eu-north-1
중동(바레인)	me-south-1
남아메리카(상파울루)	sa-east-1
AWS GovCloud (미국 동부)	us-gov-east-1
AWS GovCloud (미국 서부)	us-gov-west-1
이스라엘(텔아비브)	il-central-1

# 사용 AWS ParallelCluster

## 주제

- [AWS ParallelCluster UI](#)
- [AWS Lambda 에서 VPC 구성 AWS ParallelCluster](#)
- [AWS Identity and Access Management 권한 입력 AWS ParallelCluster](#)
- [네트워크 구성](#)
- [로그인 노드](#)
- [사용자 지정 부트스트랩 작업](#)
- [Amazon S3 작업](#)
- [스팟 인스턴스 작업](#)
- [에서 지원하는 스케줄러 AWS ParallelCluster](#)
- [공유 스토리지](#)
- [AWS ParallelCluster 리소스 및 태깅](#)
- [모니터링 AWS ParallelCluster 및 로그](#)
- [AWS CloudFormation 사용자 지정 리소스](#)
- [Elastic Fabric Adapter](#)
- [인텔 활성화 MPI](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster 테라폼용](#)
- [NICE DCV를 통해 헤드 노드에 연결합니다.](#)
- [pcluster update-cluster 사용하기](#)
- [AWS ParallelCluster AMI 사용자 지정](#)
- [온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작](#)
- [용량 블록 \(CB\) 으로 인스턴스 시작](#)
- [AMI 패치 적용 및 Amazon EC2 인스턴스 교체](#)
- [운영 체제](#)

# AWS ParallelCluster UI

AWS ParallelCluster UI는 클러스터를 생성, 모니터링 및 관리하기 위한 대시보드 역할을 하는 웹 기반 사용자 인터페이스입니다. AWS 계정에 AWS ParallelCluster UI를 설치하고 액세스할 수 있습니다. AWS ParallelCluster UI는 AWS ParallelCluster 버전 3.5.0에 추가되었습니다.

AWS ParallelCluster UI를 설치하고 시작하려면 [AWS ParallelCluster UI 설치하기](#) 및 [AWS ParallelCluster UI를 사용하여 클러스터 구성 및 생성을 참조하세요](#).

The screenshot displays the AWS ParallelCluster UI interface. At the top, the header shows 'AWS ParallelCluster' and user information. The main content area is titled 'Clusters (2)' and includes a search bar and a table of clusters. The table has columns for Name, Status, and Version. The first cluster, 'hpc-cluster-1', is selected and highlighted, showing a status of 'CREATE COMPLETE' and version '3.5.0'. The second cluster, 'hpc-cluster-2', has a status of 'DELETE IN PROGRESS' and version '3.5.0'. Below the table, the details for 'hpc-cluster-1' are expanded, showing various tabs like 'Details', 'Instances', 'Storage', 'Job scheduling', and 'Stack events'. The 'Properties' section includes 'Cluster configuration', 'SSH command', 'EC2 Instance Connect', 'Cluster status', 'Compute fleet status', 'Version', 'Region', 'Created time', and 'Latest update time'.

AWS ParallelCluster UI는 다음 기능을 지원합니다.

- 다음이 표시됩니다.
  - AWS ParallelCluster로 사용자의 AWS 계정에 생성한 클러스터 목록
  - 목록에 있는 클러스터의 사용 가능한 상태 및 세부 정보
  - 모니터링에 사용할 수 있는 CloudFormation 스택 이벤트 및 AWS ParallelCluster 로그입니다.

- 클러스터에서 실행 중인 작업의 상태
- 클러스터를 빌드하는 데 사용할 수 있는 사용자 지정 이미지 목록
- UI가 클러스터를 생성하는 데 사용하는 공식 이미지 목록
- AWS ParallelCluster UI에 액세스할 수 있는 사용자 목록 사용자를 제거하고 추가할 수 있습니다.
- 클러스터를 생성 및 편집(업데이트)하고 추가, 편집 또는 제거할 지원되는 클러스터 특성을 선택하는 방법에 대한 단계별 지침을 제공합니다. 액세스할 수 없는 입력 필드는 편집 중인 클러스터 구성에서 변경할 수 없습니다. 클러스터를 배포하기 전에 클러스터 구성의 모의 실행 검증을 수행할 수 있습니다.
- 클러스터 보기에서 헤드 노드에 액세스할 수 있는 다이렉트 웹 링크를 제공합니다. 단계별 지침에서 SSM 세션 추가를 선택하여 헤드 노드에 직접 웹 액세스 및 SSM 관리형 인스턴스 코어 정책을 추가합니다.

AWS ParallelCluster UI를 사용하여 클러스터를 생성하고 관리할 때는 다음 사항을 고려합니다.

- 클러스터를 생성 및 편집하거나 AWS ParallelCluster UI를 만드는 데 사용한 것과 동일한 AWS ParallelCluster 버전으로만 이미지를 빌드할 수 있습니다. 이전 버전의 클러스터 또는 이미지는 보기만 가능합니다. 여러 버전의 클러스터와 이미지를 관리하는 경우 각 버전을 지원하는 AWS ParallelCluster UI 인스턴스를 만드는 것이 좋습니다.
- AWS ParallelCluster UI는 `pcluster` CLI 기능을 미러링하도록 설계되었습니다. 일부 다른 점이 있습니다. 단계별 지침을 준수하면 지원되는 모든 특성을 사용하는 것입니다. 배포하기 전에 클러스터 또는 이미지 구성을 수동으로 편집할 수 있습니다. 이렇게 하는 경우 모의 실행을 선택하여 구성을 검증하여 편집이 완전히 지원되는지 확인하는 것이 좋습니다.

#### Note

AWS ParallelCluster UI는 AWS Batch를 지원하지 않습니다.

## AWS Lambda 에서 VPC 구성 AWS ParallelCluster

AWS ParallelCluster 클러스터의 수명 주기 동안 작업을 수행하는 AWS Lambda 데 사용합니다. [AWS Lambda 함수는 항상 Lambda 서비스가 소유한 VPC 내에서 실행됩니다.](#) 이 Lambda 함수는 프라이빗 리소스에 액세스하기 위해 Virtual Private Cloud(VPC)의 프라이빗 서브넷에 연결할 수 있습니다.

**Note**

Lambda 함수는 전용 인스턴스 테넌시를 사용하여 VPC에 직접 연결할 수 없습니다. 전용 VPC의 리소스에 연결하려면 전용 VPC에 연결할 수 있는 기본 테넌시를 사용하여 전용 VPC를 두 번째 VPC에 피어로 연결합니다.

자세한 내용은 Linux [인스턴스용 Amazon EC2 사용 설명서의 전용 인스턴스](#) 및 [Lambda 함수를 전용 VPC에 연결하려면 어떻게 해야 하나요?](#) 를 참조하십시오. 지식 센터에서 AWS

에서 AWS ParallelCluster 생성한 Lambda 함수를 프라이빗 VPC에 연결할 수 있습니다. 이러한 Lambda 함수는 AWS 서비스에 액세스해야 합니다. 다음 방법을 사용하여 인터넷 또는 VPC 엔드포인트를 통해 액세스를 제공할 수 있습니다.

- 인터넷 액세스

인터넷에 액세스하려면 Lambda 함수에 네트워크 주소 변환 (NAT) 이 필요합니다. AWS 서비스 프라이빗 서브넷의 아웃바운드 트래픽을 퍼블릭 서브넷의 [NAT 게이트웨이](#)로 라우팅합니다.

- VPC 엔드포인트

여러 AWS 서비스가 [VPC 엔드포인트](#)를 제공합니다. VPC 엔드포인트를 사용하여 인터넷에 액세스할 수 없는 VPC에서 연결할 AWS 서비스 수 있습니다. AWS ParallelCluster [VPC 엔드포인트 목록을 보려면 네트워킹을 참조하십시오.](#)

**Note**

서브넷과 보안 그룹의 모든 조합은 다음 방법 중 하나를 AWS 서비스 사용하여 액세스할 수 있어야 합니다. 서브넷과 보안 그룹이 동일한 VPC에 있어야 합니다.

자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC 엔드포인트](#)와 AWS Lambda 개발자 안내서의 [VPC 연결 기능에 대한 인터넷 및 서비스 액세스](#)를 참조하세요.

Lambda 함수 및 VPC의 사용을 구성하려면 클러스터의 경우 [DeploymentSettings/LambdaFunctionsVpcConfig](#)를, 이미지는 [DeploymentSettings/LambdaFunctionsVpcConfig](#)를 참조하세요.



# AWS Identity and Access Management 권한 입력 AWS ParallelCluster

AWS ParallelCluster 클러스터를 생성하고 관리할 때 IAM 권한을 사용하여 리소스에 대한 액세스를 제어합니다.

AWS 계정에서 클러스터를 생성하고 관리하려면 다음 두 수준의 권한이 AWS ParallelCluster 필요합니다.

- `pcluster` 사용자가 클러스터를 생성하고 관리하기 위한 `pcluster` CLI 명령을 간접 호출하는 데 필요한 권한입니다.
- 클러스터 리소스가 클러스터 작업을 수행하는 데 필요한 권한입니다.

AWS ParallelCluster [Amazon EC2 인스턴스 프로파일 및 역할을](#) 사용하여 클러스터 리소스 권한을 제공합니다. 클러스터 리소스 권한을 관리하려면 IAM 리소스에 대한 AWS ParallelCluster 권한도 필요합니다. 자세한 정보는 [AWS ParallelCluster IAM 리소스 관리를 위한 사용자 예제 정책](#)을 참조하세요.

`pcluster` 사용자는 `pcluster` CLI를 사용하여 클러스터와 해당 리소스를 생성하고 관리하려면 IAM 권한이 필요합니다. 이러한 권한은 사용자 또는 역할에 추가할 수 있는 IAM 정책에 포함됩니다. IAM 역할에 대한 자세한 정보는 AWS Identity and Access Management 사용 설명서의 [사용자 역할 생성](#)을 참조하세요.

[AWS ParallelCluster IAM 권한을 관리하기 위한 구성 파라미터](#)도 사용할 수 있습니다.

다음 섹션에는 필수 권한이 예제와 함께 포함되어 있습니다.

예제 정책을 사용하려면 `<REGION>`, `<AWS ACCOUNT ID>` 및 유사한 문자열을 적절한 값으로 바꾸세요.

다음 정책 예시에는 리소스의 Amazon 리소스 이름(ARN)이 포함되어 있습니다. AWS GovCloud (US) 또는 AWS 중국 파티션에서 작업하는 경우 ARN을 변경해야 합니다. 특히 파티션의 경우 “arn:aws”에서 “arn:aws-us-gov”로, 중국 AWS GovCloud (US) 파티션의 경우 “arn:aws-cn”으로 변경해야 합니다. AWS 자세한 내용은 AWS GovCloud (US) 사용 설명서의 AWS GovCloud (US) [지역별 Amazon 리소스 이름 \(ARN\)](#) 및 중국 내 [AWS 서비스 시작하기에서 중국 내 서비스용 ARN](#)을 참조하십시오. AWS

의 [AWS ParallelCluster 설명서에서](#) 예제 정책의 변경 사항을 추적할 수 있습니다. GitHub

주제

- [AWS ParallelCluster Amazon EC2 인스턴스 역할](#)

- [AWS ParallelCluster 사용자 정책 예시 pcluster](#)
- [AWS ParallelCluster IAM 리소스 관리를 위한 사용자 예제 정책](#)
- [AWS ParallelCluster IAM 권한을 관리하기 위한 구성 파라미터](#)

## AWS ParallelCluster Amazon EC2 인스턴스 역할

기본 구성 설정으로 클러스터를 생성하면 Amazon EC2 [인스턴스 프로필을 AWS ParallelCluster](#) 사용하여 클러스터와 해당 리소스를 생성하고 관리하는 데 필요한 권한을 제공하는 기본 클러스터 Amazon [EC2 인스턴스](#) 역할을 자동으로 생성합니다.

### 기본 AWS ParallelCluster 인스턴스 역할을 사용하는 것에 대한 대안

기본 AWS ParallelCluster 인스턴스 역할 대신 InstanceRole 클러스터 구성 설정을 사용하여 EC2용 기존 IAM 역할을 지정할 수 있습니다. 자세한 정보는 [AWS ParallelCluster IAM 권한을 관리하기 위한 구성 파라미터](#)를 참조하세요. 일반적으로 기존 IAM 역할을 지정하여 EC2에 부여된 권한을 완전히 제어합니다.

기본 인스턴스 역할에 추가 정책을 추가하려는 경우, [InstanceProfile 또는 InstanceRole](#) 설정을 사용하는 대신 [AdditionalIamPolicies](#) 설정을 사용하여 추가 IAM 정책을 전달하는 것이 좋습니다. 클러스터를 업데이트할 때 AdditionalIamPolicies를 업데이트할 수 있지만 클러스터를 업데이트할 때는 InstanceRole을 업데이트할 수 없습니다.

## AWS ParallelCluster 사용자 정책 예시 **pcluster**

다음 예는 pcluster CLI를 사용하여 리소스를 생성하고 관리하는 데 필요한 사용자 AWS ParallelCluster 정책과 해당 리소스를 보여줍니다. 사용자 또는 역할에 정책을 연결할 수 있습니다.

### 주제

- [기본 AWS ParallelCluster pcluster 사용자 정책](#)
- [AWS Batch 스케줄러 사용 시 추가 AWS ParallelCluster pcluster 사용자 정책](#)
- [Amazon FSx for Lustre를 사용할 때의 추가 AWS ParallelCluster pcluster 사용자 정책](#)
- [AWS ParallelCluster 이미지 빌드 pcluster 사용자 정책](#)

### 기본 AWS ParallelCluster **pcluster** 사용자 정책

다음 정책은 AWS ParallelCluster pcluster 명령을 실행하는 데 필요한 권한을 보여줍니다.

정책에 나열된 마지막 작업은 클러스터 구성에 지정된 모든 암호의 검증을 제공하기 위해 포함됩니다. 예를 들어, AWS Secrets Manager 암호는 [DirectoryService](#) 통합을 구성하는 데 사용됩니다. 이 경우 클러스터는 [PasswordSecretArn](#)에 유효한 보안 암호가 있는 경우에만 생성됩니다. 이 작업을 생략하면 보안 암호 검증이 생략됩니다. 보안 태세를 개선하려면 클러스터 구성에 지정된 비밀만 추가하여 이 정책 설명의 범위를 좁히는 것이 좋습니다.

### Note

기존 Amazon EFS 파일 시스템이 클러스터에서 사용되는 유일한 파일 시스템인 경우, Amazon EFS 정책 설명 예제를 클러스터 구성 파일의 [SharedStorage 섹션](#)에서 참조하는 특정 파일 시스템으로 범위를 좁힐 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Read"
    },
    {
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateFleet",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateNetworkInterface",
        "ec2:CreatePlacementGroup",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateTags",
        "ec2>DeleteTags",
        "ec2:CreateVolume",

```

```

        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteNetworkInterface",
        "ec2:DeletePlacementGroup",
        "ec2:DeleteSecurityGroup",
        "ec2:DeleteVolume",
        "ec2:DisassociateAddress",
        "ec2:ModifyLaunchTemplate",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifyVolume",
        "ec2:ModifyVolumeAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RunInstances",
        "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Write"
},
{
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTagsOfResource",
        "dynamodb:CreateTable",
        "dynamodb>DeleteTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:Query",
        "dynamodb:TagResource"
    ],
    "Resource": "arn:aws:dynamodb:*:<AWS ACCOUNT ID>:table/parallelcluster-*",
    "Effect": "Allow",
    "Sid": "DynamoDB"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:CreateHostedZone",
        "route53>DeleteHostedZone",
        "route53:GetChange",
        "route53:GetHostedZone",

```

```

        "route53:ListResourceRecordSets",
        "route53:ListQueryLoggingConfigs"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
},
{
    "Action": [
        "cloudformation:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudFormation"
},
{
    "Action": [
        "cloudwatch:PutDashboard",
        "cloudwatch:ListDashboards",
        "cloudwatch>DeleteDashboards",
        "cloudwatch:GetDashboard",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch>DeleteAlarms",
        "cloudwatch:DescribeAlarms"
        "cloudwatch:PutCompositeAlarm"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatch"
},
{
    "Action": [
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetPolicy",
        "iam:SimulatePrincipalPolicy",
        "iam:GetInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/*",
        "arn:aws:iam::aws:policy/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*"
    ]
},

```

```
    "Effect": "Allow",
    "Sid": "IamRead"
  },
  {
    "Action": [
      "iam:CreateInstanceProfile",
      "iam>DeleteInstanceProfile",
      "iam:AddRoleToInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamInstanceProfile"
  },
  {
    "Condition": {
      "StringEqualsIfExists": {
        "iam:PassedToService": [
          "lambda.amazonaws.com",
          "ec2.amazonaws.com",
          "spotfleet.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda>DeleteFunction",
      "lambda:GetFunctionConfiguration",
      "lambda:GetFunction",
      "lambda:InvokeFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
```

```

        "lambda:UpdateFunctionConfiguration",
        "lambda:TagResource",
        "lambda:ListTags",
        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:parallelcluster-*",
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "s3:*"
    ],
    "Resource": [
        "arn:aws:s3:::parallelcluster-*",
        "arn:aws:s3:::aws-parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
},
{
    "Action": [
        "s3:Get*",
        "s3:List*"
    ],
    "Resource": "arn:aws:s3:::*-aws-parallelcluster*",
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
},
{
    "Action": [
        "elasticfilesystem:*"
    ],
    "Resource": [
        "arn:aws:elasticfilesystem:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "EFS"
},
{
    "Action": [

```

```

        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy",
        "logs:DescribeLogGroups",
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:CreateExportTask",
        "logs:DescribeLogStreams",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:PutMetricFilter",
        "logs>DeleteMetricFilter"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogs"
},
{
    "Action": [
        "resource-groups:ListGroupResources"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ResourceGroupRead"
},
{
    "Sid": "AllowDescribingFileCache",
    "Effect": "Allow",
    "Action": [
        "fsx:DescribeFileCaches"
    ],
    "Resource": "*"
},
{
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET
NAME>",
    "Effect": "Allow"
}
]
}

```



## AWS Batch 스케줄러 사용 시 추가 AWS ParallelCluster **pcluster** 사용자 정책

AWS Batch 스케줄러를 사용하여 클러스터를 생성하고 관리해야 하는 경우 다음과 같은 추가 정책이 필요합니다.

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

```

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

```

}

## Amazon FSx for Lustre를 사용할 때의 추가 AWS ParallelCluster **pcluster** 사용자 정책

Amazon FSx for Lustre를 사용하여 클러스터를 생성하고 관리해야 하는 경우 다음과 같은 추가 정책이 필요합니다.

### Note

기존 Amazon FSx 파일 시스템이 클러스터에서 사용되는 유일한 파일 시스템인 경우, Amazon FSx 정책 설명 예제를 클러스터 구성 파일의 [SharedStorage 섹션](#)에서 참조하는 특정 파일 시스템으로 범위를 좁힐 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "fsx.amazonaws.com",
            "s3.data-source.lustre.fsx.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "fsx:*"
      ],
      "Resource": [
        "arn:aws:fsx:*:<AWS ACCOUNT ID>:*"
      ],
    }
  ]
}
```

```

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

```

## AWS ParallelCluster 이미지 빌드 **pcluster** 사용자 정책

를 사용하여 사용자 지정 Amazon EC2 이미지를 생성하려는 사용자는 다음과 같은 권한 세트를 AWS ParallelCluster 가져야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeImages",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeInstanceTypes",
        "ec2:DeregisterImage",
        "ec2:DeleteSnapshot"
      ],
      "Resource": "*",
      "Effect": "Allow",
    }
  ]
}

```

```

        "Sid": "EC2"
    },
    {
        "Action": [
            "iam:CreateInstanceProfile",
            "iam:AddRoleToInstanceProfile",
            "iam:GetRole",
            "iam:GetRolePolicy",
            "iam:GetInstanceProfile",
            "iam:RemoveRoleFromInstanceProfile"
        ],
        "Resource": [
            "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
            "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/ParallelClusterImage*",
            "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
        ],
        "Effect": "Allow",
        "Sid": "IAM"
    },
    {
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": [
                    "lambda.amazonaws.com",
                    "ec2.amazonaws.com"
                ]
            }
        },
        "Action": [
            "iam:PassRole"
        ],
        "Resource": [
            "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
            "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
        ],
        "Effect": "Allow",
        "Sid": "IAMPassRole"
    },
    {
        "Action": [
            "logs:CreateLogGroup",
            "logs:TagResource",
            "logs:UntagResource",
            "logs>DeleteLogGroup"
        ]
    }

```

```

    ],
    "Resource": [
      "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/imagebuilder/
ParallelClusterImage-*",
      "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/lambda/
ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "CloudWatch"
  },
  {
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack"
    ],
    "Resource": [
      "arn:aws:cloudformation:*:<AWS ACCOUNT ID>:stack/*"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormation"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda:GetFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
      "lambda>DeleteFunction",
      "lambda:TagResource",
      "lambda:ListTags",
      "lambda:UntagResource"
    ],
    "Resource": [
      "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
  },
  {
    "Action": [
      "imagebuilder:Get*"
    ],
    "Resource": "*",

```

```

    "Effect": "Allow",
    "Sid": "ImageBuilderGet"
  },
  {
    "Action": [
      "imagebuilder:CreateImage",
      "imagebuilder:TagResource",
      "imagebuilder:CreateImageRecipe",
      "imagebuilder:CreateComponent",
      "imagebuilder:CreateDistributionConfiguration",
      "imagebuilder:CreateInfrastructureConfiguration",
      "imagebuilder>DeleteImage",
      "imagebuilder>DeleteComponent",
      "imagebuilder>DeleteImageRecipe",
      "imagebuilder>DeleteInfrastructureConfiguration",
      "imagebuilder>DeleteDistributionConfiguration"
    ],
    "Resource": [
      "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image-recipe/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:component/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:distribution-configuration/parallelclusterimage-*",
      "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:infrastructure-configuration/parallelclusterimage-*"
    ],
    "Effect": "Allow",
    "Sid": "ImageBuilder"
  },
  {
    "Action": [
      "s3:CreateBucket",
      "s3:ListBucket",
      "s3:ListBucketVersions"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3Bucket"
  },
  {

```

```

    "Action": [
      "sns:GetTopicAttributes",
      "sns:TagResource",
      "sns:CreateTopic",
      "sns:Subscribe",
      "sns:Publish",
      "SNS:DeleteTopic",
      "SNS:Unsubscribe"
    ],
    "Resource": [
      "arn:aws:sns:*:<AWS ACCOUNT ID>:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "SNS"
  },
  {
    "Action": [
      "s3:PutObject",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "s3:DeleteObject",
      "s3:DeleteObjectVersion"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "S3Objects"
  },
  {
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "imagebuilder.amazonaws.com"
      }
    }
  }
]
}

```



## AWS ParallelCluster IAM 리소스 관리를 위한 사용자 예제 정책

클러스터 또는 사용자 지정 AMI를 생성하는 AWS ParallelCluster 데 사용하는 경우 구성 요소에 필요한 권한 세트를 부여할 수 있는 권한이 포함된 IAM 정책을 제공해야 합니다. AWS ParallelCluster 이러한 IAM 리소스는 클러스터 AWS ParallelCluster 또는 사용자 지정 이미지를 생성할 때 자동으로 생성되거나 입력으로 제공될 수 있습니다.

다음 모드를 사용하면 구성에서 추가 IAM 정책을 사용하여 IAM 리소스에 액세스하는 데 필요한 권한을 AWS ParallelCluster 사용자에게 제공할 수 있습니다.

### 주제

- [권한 있는 IAM 액세스 모드](#)
- [제한된 IAM 액세스 모드](#)
- [PermissionsBoundary 모드](#)

### 권한 있는 IAM 액세스 모드

이 모드에서는 필요한 모든 IAM 리소스를 AWS ParallelCluster 자동으로 생성합니다. 이러한 IAM 정책은 클러스터 리소스에만 액세스할 수 있도록 범위가 축소되었습니다.

권한 있는 IAM 액세스 모드를 활성화하려면 사용자 역할에 다음 정책을 추가하세요.

#### Note

//[AdditionalPolicies](#) 또는 [HeadNodeIam/SchedulingSlurmQueuesIam/AdditionalPolicies](#) 파라미터를 구성하는 경우 다음 정책과 같이 각 추가 정책에 대한 역할 정책을 연결하고 분리할 수 있는 권한을 AWS ParallelCluster 사용자에게 제공해야 합니다. 역할 정책을 연결하고 분리하기 위한 조건에 추가 정책 ARN을 추가합니다.

#### Warning

이 모드를 사용하면 사용자가 IAM 관리자 권한을 가질 수 있습니다. AWS 계정

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

```

"Statement": [
  {
    "Action": [
      "iam:CreateServiceLinkedRole",
      "iam>DeleteRole",
      "iam:TagRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRole"
  },
  {
    "Action": [
      "iam>CreateRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamCreateRole"
  },
  {
    "Action": [
      "iam:PutRolePolicy",
      "iam>DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamInlinePolicy"
  },
  {
    "Condition": {
      "ArnLike": {
        "iam:PolicyARN": [
          "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
          "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
          "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
          "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
          "arn:aws:iam::aws:policy/AWSBatchFullAccess",
          "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
          "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",

```

```

        "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
        "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
        "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
    ]
  },
  "Action": [
    "iam:AttachRolePolicy",
    "iam:DetachRolePolicy"
  ],
  "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
  "Effect": "Allow",
  "Sid": "IamPolicy"
}
]
}

```

## 제한된 IAM 액세스 모드

사용자에게 추가 IAM 정책이 부여되지 않은 경우 클러스터나 사용자 지정 이미지 빌드에 필요한 IAM 역할을 관리자가 수동으로 생성하여 클러스터 구성의 일부로 전달해야 합니다.

클러스터를 생성할 때 다음 파라미터는 필수 파라미터입니다.

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

사용자 지정 이미지 빌드 시 다음 파라미터는 필수 파라미터입니다.

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

위에 나열된 파라미터의 일부로 전달된 IAM 역할은 `/parallelcluster/ path` 접두사에 생성되어야 합니다. 이렇게 할 수 없는 경우 다음 예와 같이 특정 사용자 지정 역할에 대한 `iam:PassRole` 권한을 부여하도록 사용자 정책을 업데이트해야 합니다.

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

### Warning

클러스터 구성에서 모든 IAM 역할을 전달할 수 없기 때문에 현재 이 모드에서는 AWS Batch 클러스터를 관리할 수 없습니다.

## PermissionsBoundary 모드

이 모드는 구성된 IAM 권한 AWS ParallelCluster 경계에 바인딩되는 IAM 역할 생성을 위임합니다. IAM 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티에 대한 권한 경계](#)를 참조하세요.

사용자 역할에 다음 정책을 추가해야 합니다.

정책에서 `< permissions-boundary-arn ># ## ###` 적용할 IAM 정책 ARN으로 대체하십시오.

**⚠ Warning**

[HeadNode/Iam/AdditionalPolicies](#) 또는 [Scheduling/SlurmQueues/Iam/AdditionalPolicies](#) 파라미터를 구성하는 경우 다음 정책과 같이 각 추가 정책에 대한 역할 정책을 연결하고 분리할 수 있는 권한을 사용자에게 부여해야 합니다. 역할 정책을 연결하고 분리하기 위한 조건에 추가 정책 ARN을 추가합니다.

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

```

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

```

```

        "iam:DetachRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamPolicy"
  }
]
}

```

이 모드를 활성화하면 클러스터를 생성하거나 업데이트할 때는 [Iam/PermissionsBoundary](#) 구성 파라미터에, 사용자 지정 이미지를 빌드할 때는 [Build/Iam/PermissionBoundary](#) 파라미터에 권한 경계 ARN을 지정해야 합니다.

## AWS ParallelCluster IAM 권한을 관리하기 위한 구성 파라미터

AWS ParallelCluster 클러스터에서 또는 사용자 지정 AMI 생성 프로세스 중에 사용되는 IAM 권한 및 역할을 사용자 지정하고 관리하기 위한 일련의 구성 옵션을 제공합니다.

주제

- [클러스터 구성](#)
- [사용자 지정 이미지 구성](#)

### 클러스터 구성

주제

- [헤드 노드 IAM 역할](#)
- [Amazon S3 액세스](#)
- [추가 IAM 정책](#)
- [AWS Lambda 함수 역할](#)
- [컴퓨팅 노드 IAM 역할](#)
- [권한 경계](#)

헤드 노드 IAM 역할

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

이 옵션을 사용하면 클러스터의 헤드 노드에 할당된 기본 IAM 역할을 재정의합니다. 자세한 내용은 [InstanceProfile](#) 자료를 참조하세요.

스케줄러가 Slurm일 때 이 역할의 일부로 사용되는 최소 정책 세트는 다음과 같습니다.

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` 관리형 IAM 정책입니다. 자세한 내용은 Amazon 사용 설명서에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 및 CloudWatch 사용자 [생성을](#) 참조하십시오.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 관리형 IAM 정책입니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 [AWS Systems Manager용 AWS 관리형 정책을](#) 참조하세요.
- 추가 IAM 정책:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.<REGION>/*",
        "arn:aws:s3:::parallelcluster-*v1-do-not-delete/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:BatchGetItem"
      ],
      "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/parallelcluster:node-type": "Compute"
        }
      }
    }
  ]
}
```



```

    },
    "Action": "ec2:TerminateInstances",
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:RunInstances",
      "ec2:CreateFleet"
    ]
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "ec2.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeVolumes",
      "ec2:DescribeInstanceAttribute",
      "ec2:DescribeCapacityReservations"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:CreateTags",

```

```

        "ec2:AttachVolume"
    ],
    "Resource": [
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackResource",
        "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT
ID>:secret:<SECRET_ID>",
    "Effect": "Allow"
}
]
}

```

참고로 [Scheduling/SlurmQueues/Iam/InstanceRole](#)를 사용하여 컴퓨팅 IAM 역할을 재정의하는 경우 위에 보고된 헤드 노드 정책은 iam:PassRole 권한의 Resource 섹션에 해당 역할을 포함해야 합니다.

스케줄러가 AWS Batch일 때 이 역할의 일부로 사용할 수 있는 최소 정책 세트는 다음과 같습니다.

- arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy 관리형 IAM 정책입니다. 자세한 내용은 Amazon 사용 설명서에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 및 CloudWatch 사용자 [생성을](#) 참조하십시오.

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 관리형 IAM 정책입니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 [AWS Systems Manager용 AWS 관리형 정책을 참조](#)하세요.
- 추가 IAM 정책:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::dcv-license.<REGION>/*",
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

```

    },
    "Action": [
        "batch:DescribeJobQueues",
        "batch:DescribeJobs",
        "batch:ListJobs",
        "batch:DescribeComputeEnvironments"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "batch:SubmitJob",
        "batch:TerminateJob",
        "logs:GetLogEvents",
        "ecs:ListContainerInstances",
        "ecs:DescribeContainerInstances",
    ],
    "Resource": [
        "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-stream:PclusterJobDefinition*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/AWSBatch-PclusterComputeEnviron*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/AWSBatch-Pcluster*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/PclusterJobQueue*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-definition/PclusterJobDefinition:*:*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{

```

```

    "Action": [
      "ec2:CreateTags",
      "ec2:AttachVolume"
    ],
    "Resource": [
      "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
      "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStacks",
      "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET_ID>",
    "Effect": "Allow"
  }
]
}

```

## Amazon S3 액세스

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

이 구성 섹션에서는 클러스터의 헤드 노드 또는 컴퓨팅 노드에 연결된 IAM 역할이 AWS ParallelCluster에 의해 생성될 때 역할에 추가 Amazon S3 정책을 부여하여 Amazon S3 액세스를 사용자 지정할 수 있습니다. 자세한 내용은 각 구성 파라미터에 대한 참조 설명서를 참조하세요.

이 파라미터는 사용자가 [권한 있는 IAM 액세스 모드](#) 또는 [PermissionsBoundary 모드](#)로 구성된 경우에만 사용할 수 있습니다.

## 추가 IAM 정책

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

에서 해당 역할을 생성할 때 클러스터의 헤드 노드 또는 컴퓨팅 노드와 관련된 IAM 역할에 추가 관리형 IAM 정책을 추가하려면 이 옵션을 사용하십시오. AWS ParallelCluster

### ⚠ Warning

이 옵션을 사용하려면, 연결해야 하는 IAM 정책에 대한 `iam:AttachRolePolicy` 및 `iam:DetachRolePolicy` 권한을 [AWS ParallelCluster 사용자](#)에게 부여해야 합니다.

## AWS Lambda 함수 역할

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

이 옵션은 클러스터 생성 프로세스 중에 사용되는 모든 AWS Lambda 함수에 연결된 역할을 대체합니다. AWS Lambda 역할을 맡을 수 있는 주도자로 구성해야 합니다.

### 📌 Note

[DeploymentSettings/LambdaFunctionsVpcConfig](#)가 설정된 경우 `LambdaFunctionsRole`에는 VPC 구성을 설정할 수 있는 [AWS Lambda 역할 권한](#)이 포함되어야 합니다.

이 역할의 일부로 사용할 최소 정책 세트는 다음과 같습니다.

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

```

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

```

## 컴퓨팅 노드 IAM 역할

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

이 옵션을 사용하면 클러스터의 컴퓨팅 노드에 할당된 IAM 역할을 재정의할 수 있습니다. 자세한 정보는 [InstanceProfile](#)을 참조하세요.

이 역할의 일부로 사용할 최소 정책 세트는 다음과 같습니다.

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` 관리형 IAM 정책입니다. 자세한 내용은 Amazon 사용 설명서에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 및 CloudWatch 사용자 [생성을](#) 참조하십시오.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 관리형 IAM 정책입니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 [AWS Systems Manager용 AWS 관리형 정책을](#) 참조하십시오.
- 추가 IAM 정책:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:Query",
        "dynamodb:UpdateItem",
        "dynamodb:PutItem",
        "dynamodb:GetItem"
      ],
      "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "ec2:DescribeInstanceAttribute",
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": "cloudformation:DescribeStackResource",
      "Resource": [
        "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```



```
}

```

## 권한 경계

### [Iam / PermissionsBoundary](#)

이 파라미터는 지정된 IAM 정책을 클러스터 배포의 일부로 생성되는 모든 IAM 역할에 강제로 AWS ParallelCluster 연결합니다. `PermissionsBoundary`

이 설정을 정의할 때 사용자에게 필요한 정책 목록은 [PermissionsBoundary 모드](#)를 참조하세요.

## 사용자 지정 이미지 구성

### 주제

- [EC2 Image Builder 인스턴스 역할](#)
- [AWS Lambda 정리 역할](#)
- [추가 IAM 정책](#)
- [권한 경계](#)

### EC2 Image Builder 인스턴스 역할

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

이 옵션을 사용하면 EC2 Image Builder에서 시작하는 Amazon EC2 인스턴스에 할당된 IAM 역할을 재정의하여 사용자 지정 AMI를 생성할 수 있습니다.

이 역할의 일부로 사용할 최소 정책 세트는 다음과 같습니다.

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` 관리형 IAM 정책입니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 [AWS Systems Manager용 AWS 관리형 정책](#)을 참조하세요.
- `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder` 관리형 IAM 정책입니다. 자세한 내용은 Image Builder 사용 설명서의 [EC2InstanceProfileForImageBuilder 정책](#)을 참조하세요.
- 추가 IAM 정책:

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

```

```

    {
      "Action": [
        "ec2:CreateTags",
        "ec2:ModifyImageAttribute"
      ],
      "Resource": "arn:aws:ec2:<REGION>::image/*",
      "Effect": "Allow"
    }
  ]
}

```

## AWS Lambda 정리 역할

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

이 옵션은 사용자 지정 이미지 빌드 프로세스 중에 사용되는 모든 AWS Lambda 함수에 연결된 역할을 재정의합니다. AWS Lambda 주도자가 역할을 맡을 수 있도록 구성해야 합니다.

#### Note

[DeploymentSettings/LambdaFunctionsVpcConfig](#)가 설정된 경우 CleanupLambdaRole에는 VPC 구성을 설정할 수 있는 [AWS Lambda 역할 권한](#)이 포함되어야 합니다.

이 역할의 일부로 사용할 최소 정책 세트는 다음과 같습니다.

- `arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` 관리형 IAM 정책입니다. 자세한 내용은 AWS Lambda 개발자 안내서에서 [Lambda 기능을 위한 AWS 관리형 정책](#)을 참조하세요.
- 추가 IAM 정책:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:DetachRolePolicy",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy"
      ],
    }
  ],
}

```

```

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

```

```

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

```

```

    "Resource": "arn:aws:sns:<REGION>:<AWS ACCOUNT ID>:ParallelClusterImage-
**",
    "Effect": "Allow"
  }
]
}

```

## 추가 IAM 정책

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

이 옵션을 사용하여 EC2 Image Builder에서 사용자 지정 AMI를 생성하는 데 사용하는 Amazon EC2 인스턴스와 연결된 역할에 관리형 IAM 정책을 추가할 수 있습니다.

#### Warning

이 옵션을 사용하려면, 연결해야 하는 IAM 정책에 대한 iam:AttachRolePolicy 및 iam:DetachRolePolicy 권한을 [AWS ParallelCluster사용자](#)에게 부여해야 합니다.

## 권한 경계

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

이 파라미터는 지정된 IAM 정책을 사용자 지정 AMI 빌드의 일부로 생성되는 모든 IAM 역할에 강제로 AWS ParallelCluster 연결합니다. PermissionsBoundary

이러한 기능을 사용하는 데 필요한 정책 목록은 [PermissionsBoundary 모드](#)을 참조하세요.

## 네트워크 구성

AWS ParallelCluster 네트워킹에 Amazon VPC (가상 사설 클라우드) 를 사용합니다. VPC는 클러스터를 배포하기 위한 유연하고 구성 가능한 네트워킹 플랫폼을 제공합니다.

VPC에는 리전에 대해 올바른 도메인 이름이 있는 DHCP 옵션, DNS Resolution = yes 및 DNS Hostnames = yes이 있어야 합니다. 기본 DHCP 옵션 세트는 이미 필수 AmazonProvided DNS를 지정하고 있습니다. 도메인 이름 서버를 두 개 이상 지정하는 경우 Amazon VPC 사용 설명서의 [DHCP 옵션 세트](#)를 참조하세요.

AWS ParallelCluster 다음과 같은 상위 수준 구성을 지원합니다.

- 헤드 노드와 컴퓨팅 노드 모두를 위한 하나의 서브넷
- 퍼블릭 서브넷에 헤드 노드가 있고 프라이빗 서브넷에 컴퓨팅 노드가 있는 두 개의 서브넷 서브넷은 새 서브넷이거나 기존 서브넷일 수 있습니다.

이러한 모든 구성은 퍼블릭 IP 주소 지정을 사용하거나 사용하지 않고 작동할 수 있습니다. AWS ParallelCluster 모든 AWS 요청에 HTTP 프록시를 사용하도록 배포할 수도 있습니다. 이러한 구성을 조합하면 많은 배포 시나리오가 가능해집니다. 예를 들어 인터넷을 통해 모든 액세스가 가능한 단일 퍼블릭 서브넷을 구성할 수 있습니다. 또는 모든 트래픽에 대해 HTTP 프록시를 사용하여 AWS Direct Connect 완전 사설망을 구성할 수 있습니다.

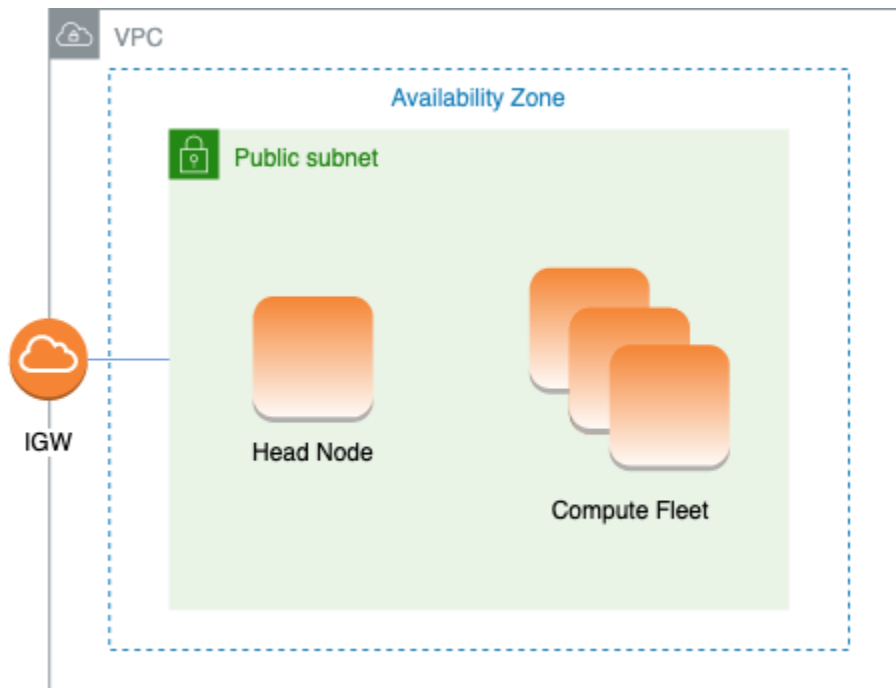
AWS ParallelCluster 3.0.0부터 각 대기열에 대해 서로 다른 SecurityGroups AdditionalSecurityGroups PlacementGroup 설정과 설정을 구성할 수 있습니다. 자세한 내용은 [HeadNode/Networking](#) 및 [SlurmQueues/Networking](#) 및 [AwsBatchQueues/Networking](#)를 참조하세요.

이러한 네트워킹 시나리오의 일러스트를 보려면 다음 아키텍처 다이어그램을 참조하세요.

#### 주제

- [AWS ParallelCluster 단일 퍼블릭 서브넷에서](#)
- [AWS ParallelCluster 서브넷 2개 사용](#)
- [AWS ParallelCluster 를 사용하여 연결된 단일 프라이빗 서브넷에서 AWS Direct Connect](#)
- [AWS ParallelCluster 스케줄러 포함 AWS Batch](#)
- [AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서](#)

## AWS ParallelCluster 단일 퍼블릭 서브넷에서



이 아키텍처의 구성에는 다음 설정이 필요합니다.

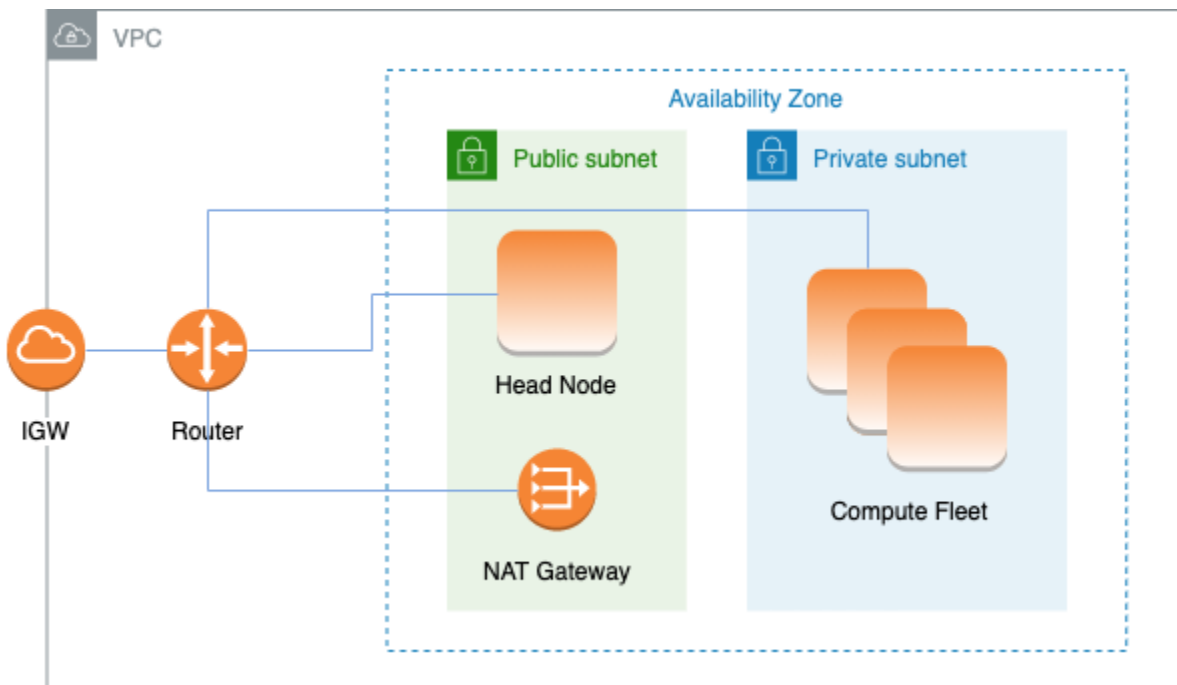
```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
        - subnet-12345678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

이 구성에서는 클러스터의 모든 인스턴스에 퍼블릭 IP를 할당해야 인터넷에 액세스할 수 있습니다. 이를 위해 다음을 수행합니다.

- [HeadNode/Networking/SubnetId](#)에서 사용되는 서브넷에 대해 “퍼블릭 IPv4 주소 자동 할당 활성화” 설정을 켜거나 [HeadNode/Networking/ElasticIp](#)에서 탄력적 IP를 할당하여 헤드 노드에 퍼블릭 IP 주소를 할당해야 합니다.
- [Scheduling/SlurmQueues/Networking/SubnetIds](#) 에서 사용되는 서브넷에 대해 “퍼블릭 IPv4 주소 자동 할당 활성화” 설정을 켜거나 [Scheduling/SlurmQueues/Networking](#)에서 [AssignPublicIp](#): true로 설정하여 컴퓨팅 노드에 퍼블릭 IP 주소를 할당해야 합니다.
- p4d인스턴스 유형을 정의하거나 헤드 노드에 대한 네트워크 인터페이스 카드가 여러 개 있는 다른 인스턴스 유형을 정의하는 경우 [HeadNodeNetworking](#)를 설정하여 퍼블릭 [ElasticIp](#) true 액세스를 제공해야 합니다. AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작된 인스턴스에만 할당할 수 있습니다. 이 경우 [NAT 게이트웨이](#)를 사용하여 클러스터 컴퓨팅 노드에 대한 퍼블릭 액세스를 제공하는 것이 좋습니다. IP 주소에 관한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.
- AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작하는 인스턴스에만 할당할 수 있기 때문에, p4d 또는 hp6id 인스턴스 유형이나 컴퓨팅 노드용 네트워크 인터페이스 카드가 여러 개 또는 한 개 있는 다른 인스턴스 유형을 정의할 수 없습니다. IP 주소에 관한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.

자세한 내용은 Amazon VPC 사용 설명서의 [인터넷 액세스 활성화](#)를 참조하세요.

## AWS ParallelCluster 서브넷 2개 사용





컴퓨팅 인스턴스에 대한 기존 프라이빗 서브넷을 사용하기 위한 구성에는 다음 설정이 필요합니다.

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

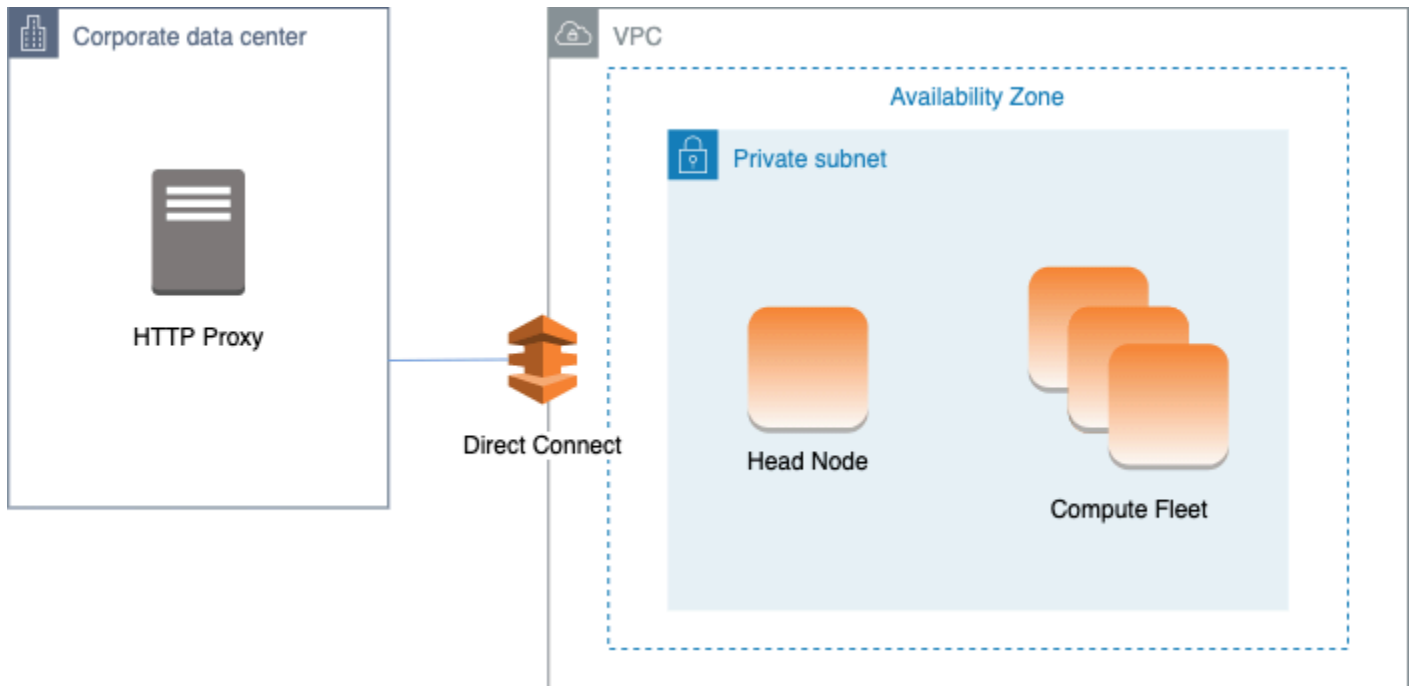
이 구성에서는 클러스터의 헤드 노드에만 퍼블릭 IP를 할당해야 합니다.

[HeadNode/Networking/SubnetId](#)에서 사용되는 서브넷에 대해 “퍼블릭 IPv4 주소 자동 할당 활성화” 설정을 켜거나 [HeadNode/Networking/ElasticIp](#)에서 탄력적 IP를 할당하면 됩니다.

헤드 노드에 대한 네트워크 인터페이스 카드 또는 네트워크 인터페이스 카드가 여러 개 있는 p4d 인스턴스 유형이나 다른 인스턴스 유형을 정의하는 경우 [HeadNodeNetworking//](#)를 설정하여 퍼블릭 [ElasticIptrue](#) 액세스를 제공해야 합니다. AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작된 인스턴스에만 할당할 수 있습니다. IP 주소에 관한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.

이 구성에는 컴퓨팅 인스턴스에 대한 인터넷 액세스를 제공하기 위해 대기열에 사용되는 서브넷의 [NAT 게이트웨이](#) 또는 내부 프록시가 필요합니다.

## AWS ParallelCluster 를 사용하여 연결된 단일 프라이빗 서브넷에서 AWS Direct Connect



이 아키텍처의 구성에는 다음 설정이 필요합니다.

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

[Scheduling/SlurmQueues/Networking/AssignPublicIp](#)를 `false`로 설정한 경우 모든 트래픽에 프록시를 사용하려면 VPC를 올바르게 설정해야 합니다. 헤드 노드와 컴퓨팅 노드 모두에 웹 액세스가 필요합니다.

## AWS ParallelCluster 스케줄러 포함 AWS Batch

를 스케줄러 `awsbatch` 유형으로 사용하면 AWS Batch 관리형 컴퓨팅 환경이 AWS ParallelCluster 생성됩니다. AWS Batch 환경은 Amazon Elastic Container Service(Amazon ECS) 컨테이너 인스턴스를 관리합니다. 이러한 인스턴스는 [AwsBatchQueues/Networking/SubnetIds](#) 파라미터로 구성된 서브넷에서 시작됩니다. Amazon ECS 컨테이너 인스턴스가 제대로 AWS Batch 작동하려면 Amazon ECS 서비스 엔드포인트와 통신하기 위한 외부 네트워크 액세스가 필요합니다. 이 경우는 다음 시나리오로 전환됩니다.

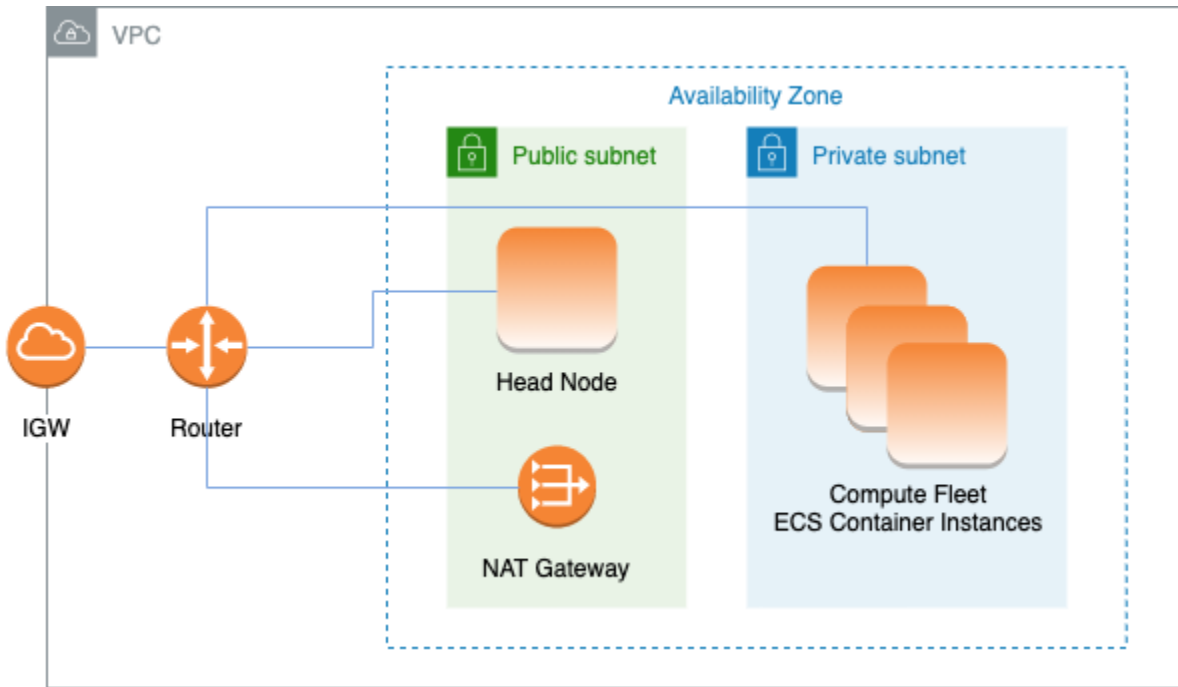
- 대기열에 지정된 서브넷 ID는 [NAT 게이트웨이](#)를 사용하여 인터넷에 액세스합니다. 이 방법이 권장 방법입니다.
- 대기열 서브넷에서 시작된 인스턴스는 퍼블릭 IP 주소를 가지고 있고 인터넷 게이트웨이를 통해 인터넷에 연결할 수 있습니다.

또한 다중 노드 병렬 작업에 관심이 있는 경우([AWS Batch 설명서](#)에 따라):

AWS Batch 다중 노드 병렬 작업은 Amazon ECS `awsvpc` 네트워크 모드를 사용합니다. 이렇게 하면 다중 노드 병렬 작업 컨테이너에 Amazon EC2 인스턴스와 동일한 네트워킹 속성이 적용됩니다. 각 다중 노드 병렬 작업 컨테이너는 고유의 탄력적 네트워크 인터페이스, 기본 프라이빗 IP 주소, 내부 DNS 호스트 이름을 가져옵니다. 네트워크 인터페이스는 호스트 컴퓨팅 리소스와 동일한 Amazon VPC 서브넷에서 생성됩니다. 컴퓨팅 리소스에 적용되는 모든 보안 그룹은 네트워크 인터페이스에도 적용됩니다.

Amazon ECS 작업 네트워킹을 사용할 경우 `awsvpc` 네트워크 모드는 Amazon EC2 시작 유형을 사용하는 작업에 대한 퍼블릭 IP 주소를 탄력적 네트워크 인터페이스에 제공하지 않습니다. 인터넷에 액세스하려면 Amazon EC2 시작 유형을 사용하는 작업이 NAT 게이트웨이를 사용하도록 구성된 프라이빗 서브넷에서 시작되어야 합니다.

클러스터가 다중 노드 병렬 작업을 실행할 수 있도록 하려면 [NAT 게이트웨이](#)를 구성해야 합니다.



이전의 모든 구성 및 고려 사항도 AWS Batch 유효합니다. 다음은 AWS Batch 네트워킹 구성의 예입니다.

# Note that all values are only provided as examples

HeadNode:

...

Networking:

SubnetId: subnet-12345678 # subnet with internet gateway, NAT gateway or proxy

#ElasticIp: true | false | eip-12345678

#Proxy:

#HttpProxyAddress: http://proxy-address:port

Ssh:

KeyName: ec2-key-name

Scheduling:

Scheduler: awsbatch

AwsBatchQueues:

- ...

Networking:

SubnetIds:

- subnet-23456789 # subnet with internet gateway, NAT gateway or proxy

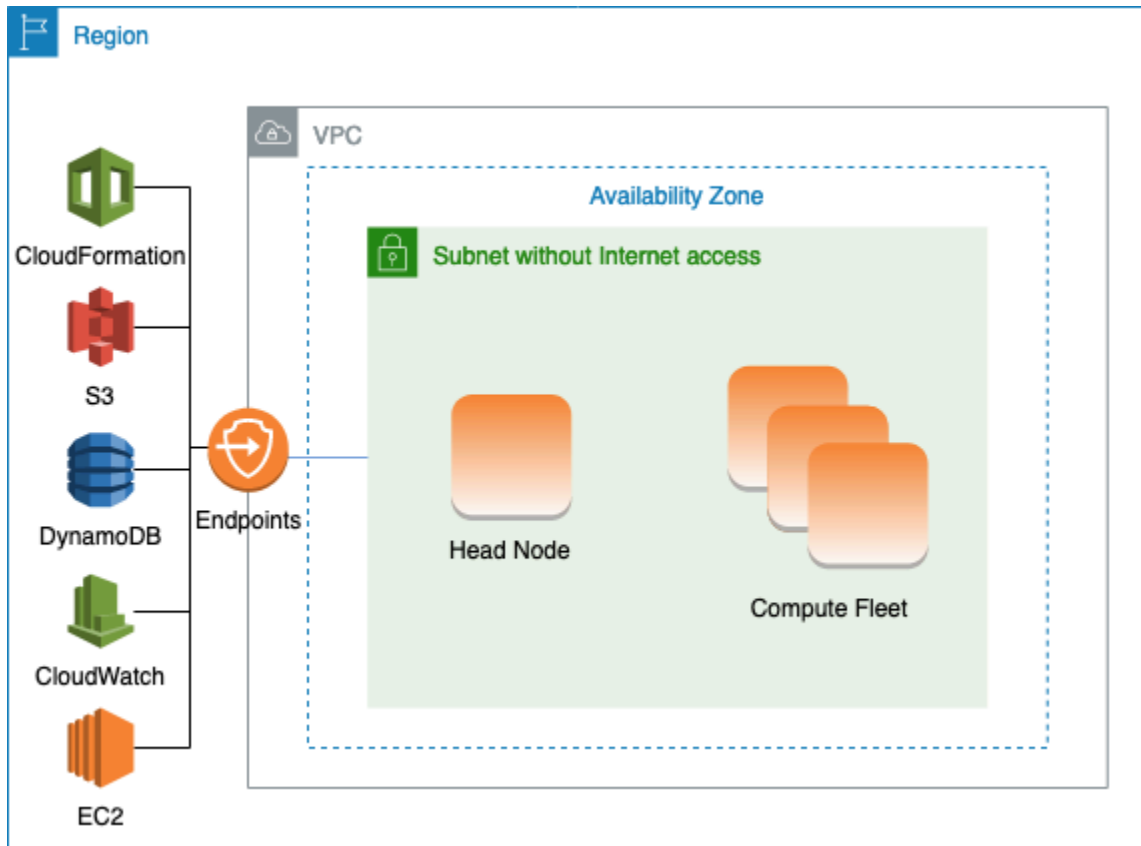
#AssignPublicIp: true | false

[Scheduling/AwsBatchQueues/Networking](#) 섹션에서, [SubnetIds](#)는 목록 유형이지만 현재는 하나의 서브넷만 지원됩니다.

자세한 정보는 다음 주제를 참조하세요.

- [AWS Batch 관리형 컴퓨팅 환경](#)
- [AWS Batch 다중 노드 Parallel 작업](#)
- [awscli 네트워크 모드를 이용한 Amazon ECS 태스크 네트워킹](#)

## AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서



인터넷에 액세스할 수 없는 서브넷은 인터넷으로의 인바운드 또는 아웃바운드 연결을 허용하지 않습니다. 이 AWS ParallelCluster 구성은 보안에 관심이 있는 고객이 리소스 보안을 더욱 강화하는데 도움이 될 수 있습니다. AWS ParallelCluster AWS ParallelCluster 노드는 인터넷 액세스 없이 클러스터를 실행하는 데 필요한 모든 소프트웨어가 포함된 AMI를 기반으로 구축됩니다. 이렇게 하면 AWS ParallelCluster가 인터넷에 액세스할 수 없는 노드로 클러스터를 만들고 관리할 수 있습니다.

이 섹션에서는 클러스터를 구성하는 방법에 대해 알아보십시오. 또한 인터넷에 접속하지 않고 클러스터를 실행할 때의 제한 사항에 대해서도 알아보십시오.

### VPC 엔드포인트 구성

클러스터가 제대로 작동하려면 클러스터 노드가 여러 AWS 서비스와 상호 작용할 수 있어야 합니다.

클러스터 노드가 인터넷에 액세스하지 않고도 AWS 서비스와 상호 작용할 수 있도록 다음 [VPC 엔드포인트](#)를 만들고 구성하십시오.

#### Commercial and AWS GovCloud (US) partitions

Service	서비스 이름	유형
아마존 CloudWatch	com.amazonaws. <i>region-id</i> .logs	인터페이스
AWS CloudFormation	com.amazonaws. <i>region-id</i> .cloudformation	인터페이스
Amazon EC2	com.amazonaws. <i>region-id</i> .ec2	인터페이스
Amazon S3	com.amazonaws. <i>region-id</i> .s3	게이트웨이
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	게이트웨이
AWS Secrets Manager**	com.amazonaws. <i>region-id</i> .secretsmanager	인터페이스

#### China partition

Service	서비스 이름	유형
아마존 CloudWatch	com.amazonaws. <i>region-id</i> .logs	인터페이스
AWS CloudFormation	cn.com.amazonaws. <i>region-id</i> .cloudformation	인터페이스

Service	서비스 이름	유형
Amazon EC2	cn.com.am azonaws. <i>region-id</i> .ec2	인터페이스
Amazon S3	com.amazonaws. <i>region-id</i> .s3	게이트웨이
Amazon DynamoDB	com.amazonaws. <i>region-id</i> .dynamodb	게이트웨이
AWS Secrets Manager**	com.amazonaws. <i>region-id</i> .secretsmanager	인터페이스

\*\* 이 엔드포인트는 [DirectoryService](#)가 활성화된 경우에만 필요하며, 그렇지 않으면 선택 사항입니다.

VPC의 모든 인스턴스에는 엔드포인트와 통신할 수 있는 적절한 보안 그룹이 있어야 합니다. [HeadNode](#) 아래의 [AdditionalSecurityGroups](#) 및 [SlurmQueues](#) 구성 아래의 [AdditionalSecurityGroups](#)에 보안 그룹을 추가하여 이 작업을 수행할 수 있습니다. 예를 들어 보안 그룹을 명시적으로 지정하지 않고 VPC 엔드포인트를 생성하는 경우 기본 보안 그룹이 엔드포인트와 연결됩니다. [AdditionalSecurityGroups](#)에 기본 보안 그룹을 추가하면 클러스터와 엔드포인트 간의 통신을 활성화할 수 있습니다.

#### Note

IAM 정책을 사용하여 VPC 엔드포인트에 대한 액세스를 제한하는 경우 Amazon S3 VPC 엔드포인트에 다음을 추가해야 합니다.

```
PolicyDocument:
  Version: 2012-10-17
  Statement:
    - Effect: Allow
      Principal: "*"
      Action:
        - "s3:PutObject"
      Resource:
```

```
- !Sub "arn:${AWS::Partition}:s3:::cloudformation-waitcondition-
${AWS::Region}/*"
```

Route 53을 비활성화하고 Amazon EC2 호스트 이름을 사용하십시오.

Slurm클러스터를 생성할 때 사용자 지정 컴퓨팅 노드 호스트 이름 (예:) 을 확인하는 데 사용되는 프라이빗 Route 53 호스팅 영역을 AWS ParallelCluster 생성합니다. {queue\_name}-{st|dy}-{compute\_resource}-{N} Route 53은 VPC 엔드포인트를 지원하지 않으므로 이 기능을 비활성화해야 합니다. 또한 기본 Amazon EC2 호스트 이름 (예:) 을 사용하도록 AWS ParallelCluster 구성해야 합니다. ip-1-2-3-4 클러스터 구성에 다음 설정을 적용하세요.

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

#### Warning

[SlurmSettingsDnsDisableManagedDns](#)로 생성한 클러스터의 경우 [UseEc2Hostnames](#) 설정된 클러스터의 true 경우 Slurm NodeName DNS가 이를 확인하지 않습니다. SlurmNodeHostName 대신 를 사용하세요.

#### Note

이 노트는 AWS ParallelCluster 버전 3.3.0부터는 관련이 없습니다.

3.3.0 이전 AWS ParallelCluster 지원 버전의 경우:

UseEc2Hostnames가 로 설정된 true 경우 Slurm 구성 파일은 AWS ParallelCluster prolog 및 epilog 스크립트로 설정됩니다.

- prolog를 실행하여 각 작업이 할당될 때 컴퓨팅 노드의 /etc/hosts에 노드 정보를 추가합니다.
- epilog를 실행하여 prolog가 작성한 내용을 정리합니다.



사용자 정의 prolog 또는 epilog 스크립트를 추가하려면 각각 /opt/slurm/etc/pcluster/prolog.d/ 또는 /opt/slurm/etc/pcluster/epilog.d/ 폴더에 추가하세요.

## 클러스터 구성

인터넷에 연결되지 않은 서브넷에서 클러스터가 실행되도록 구성하는 방법을 알아보세요.

이 아키텍처의 구성에는 다음 설정이 필요합니다.

```
# Note that all values are only provided as examples
...
HeadNode:
  ...
  Networking:
    SubnetId: subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
    endpoints
    AdditionalSecurityGroups:
      - sg-abcdef01234567890 # optional, the security group that enables the
      communication between the cluster and the VPC endpoints
  Scheduling:
    Scheduler: Slurm # Cluster in a subnet without internet access is supported only when
    the scheduler is Slurm.
    SlurmSettings:
      Dns:
        DisableManagedDns: true
        UseEc2Hostnames: true
    SlurmQueues:
      - ...
      Networking:
        SubnetIds:
          - subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
          endpoints attached
        AdditionalSecurityGroups:
          - sg-1abcdef01234567890 # optional, the security group that enables the
          communication between the cluster and the VPC endpoints
```

- [SubnetId\(s\)](#): 인터넷에 액세스할 수 없는 서브넷.

AWS ParallelCluster AWS 서비스와 서비스 간의 통신을 활성화하려면 서브넷의 VPC에 VPC 엔드포인트가 연결되어 있어야 합니다. 클러스터를 생성하기 전에 서브넷에서 [퍼블릭 IPv4 주소 자동 할당이 비활성화되어](#) 있는지 확인하여 pcluster 명령이 클러스터에 액세스할 수 있는지 확인하세요.

- [AdditionalSecurityGroups](#): 클러스터와 VPC 엔드포인트 간의 통신을 지원하는 보안 그룹입니다.

#### 선택 사항:

- 보안 그룹을 명시적으로 지정하지 않고 VPC 엔드포인트를 생성하는 경우 VPC의 기본 보안 그룹이 연결됩니다. 따라서 AdditionalSecurityGroups에 기본 보안 그룹을 제공하세요.
- 클러스터 및/또는 VPC 엔드포인트를 생성할 때 사용자 지정 보안 그룹을 사용하는 경우, AdditionalSecurityGroups는 사용자 지정 보안 그룹이 클러스터와 VPC 엔드포인트 간의 통신을 지원하는 한 필요하지 않습니다.

- [Scheduler](#): 클러스터 스케줄러.

유일한 유효 값은 slurm입니다. 인터넷 액세스가 없는 서브넷의 클러스터를 지원하는 것은 Slurm 스케줄러뿐입니다.

- [SlurmSettings](#): 설정. Slurm

이전 섹션 Route53 비활성화 및 Amazon EC2 호스트 이름 사용을 참조하십시오.

#### 제한 사항

- SSH 또는 NICE DCV를 통해 헤드 노드에 연결: 클러스터에 연결할 때 연결 클라이언트가 프라이빗 IP 주소를 통해 클러스터의 헤드 노드에 도달할 수 있는지 확인합니다. 클라이언트가 헤드 노드와 같은 VPC에 있지 않은 경우 VPC의 퍼블릭 서브넷에 있는 프록시 인스턴스를 사용하세요. 이 요구 사항은 SSH 및 DCV 연결 모두에 적용됩니다. 서브넷이 인터넷에 액세스할 수 없는 경우 헤드 노드의 퍼블릭 IP에 액세스할 수 없습니다. pcluster ssh 및 dcv-connect 명령은 퍼블릭 IP가 있는 경우 퍼블릭 IP를 사용하거나 프라이빗 IP를 사용합니다. 클러스터를 생성하기 전에 서브넷에서 [퍼블릭 IPv4 주소 자동 할당이 비활성화되어](#) 있는지 확인하여 pcluster 명령이 클러스터에 액세스할 수 있는지 확인하세요.

다음 예는 클러스터의 헤드 노드에서 실행되는 DCV 세션에 연결하는 방법을 보여줍니다. 프록시 Amazon EC2 인스턴스를 통해 연결합니다. 인스턴스는 PC의 NICE DCV 서버, 프라이빗 서브넷의 헤드 노드용 클라이언트 역할을 합니다.

퍼블릭 서브넷의 프록시 인스턴스를 통해 DCV를 통해 연결:

1. 클러스터의 서브넷과 동일한 VPC에 있는 퍼블릭 서브넷에 Amazon EC2 인스턴스를 생성합니다.
  2. NICE DCV 클라이언트와 서버가 Amazon EC2 인스턴스에 설치되어 있는지 확인합니다.
  3. 프록시 Amazon EC2 인스턴스에 AWS ParallelCluster 사용자 정책을 연결합니다. 자세한 정보는 [AWS ParallelCluster 사용자 정책 예시 pcluster](#)을 참조하세요.
  4. 프록시 Amazon EC2 인스턴스에 AWS ParallelCluster 설치합니다.
  5. DCV를 통해 프록시 Amazon EC2 인스턴스에 연결합니다.
  6. 프록시 인스턴스의 `pcluster dcv-connect` 명령을 사용하여 인터넷 액세스 없이 서브넷 내부의 클러스터에 연결할 수 있습니다.
- 다른 AWS 서비스와의 상호 작용: 반드시 필요한 서비스만 위에 나열되어 AWS ParallelCluster 있습니다. 클러스터가 다른 서비스와 상호 작용해야 하는 경우 해당 VPC 엔드포인트를 생성하세요.

## 로그인 노드

버전 3.7.0부터 AWS ParallelCluster 클러스터 관리자는 사용자에게 작업을 실행하는 데 사용할 수 있는 액세스 권한을 제공하는 데 사용할 수 있는 로그인 노드를 프로비저닝하는 대신 클러스터 헤드 노드에 직접 액세스하는 데 사용할 수 있습니다. 적절한 권한이 있는 클러스터 사용자는 Active Directory 또는 ssh 보안 인증 정보를 사용하여 로그인하고 작업을 제출하고 관리할 수 있습니다. 따라서 클러스터 관리가 개선되고 클러스터를 관리하는 Slurm 데 필요한 헤드 노드의 리소스가 고갈될 가능성을 최소화할 수 있습니다. 로그인한 사용자는 로그인 노드에 탑재된 클러스터의 모든 공유 스토리지에도 액세스할 수 있습니다. 로그인 노드를 중지해야 하는 경우 로그인한 사용자는 사용 중인 활성 셸 세션을 통해 사전에 알림을 받게 됩니다.

로그인 노드는 풀로 지정되며, 풀은 동일한 리소스 구성을 가진 로그인 노드 그룹을 정의합니다. 풀의 모든 로그인 노드는 라운드 로빈 방식으로 로그인 노드 간에 세션을 분산할 수 있는 [Network Load Balancer](#)의 일부로 구성됩니다. 현재 구현에서는 여러 로그인 노드를 포함하는 로그인 노드 풀 하나를 지정할 수 있습니다.

### 보안

로그인 노드는 헤드 노드의 AllowedIPs 설정 [AllowedIps](#)를 상속합니다. 이러한 방식으로 클러스터 관리자는 SSH 연결이 허용되는 소스 CIDR 또는 접두사 목록을 지정하여 클러스터의 보안 상태를 제한할 수 있습니다.

현재 구현에서는 로그인 노드를 활성화할 때 헤드 노드에 대한 액세스가 자동으로 제한되지 않습니다. 필요한 경우 클러스터 관리자는 표준 Linux 명령을 사용하여 헤드 노드의 ssh 구성을 업데이트하

여 이러한 액세스를 제한할 수 있습니다. 또한 ParallelCluster YAML 파일의 헤드 노드 섹션에 있는 `AdditionalSecurityGroups` 설정을 사용하여 헤드 노드에 사용자 지정 보안 그룹을 지정하여 승인되지 않은 사용자의 연결을 거부하는 방법으로도 이 작업을 수행할 수 있습니다.

## 네트워킹

로그인 노드에는 로그인 노드 풀용으로 구성된 Network Load Balancer에 대한 단일 연결 주소가 제공됩니다. 주소의 연결 설정은 로그인 노드 풀 구성에 지정된 서브넷 유형을 기반으로 합니다.

- 서브넷이 프라이빗인 경우 주소는 프라이빗이 되며, 로그인 노드에 대한 액세스 권한을 부여하려면 클러스터 관리자가 Bastion Host를 프로비저닝해야 합니다.
- 서브넷이 퍼블릭인 경우 주소는 퍼블릭입니다.

모든 연결 요청은 Network Load Balancer에서 라운드 로빈 라우팅을 사용하여 관리합니다.

## 스토리지

관리형 스토리지를 사용하여 ParallelCluster 클러스터에 구성된 모든 공유 스토리지는 모든 로그인 노드에 마운트됩니다.

## 로그인 노드 정보 검색

로그인 노드에 액세스하기 위해 프로비저닝된 단일 연결의 주소를 검색하려면 클러스터 관리자가 [describe-cluster](#) 명령을 실행할 수 있습니다. 이 명령은 로그인 노드의 상태에 대한 자세한 정보도 제공합니다.

로그인 노드는 지원되는 새 노드 유형으로, 특정 노드 유형의 상태를 쿼리할 때 [describe-cluster-instances](#) 명령으로 지정할 수 있습니다. ParallelCluster

로그인 노드 풀에 단일 연결 주소를 사용할 수 있다고 해서 특정 로그인 노드에 직접 액세스할 수 있는 것은 아닙니다. 하지만 ssh 클라이언트의 경고를 피하려면 직접 연결을 사용하지 않는 것이 좋습니다. ssh 클라이언트는 각 대상 주소의 호스트 식별자를 로컬에 저장합니다. 호스트 식별자는 풀마다 다르므로 다른 IP 및/또는 단일 연결 주소를 사용하면 다른 대상 주소와 관련된 동일한 호스트 식별자를 가질 수 있습니다. 이렇게 하면 동일한 호스트 식별자가 여러 대상에 연결되므로 ssh 클라이언트에서 경고가 발생할 수 있습니다.

## Imds 속성

로그인 노드의 IMDS (및 인스턴스 프로파일 자격 증명)에 대한 액세스는 루트 사용자, 클러스터 관리자 사용자 (기본 설정) 및 운영 `pc-cluster-admin` 체제별 기본 사용자 (Amazon ec2-user Linux 2, Ubuntu 18.04 RedHat, ubuntu centos CentOS 7) 로 제한됩니다.

## IMDS 액세스를 제한하기 위해 연쇄를 관리합니다. AWS ParallelCluster iptables

### Note

iptables 또는 ip6tables 규칙을 사용자 지정하면 로그인 노드에서 IMDS 액세스를 제한하는 데 사용되는 메커니즘이 방해될 수 있습니다. [Imds property setting](#)을 참조하세요.

### 로그인 노드 생명 주기

현재 풀에서 로그인 노드를 중지하고 시작하는 전용 명령은 없습니다. 풀의 로그인 노드를 중지하려면 클러스터 관리자가 로그인 노드(Count: 0) 수에 0을 지정하여 클러스터 구성을 업데이트한 다음 [pcluster.update-cluster-v3](#) 명령을 실행해야 합니다.

### Note

로그인한 사용자에게는 특정 인스턴스의 종료 및 관련 유예 기간에 대한 알림이 전송됩니다. 유예 기간 동안에는 [클러스터 기본 사용자](#)의 연결을 제외하고 새 연결이 허용되지 않습니다. 표시된 메시지는 클러스터 관리자가 헤드 노드 또는 로그인 노드에서 `/opt/parallelcluster/shared_login_nodes/loginmgt_config.json` 파일을 편집하여 사용자 정의할 수 있습니다.

로그인 노드 풀을 시작하려면 클러스터 관리자가 클러스터 구성의 이전 Count 값을 복원한 다음 [update-cluster](#) 명령을 실행해야 합니다.

로그인 노드 풀을 실행하는 데 필요한 권한

로그인 노드 풀을 관리하려면 클러스터 관리자에게 다음과 같은 추가 권한이 있어야 합니다.

- Action:
  - autoscaling:DeleteAutoScalingGroup
  - autoscaling:DeleteLifecycleHook
  - autoscaling:Describe\*
  - autoscaling:PutLifecycleHook
  - autoscaling:UpdateAutoScalingGroup
  - elasticloadbalancing:CreateListener
  - elasticloadbalancing:CreateTargetGroup
  - elasticloadbalancing>DeleteListener

```

- elasticloadbalancing:DeleteLoadBalancer
- elasticloadbalancing:DeleteTargetGroup
- elasticloadbalancing:Describe*
- elasticloadbalancing:ModifyLoadBalancerAttributes
Resource: '*'
Condition:
  ForAllValues:StringEquals:
    aws:TagKeys: [ "parallelcluster:cluster-name" ]
- Action:
  - autoscaling:CreateAutoScalingGroup
  - elasticloadbalancing:AddTags
  - elasticloadbalancing:CreateLoadBalancer
Resource: '*'
Effect: Allow

```

## 사용자 지정 부트스트랩 작업

[HeadNode/CustomActions/OnNodeStart](#) 구성 설정을 정의하면 노드가 시작된 직후 임의의 코드를 AWS ParallelCluster 실행합니다. [HeadNode/CustomActions/OnNodeConfigured](#) 구성 설정을 정의하는 경우 노드 구성이 올바르게 완료된 후 코드를 AWS ParallelCluster 실행합니다.

AWS ParallelCluster 버전 3.4.0부터 [HeadNodeCustomActions/OnNodeUpdated](#) 구성 설정을 정의하면 헤드 노드 업데이트 후 코드를 실행할 수 있습니다.

대부분의 경우 이 코드는 Amazon Simple Storage Service(S3)에 저장되며 HTTPS 연결을 통해 액세스됩니다. 이 코드는 root로서 실행되며 클러스터 OS에서 지원되는 어떠한 스크립트 언어로든 작성될 수 있습니다. 코드는 일반적으로 Bash 또는 Python으로 작성됩니다.

### Note

AWS ParallelCluster 버전 3.7.0부터 클러스터 [Imds/ImdsSupport](#) 설정 기본값은 `v2.0`입니다.  
 새 클러스터를 생성하여 버전 3.7.0 이상으로 업그레이드할 때는 사용자 지정 부트스트랩 액션 스크립트를 IMDSv2와 호환되도록 업데이트하거나 클러스터 구성 파일에서 [Imds/ImdsSupport](#)를 `v1.0`에 설정하세요.

**⚠ Warning**

[공동 책임 모델](#)에 설명된 대로 사용자 정의 스크립트와 인수를 구성하는 것은 사용자의 책임입니다. 사용자 지정 부트스트랩 스크립트와 인수가 클러스터 노드에 대한 전체 액세스 권한이 있다고 신뢰할 수 있는 소스에서 가져온 것인지 확인하세요.

**⚠ Warning**

AWS ParallelCluster 파일을 통해 제공되는 내부 변수의 사용을 지원하지 않습니다. `/etc/parallelcluster/cfnconfig` 이 파일은 향후 릴리스에서 제거될 수 있습니다.

`OnNodeStart` 작업은 NAT, Amazon Elastic Block Store(Amazon EBS) 또는 스케줄러 구성과 같은 노드 배포 부트스트랩 작업이 시작되기 전에 호출됩니다. `OnNodeStart` 부트스트랩 작업에는 스토리지 수정, 사용자 추가, 패키지 추가 등이 포함될 수 있습니다.

**i Note**

클러스터용 [HeadNode/CustomActions/OnNodeStart](#) 스크립트를 구성하는 [DirectoryService](#) 경우 스크립트를 실행하기 전에 `sssd` 를 AWS ParallelCluster `DirectoryService` 구성하고 다시 시작합니다. `sssd OnNodeStart`

노드 부트스트랩 프로세스가 완료된 후 `OnNodeConfigured` 작업이 호출됩니다.

`OnNodeConfigured` 작업은 인스턴스가 완전히 구성되고 완료된 것으로 간주되기 전에 발생하는 마지막 작업을 수행합니다. 일부 `OnNodeConfigured` 작업에는 스케줄러 설정 변경, 스토리지 및 패키지 수정이 포함될 수 있습니다. 구성 중에 인수를 지정하여 인수를 스크립트에 전달할 수 있습니다.

헤드 노드 업데이트가 완료되고 스케줄러와 공유 스토리지가 최신 클러스터 구성 변경 사항에 맞게 조정된 후에 `OnNodeUpdated` 작업이 호출됩니다.

`OnNodeStart` 또는 `OnNodeConfigured` 사용자 지정 작업이 성공하면 종료 코드 제로(0)로 성공 여부가 표시됩니다. 다른 종료 코드는 인스턴스 부트스트랩이 실패했음을 나타냅니다.

`OnNodeUpdated` 사용자 지정 작업이 성공하면 종료 코드 제로(0)와 함께 성공 신호가 표시됩니다. 다른 종료 코드는 업데이트가 실패했음을 나타냅니다.

**Note**

[OnNodeUpdated](#)를 구성한 경우 업데이트 실패 시 수동으로 OnNodeUpdated 작업을 이전 상태로 복원해야 합니다.

OnNodeUpdated 사용자 지정 작업이 실패하면 업데이트가 이전 상태로 롤백됩니다. 하지만 OnNodeUpdated 작업은 업데이트 시에만 실행되며 스택 롤백 시에는 실행되지 않습니다.

[HeadNode/CustomActions](#) 및 [Scheduling/SlurmQueues/CustomActions](#) 구성 섹션에서 헤드 노드와 각 대기열에 대해 서로 다른 스크립트를 지정할 수 있습니다. [OnNodeUpdated](#)는 HeadNode 섹션에서만 구성할 수 있습니다.

**Note**

AWS ParallelCluster 버전 3.0 이전에는 헤드 노드와 컴퓨팅 노드에 다른 스크립트를 지정할 수 없었습니다. [AWS ParallelCluster 2.x에서 3.x로 이동](#) 섹션을 참조하세요.

## 주제

- [구성](#)
- [인수](#)
- [사용자 지정 부트스트랩 작업이 포함된 예제 클러스터](#)
- [IMDSv2의 사용자 지정 부트스트랩 스크립트 업데이트 예제](#)
- [IMDSv1의 구성 업데이트 예제](#)

## 구성

다음 구성 설정은 [HeadNode/CustomActions/OnNodeStart](#) & [OnNodeConfigured](#) & [OnNodeUpdated](#) 및 [Scheduling/CustomActions/OnNodeStart](#) & [OnNodeConfigured](#) 작업 및 인수를 정의하는 데 사용됩니다.

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
      # Script URL. This is run before any of the bootstrap scripts are run
      Script: s3://bucket-name/on-node-start.sh
```



```
  Args:
    - arg1
  OnNodeConfigured:
    # Script URL. This is run after all the bootstrap scripts are run
    Script: s3://bucket-name/on-node-configured.sh
    Args:
      - arg1
  OnNodeUpdated:
    # Script URL. This is run after the head node update is completed.
    Script: s3://bucket-name/on-node-updated.sh
    Args:
      - arg1
  # Bucket permissions
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false
  Scheduling:
    Scheduler: slurm
    [...]
  SlurmQueues:
    - Name: queue1
      [...]
    CustomActions:
      OnNodeStart:
        Script: s3://bucket-name/on-node-start.sh
        Args:
          - arg1
      OnNodeConfigured:
        Script: s3://bucket-name/on-node-configured.sh
        Args:
          - arg1
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false
```

Sequence 설정 사용 ( AWS ParallelCluster 버전 3.6.0에 추가됨):

```
HeadNode:
  [...]
  CustomActions:
    OnNodeStart:
```

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

```

OnNodeStart:
  # Script URLs. The scripts are run in the same order as listed in the
  configuration, before any of the bootstrap scripts are run
  Sequence:
    - Script: s3://bucket-name/on-node-start1.sh
      Args:
        - arg1
    - Script: s3://bucket-name/on-node-start2.sh
      Args:
        - arg1
    [...]
OnNodeConfigured:
  # Script URLs. The scripts are run in the same order as listed in the
  configuration, after all the bootstrap scripts are run
  Sequence:
    - Script: s3://bucket-name/on-node-configured1.sh
      Args:
        - arg1
    - Script: s3://bucket-name/on-node-configured2.sh
      Args:
        - arg1
    [...]
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false

```

이 Sequence 설정은 AWS ParallelCluster 버전 3.6.0부터 추가되었습니다. 지정할 Sequence 때 사용자 지정 작업에 사용할 여러 스크립트를 나열할 수 있습니다. AWS ParallelCluster 스크립트는 포함하지 않고 단일 스크립트로 사용자 정의 액션을 구성할 수 있도록 계속 지원합니다. Sequence.

AWS ParallelCluster 단일 스크립트와 Sequence 동일한 사용자 지정 작업을 모두 포함하는 것은 지원하지 않습니다. 예를 들어, 다음과 같은 구성을 지정하면 AWS ParallelCluster 실패합니다.

```

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

```

```
Sequence:
- Script: s3://bucket-name/on-node-start1.sh
  Args:
    - arg1
- Script: s3://bucket-name/on-node-start2.sh
  Args:
    - arg1
```

[...]

## 인수

### Note

AWS ParallelCluster 2.x에서는 인수가 사용자 지정 스크립트의 URL을 저장하기 위한 예약된 \$1 인수였습니다. AWS ParallelCluster 2.x용으로 만든 사용자 지정 부트스트랩 스크립트를 AWS ParallelCluster 3.x에서 다시 사용하려면 인수의 변화를 고려하여 수정해야 합니다. [AWS ParallelCluster 2.x에서 3.x로 이동](#) 섹션을 참조하세요.

## 사용자 지정 부트스트랩 작업이 포함된 예제 클러스터

다음 단계는 노드를 구성한 후 실행할 간단한 스크립트를 생성하여 클러스터의 노드에 R, curl 및 wget 패키지를 설치합니다.

### 1. 스크립트를 생성합니다.

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

2. 올바른 권한을 사용하여 스크립트를 Amazon S3에 업로드합니다. 공개 읽기 권한이 적절하지 않은 경우 [HeadNode/Iam/S3Access](#) 및 [Scheduling/SlurmQueues](#) 구성 섹션을 사용하세요. 자세한 정보는 [Amazon S3 작업](#)을 참조하세요.

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

**⚠ Important**

Windows에서 스크립트를 편집한 경우 스크립트를 Amazon S3에 업로드하기 전에 줄 끝을 CRLF에서 LF로 변경해야 합니다.

**3. 새 작업을 포함하도록 AWS ParallelCluster 구성을 업데이트하십시오. OnNodeConfigured**

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

버킷에 public-read 권한이 없는 경우 s3을 URL 프로토콜로 사용합니다.

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

**4. 클러스터를 시작합니다.**

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

**5. 출력을 확인합니다.**

- HeadNode 구성에 사용자 지정 작업을 추가한 경우 헤드 노드에 로그인하고 다음 명령을 실행하여 /var/log/cfn-init.log에 있는 cfn-init.log 파일을 확인합니다.

```
$ less /var/log/cfn-init.log
2021-09-03 10:43:54,588 [DEBUG] Command run
postinstall output: The script has 3 arguments
arg: R
arg: curl
arg: wget
```

```
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

- SlurmQueues 설정에 사용자 지정 작업을 추가한 경우 컴퓨팅 노드의 `/var/log/cloud-init.log`에 있는 `cloud-init.log`를 확인하세요. 이러한 로그를 보는 CloudWatch 데 사용됩니다.

Amazon CloudWatch 콘솔에서 이 두 로그를 모두 볼 수 있습니다. 자세한 정보는 [Amazon CloudWatch Logs와 통합](#)을 참조하세요.

## IMDSv2의 사용자 지정 부트스트랩 스크립트 업데이트 예제

다음 예제에서는 IMDSv1에서 사용하던 사용자 지정 부트스트랩 액션 스크립트를 IMDSv2와 함께 사용할 수 있도록 업데이트합니다. IMDSv1 스크립트는 아마존 EC2 인스턴스 AMI ID 메타데이터를 검색합니다.

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

다음은 IMDSv2와 호환되도록 수정된 사용자 지정 부트스트랩 작업 스크립트를 보여줍니다.

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

자세한 설명은 [Linux 인스턴스를 위한 Amazon EC2 사용자 가이드](#)의 인스턴스 메타데이터 검색을 참조하세요.

## IMDSv1의 구성 업데이트 예제

다음은 버전 3.7.0 이상을 사용할 때 IMDSv1을 지원하는 클러스터 구성의 예입니다. AWS ParallelCluster

```

Region: us-east-1
Imds:
  ImdsSupport: v1.0
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh
    KeyName: key-name
  CustomActions:
    OnNodeConfigured:
      Script: Script-path
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CustomActions:
        OnNodeConfigured:
          Script: Script-path
      ComputeResources:
        - Name: t2micro
          Instances:
            - InstanceType: t2.micro
          MinCount: 11
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890

```

## Amazon S3 작업

AWS ParallelCluster 구성에서 [HeadNode/Iam/S3Access](#) 및 [Scheduling/SlurmQueues/-Name/Iam/S3Access](#) 파라미터를 통해 Amazon S3에 대한 AWS ParallelCluster의 액세스를 구성할 수 있습니다.

### 예제

다음 예는 *firstbucket/read\_only/*의 모든 객체에 대한 읽기 전용 액세스와 *secondbucket/read\_and\_write/*의 모든 객체에 대한 읽기/쓰기 액세스를 구성합니다.

```

...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: firstbucket
        KeyName: read_only/*
        EnableWriteAccess: false
      - BucketName: secondbucket
        KeyName: read_and_write/*
        EnableWriteAccess: true
  ...

```

다음 예에서는 계정의 모든 버킷(\*)에 있는 *read\_only/* 폴더의 모든 객체에 대한 읽기 전용 액세스를 구성합니다.

```

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

```

마지막 예에서는 계정의 모든 버킷 및 객체에 대한 읽기 전용 액세스를 구성합니다.

```

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

```

## 스팟 인스턴스 작업

AWS ParallelCluster 클러스터 구성 파일에서 [SlurmQueuesAwsBatchQueues/CapacityType](#) 또는 [CapacityType/](#)를 SPOT 설정한 경우 스팟 인스턴스를 사용합니다. 스팟 인스턴스는 온디맨드 인스턴



스보다 비용 효율적이지만 중단될 수 있습니다. 또한 Amazon EC2가 스팟 인스턴스를 중지 또는 종료하기 2분 전에 경고하는 스팟 인스턴스 종료 공지를 활용할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [스팟 인스턴스 종단을 참조하십시오](#). 스팟 인스턴스의 [AwsBatchQueues](#) 작동 방식을 알아보려면 AWS Batch 사용 설명서의 [컴퓨팅 리소스](#)를 참조하세요.

AWS ParallelCluster 구성된 스케줄러는 온디맨드 인스턴스가 있는 대기열의 컴퓨팅 리소스에 작업을 할당하는 것과 같은 방식으로 스팟 인스턴스가 있는 대기열의 컴퓨팅 리소스에 작업을 할당합니다.

스팟 인스턴스를 사용하는 경우 계정에 서비스 연결 역할이 있어야 합니다.

AWSServiceRoleForEC2Spot 를 사용하여 계정에서 이 역할을 생성하려면 다음 명령을 실행합니다.

AWS CLI

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

자세한 내용은 Amazon EC2 [사용 설명서의 스팟 인스턴스 요청에 대한 서비스 연결 역할을 참조하십시오](#).

다음 섹션에서는 [SlurmQueues](#)를 사용할 경우 스팟 인스턴스가 중단될 수 있는 세 가지 시나리오를 설명합니다.

## 시나리오 1: 실행 중인 작업이 없는 스팟 인스턴스가 중단됨

이러한 중단이 발생하면 스케줄러 대기열에 추가 인스턴스가 필요한 보류 중인 작업이 있거나 활성 인스턴스 수가 //보다 적은 경우 인스턴스 교체를 AWS ParallelCluster 시도합니다.

[SlurmQueuesComputeResourcesMinCount](#) 새 인스턴스를 AWS ParallelCluster 프로비저닝할 수 없는 경우 새 인스턴스에 대한 요청이 주기적으로 반복됩니다.

## 시나리오 2: 단일 노드 작업을 실행하는 스팟 인스턴스가 중단됨

작업이 실패하고 상태 코드가 NODE\_FAIL로 표시되고 작업이 대기열에 추가됩니다(작업 제출 시 --no-requeue를 지정하지 않은 경우). 노드가 정적 노드인 경우 해당 노드가 교체됩니다. 노드가 동적 노드인 경우 노드가 종료되고 재설정됩니다. --no-requeue파라미터 포함에 대한 sbatch 자세한 내용은 Slurm 설명서를 참조하십시오 [sbatch](#).

## 시나리오 3: 다중 노드 작업을 실행하는 스팟 인스턴스가 중단됨

작업이 실패하고 상태 코드가 NODE\_FAIL로 표시되고 작업이 대기열에 추가됩니다(작업 제출 시 --no-requeue를 지정하지 않은 경우). 노드가 정적 노드인 경우 해당 노드가 교체됩니다. 노드가 동

적 노드인 경우 노드가 종료되고 재설정됩니다. 종료된 작업을 실행 중이었던 다른 노드는 구성된 [SlurmSettings/ScaledownIdleTime](#) 시간이 경과한 후에 스케일 다운될 수 있습니다.

스팟 인스턴스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [스팟 인스턴스](#)를 참조하십시오.

## 에서 지원하는 스케줄러 AWS ParallelCluster

에서 지원하는 스케줄러 AWS ParallelCluster

AWS ParallelCluster 설정을 사용하여 설정된 지원 Slurm 및 AWS Batch 스케줄러 [Scheduler](#)

주제

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

## Slurm Workload Manager (**slurm**)

### 클러스터 용량 크기 및 업데이트

클러스터의 용량은 클러스터가 확장할 수 있는 컴퓨팅 노드의 수로 정의됩니다. 컴퓨팅 노드는 AWS ParallelCluster (Scheduling/SlurmQueues/[ComputeResources](#)) 구성의 컴퓨팅 리소스 내에 정의된 Amazon EC2 인스턴스에 의해 지원되며 파티션에 1:1 로 (Scheduling/[SlurmQueues](#)) 매핑되는 대기열로 구성됩니다. Slurm

[컴퓨팅 리소스 내에서 클러스터에서 항상 실행되어야 하는 최소 컴퓨팅 노드 \(인스턴스\) 수 \(\)](#)와 [컴퓨팅 리소스가 \(3MinCount\) 까지 확장할 수 있는 최대 인스턴스 수를 구성할 수 있습니다. MaxCount](#)

클러스터 생성 시 또는 클러스터 업데이트 시 클러스터에 정의된 각 컴퓨팅 리소스 (Scheduling/SlurmQueues/[ComputeResources](#))에 MinCount 대해 구성된 수만큼 Amazon EC2 인스턴스를 AWS ParallelCluster 시작합니다. 클러스터의 컴퓨팅 리소스에 필요한 최소한의 노드를 포함하도록 시작된 인스턴스를 정적 노드라고 합니다. 정적 노드는 일단 시작되면 클러스터에서 영구 유지되며 특정 이벤트나 상황이 발생하지 않는 한 시스템에서 종료되지 않습니다. 이러한 이벤트에는 Amazon EC2 상태 점검 실패, Slurm 노드 상태가 DRAIN Slurm 또는 DOWN으로 변경되는 경우가 포함됩니다.

클러스터의 증가된 부하를 1 처리하기 위해 '**MaxCount - MinCount**' 온디맨드로 시작되는 **MaxCount ~마이너스 MinCount** 범위의 Amazon EC2 인스턴스를 동적 노드라고 합니다. 그 특성은 일시적이며 보류 중인 작업을 처리하기 위해 시작되며 클러스터 Scheduling/

SlurmSettings/[ScaledownIdleTime](#) 구성에서 정의한 기간 (기본값: 10분) 동안 유휴 상태를 유지하면 종료됩니다.

정적 노드와 동적 노드는 다음 이름 지정 스키마를 준수합니다.

- 정적 노드는 다음과 같습니다 <Queue/Name>-st-<ComputeResource/Name>-<num>. <num> = 1..ComputeResource/MinCount
- 동적 노드, <Queue/Name>-dy-<ComputeResource/Name>-<num> 위치 <num> = 1.. (ComputeResource/MaxCount - ComputeResource/MinCount)

예를 들어 다음과 같은 AWS ParallelCluster 구성이 제공됩니다.

```
Scheduling:
  Scheduler: Slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

예 다음과 같은 노드가 정의됩니다. Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up      infinite  100    idle queue1-st-c5xlarge-[1-100]
```

컴퓨팅 리소스에 리소스가 MinCount == MaxCount 있는 경우 해당하는 모든 컴퓨팅 노드는 정적이며 클러스터 생성/업데이트 시 모든 인스턴스가 시작되고 계속 실행됩니다. 예:

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
```

```

ComputeResources:
  - Name: c5xlarge
    Instances:
      - InstanceType: c5.xlarge
    MinCount: 100
    MaxCount: 100

```

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite   100   idle queue1-st-c5xlarge-[1-100]

```

## 클러스터 용량 업데이트

클러스터 용량 업데이트에는 대기열 추가 또는 제거, 컴퓨팅 리소스, 컴퓨팅 리소스 변경이 포함됩니다. MinCount/MaxCount AWS ParallelCluster 버전 3.9.0부터 대기열 크기를 줄이려면 클러스터 업데이트를 실행하기 전에 컴퓨팅 풀릿을 중지하거나 TERMINATE로 [QueueUpdateStrategy](#) 설정해야 합니다. 다음과 같은 경우에는 컴퓨팅 풀릿을 중지하거나 TERMINATE로 설정할 [QueueUpdateStrategy](#) 필요가 없습니다.

- 스케줄링에 새 대기열 추가/ [SlurmQueues](#)
- 대기열에 새 컴퓨팅 리소스 추가 Scheduling/SlurmQueues/[ComputeResources](#)
- 컴퓨팅 리소스 증가 [MaxCount](#)
- 컴퓨팅 리소스 증가 MaxCount 및 동일 컴퓨팅 리소스 최소 양 증가 MinCount

## 고려 사항 및 제한

이 섹션은 클러스터 용량 크기를 조정할 때 고려해야 하는 중요한 요소, 제약 조건 또는 제한 사항을 간략하게 설명하기 위한 것입니다.

- 이름이 있는 Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues 모든 컴퓨팅 노드에서 대기열을 제거하면 정적 및 동적 이름이 <Queue/Name>-\* Slurm 구성에서 제거되고 해당 Amazon EC2 인스턴스가 종료됩니다.
- Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues->

[ComputeResources](#) ComputeResources 대기열에서 컴퓨팅 리소스를 제거하면 정적 및 동적 이름을 `<Queue/Name>-*-<ComputeResource/Name>-*` 가진 모든 컴퓨팅 노드가 Slurm 구성에서 제거되고 해당 Amazon EC2 인스턴스가 종료됩니다.

컴퓨팅 리소스의 MinCount 파라미터를 변경할 때 두 가지 시나리오를 구분할 수 있습니다.

MaxCount 하나는 같게 유지되고 MinCount (정적 용량만 해당) 이면 MinCount (정적 용량과 동적 용량이 혼합된 경우 MaxCount) 보다 크다는 것입니다.

정적 노드에서만 용량이 변경됩니다.

- 증가 MinCount (및 MaxCount) 시 정적 노드의 수를 새로운 값까지 확장하여 클러스터를 구성하고 시스템은 새로운 필수 정적 용량을 충족하기 위해 Amazon EC2 인스턴스를 계속 시작하려고 시도합니다. `MinCount == MaxCount` `MinCount < Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>`
- 양 N을 줄이면 MinCount (및 MaxCount) 마지막 N개의 정적 노드를 `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>` 제거하여 클러스터가 구성되고 시스템은 해당 Amazon EC2 인스턴스를 종료합니다. `MinCount == MaxCount`
- 초기 상태 `MinCount = MaxCount = 100`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite   100    idle  queue1-st-c5xlarge-[1-100]
```

- MinCount 및 -30 에서 업데이트 MaxCount: `MinCount = MaxCount = 70`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite    70    idle  queue1-st-c5xlarge-[1-70]
```

## 혼합 노드의 용량 변경

N만큼 MinCount 증가하면 `MinCount < MaxCount` (변경되지 않은 상태로 유지된다고 가정 MaxCount) 정적 노드 수가 새 값인 `MinCount (old_MinCount + N)` :까지 확장되어 클러스터

가 `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` 구성되고 시스템은 새로운 필수 정적 용량을 충족하기 위해 Amazon EC2 인스턴스를 계속 시작하려고 시도합니다. 또한 컴퓨팅 리소스의 MaxCount 용량을 유지하기 위해 마지막 N개의 동적 노드를 제거하여 클러스터 구성을 업데이트합니다 `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]`. 그러면 시스템이 해당 Amazon EC2 인스턴스를 종료합니다.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- +30으로 업데이트 MinCount : MinCount = 130 (MaxCount = 150)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

같은 양 N을 MinCount 늘리면 정적 노드 수를 새 값인 MinCount (`old_MinCount + N`) `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` :까지 확장하여 클러스터를 구성하고 시스템은 새로운 필수 정적 용량을 충족하기 위해 Amazon EC2 인스턴스를 계속 시작하려고 시도합니다. MinCount < MaxCount MaxCount 또한 새 노드를 적용하기 위해 동적 노드 수는 변경되지 않습니다.

MaxCount USD 상당.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
```

```
queue1*      up    infinite    100   idle queue1-st-c5xlarge-[1-100]
```

- +30으로 업데이트 MinCount : MinCount = 130 (MaxCount = 180)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up     infinite   20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up     infinite  130    idle queue1-st-c5xlarge-[1-130]
```

N개의 양을 MinCount 줄이면 (변경되지 않은 상태로 유지된다고 가정MaxCount) 마지막 N개의 정적 노드 정적 노드를 <Queue/Name>-st-<ComputeResource/Name>-[<old\_MinCount - N>...<old\_MinCount> 제거하여 클러스터를 구성하고 시스템은 해당 Amazon EC2 인스턴스를 종료합니다. MinCount < MaxCount 또한 컴퓨팅 리소스의 MaxCount 용량을 늘리기 위해 동적 노드 수를 늘려 격차를 메우는 MaxCount - new\_MinCount: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new\_MinCount>] 방식으로 클러스터 구성을 업데이트합니다. 이 경우 동적 노드는 동적 노드이므로 스케줄러가 새 노드에 보류 중인 작업이 없는 한 새 Amazon EC2 인스턴스가 시작되지 않습니다.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up     infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up     infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 업데이트 -30 켜기 MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up     infinite   80    idle~ queue1-dy-c5xlarge-[1-80]
queue1*   up     infinite   70    idle queue1-st-c5xlarge-[1-70]
```

N개를 MinCount 줄이면 마지막 N개의 정적 노드를 <Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount - N>...<oldMinCount>] 제거하여 클러스터를 구성하고 시스템은 해당 Amazon EC2 인스턴스를 종료합니다. MinCount < MaxCount MaxCount

또한 새 MaxCount 값을 반영하기 위해 동적 노드의 수를 변경하지 않을 예정입니다.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 업데이트 -30 켜기 MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   80    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite   70    idle queue1-st-c5xlarge-[1-70]
```

N개의 양을 줄이면 MaxCount (변경되지 않은 상태로 유지된다고 가정MinCount) 마지막 N개의 동적 노드를 <Queue/Name>-dy-<ComputeResource/Name>-<old\_MaxCount - N...<oldMaxCount>] 제거하여 클러스터를 구성하고 시스템은 해당 Amazon EC2 인스턴스가 실행 중일 때 해당 Amazon EC2 인스턴스를 종료합니다. 정적 노드에는 영향이 없을 것으로 예상됩니다. MinCount < MaxCount

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
```



```
queue1*      up    infinite    100    idle queue1-st-c5xlarge-[1-100]
```

- 업데이트 -30 켜기 MaxCount : MinCount = 100 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up     infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up     infinite  100    idle queue1-st-c5xlarge-[1-100]
```

## 작업에 미치는 영향

노드가 제거되고 Amazon EC2 인스턴스가 종료되는 모든 경우, 작업 요구 사항을 충족하는 다른 노드가 없는 한 제거된 노드에서 실행 중인 sbatch 작업은 대기열에 다시 추가됩니다. 마지막 경우에는 작업이 NODE\_FAIL 상태로 실패하고 대기열에서 사라집니다. 이 경우 수동으로 다시 제출해야 합니다.

클러스터 크기 조정 업데이트를 수행하려는 경우 계획된 업데이트 중에 제거될 노드에서 작업이 실행되지 않도록 할 수 있습니다. 이는 유지 관리 시 제거될 노드를 설정하여 가능합니다. 유지 관리 중인 노드를 설정해도 결국 해당 노드에서 이미 실행 중인 작업에는 영향을 미치지 않는다는 점에 유의하십시오.

계획된 클러스터 크기 조정 업데이트를 통해 해당 노드를 queue-st-computeresource-[9-10] 제거한다고 가정해 보겠습니다.] 다음 명령으로 Slurm 예약을 생성할 수 있습니다.

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queue-st-
computeresource-[9-10]
```

그러면 maint\_for\_update 노드에 이름이 지정된 Slurm 예약이 생성됩니다 queue-st-computeresource-[9-10]. 예약이 생성된 시점부터 더 이상 노드에 작업이 실행될 수 없습니다 queue-st-computeresource-[9-10]. 예약을 한다고 해서 결국 노드에 작업이 할당되는 것을 막지는 못한다는 점에 queue-st-computeresource-[9-10] 유의하십시오.

클러스터 크기 조정 업데이트 후 크기 조정 업데이트 중에 제거된 노드에만 Slurm 예약이 설정된 경우 유지 관리 예약은 자동으로 삭제됩니다. 대신 클러스터 크기 조정 업데이트 후에도 여전히 존재하는 노

드에 대한 Slurm 예약을 생성한 경우, 크기 조정 업데이트를 수행한 후 다음 명령을 사용하여 노드의 유지 관리 예약을 제거할 수 있습니다.

```
sudo -i scontrol delete ReservationName=maint_for_update
```

Slurm [예약에 대한 자세한 내용은 여기에서 공식 SchedMD 문서를 참조하십시오.](#)

## 용량 변경에 대한 클러스터 업데이트 프로세스

스케줄러 구성이 변경되면 클러스터 업데이트 프로세스 중에 다음 단계가 실행됩니다.

- 중지 AWS ParallelCluster clustermgtd (supervisorctl stop clustermgtd)
- 구성에서 업데이트된 Slurm 파티션 AWS ParallelCluster 구성 생성
- 재시작 slurmctld (Chef 서비스 레시피를 통해 완료)
- slurmctld 상태 확인 (systemctl is-active --quiet slurmctld.service)
- 구성 다시 불러오기 Slurm (scontrol reconfigure)
- clustermgtd (supervisorctl start clustermgtd) 시작

Slurm에 대한 자세한 내용은 <https://slurm.schedmd.com>을 참조하세요. 다운로드에 대해서는 <https://github.com/SchedMD/slurm/tags>를 참조하세요. 소스 코드는 <https://github.com/SchedMD/slurm>을 참조하세요.

AWS ParallelCluster 버전 (들)	지원되는 Slurm 버전
3.9.2, 3.9.3, 3.10.0	23.11.7
3.9.0, 3.9.1	23.11.4
3.8.0	23.02.7
3.7.2	23.02.6
3.7.1	23.02.5
3.7.0	23.02.4
3.6.0, 3.6.1	23.02.2

AWS ParallelCluster 버전 (들)	지원되는 Slurm 버전
3.5.0, 3.5.1	22.05.8
3.4.0, 3.4.1	22.05.7
3.3.0, 3.3.1	22.05.5
3.1.4, 3.1.5, 3.2.0, 3.2.1	21.08.8-2
3.1.2, 3.1.3	21.08.6
3.1.1	21.08.5
3.0.0	20.11.8

## 주제

- [다중 대기열 구성](#)
- [다중 대기열 모드를 위한 Slurm 가이드](#)
- [Slurm 클러스터 보호 모드](#)
- [Slurm 클러스터 빠른 용량 부족 장애 조치](#)
- [Slurm 메모리 기반 스케줄링](#)
- [Slurm을 사용하여 여러 인스턴스 유형 할당](#)
- [동적 노드의 클러스터 스케일링](#)
- [Slurm회계 담당자: AWS ParallelCluster](#)
- [Slurm 구성 사용자 지정](#)
- [Slurmprolog 및 epilog](#)
- [클러스터 용량 크기 및 업데이트](#)

## 다중 대기열 구성

### 다중 대기열 구성

AWS ParallelCluster 버전 3에서는 틀 로 설정하고 구성 파일에 둘 이상의 대기열을 지정하여 여러 대기열을 구성할 수 있습니다. [SchedulersslurmSlurmQueues](#) 이 모드에서는 구성 파일의 [ComputeResources](#) 섹션에 지정된 컴퓨팅 노드에 여러 인스턴스 유형이 공존합니다. 인스턴스 유형

이 다른 [ComputeResources](#)은 [SlurmQueues](#)의 필요에 따라 스케일 업 또는 스케일 다운될 수 있습니다.

### 클러스터 대기열 및 컴퓨팅 리소스 할당량

Resource	할당량
<a href="#">Slurm queues</a>	클러스터당 50개의 대기열
<a href="#">Compute resources</a>	대기열당 50개의 컴퓨팅 리소스 클러스터당 50개의 컴퓨팅 리소스

### 노드 수

[ComputeResources](#) 대기열의 각 컴퓨팅 리소스에는 고유한 [Name](#), [InstanceType](#), [MinCount](#) 및 [MaxCount](#)가 있어야 합니다. [MinCount](#) 및 [MaxCount](#)는 [ComputeResources](#) 대기열의 컴퓨팅 리소스 인스턴스 범위를 정의하는 기본값이 있어야 합니다. [MinCount](#) 및 [MaxCount](#)에 대해 고유한 값을 지정할 수도 있습니다. [ComputeResources](#)의 각 컴퓨팅 리소스는 1에서 [MinCount](#) 값 사이의 번호가 매겨진 정적 노드와 [MinCount](#) 값에서 [MaxCount](#) 값 사이의 번호가 매겨진 동적 노드로 구성됩니다.

### 구성의 예제

다음은 클러스터 구성 파일의 [일정 예약](#) 섹션 예제입니다. 이 구성에는 이름이 queue1 및 queue2인 대기열이 두 개 있으며 각 대기열에는 [MaxCount](#)가 지정된 [ComputeResources](#)가 있습니다.

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

```
MaxCount: 5
Name: c5xlarge
```

## Hostnames

컴퓨팅 플릿으로 시작되는 인스턴스는 동적으로 할당됩니다. 호스트 이름은 각 노드에 대해 생성됩니다. 기본적으로 AWS ParallelCluster 다음과 같은 형식의 호스트 이름을 사용합니다.

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

- \$QUEUE은 대기열의 이름입니다. 예를 들어 [SlurmQueues](#) 섹션에 “queue-name”으로 설정된 [Name](#) 항목이 있는 경우 “\$QUEUE”는 “queue-name”입니다.
- \$STATDYN은 정적 노드에는 st 또는 동적 노드에는 dy입니다.
- \$COMPUTE\_RESOURCE은 이 노드에 대응하는 [ComputeResources](#) 컴퓨팅 리소스의 [Name](#)입니다.
- \$NODENUM은 노드의 번호입니다. \$NODENUM은 정적 노드의 경우 1과 [MinCount](#)의 값 사이, 동적 노드의 경우 1과 [MaxCount](#)~[MinCount](#) 사이입니다.

위의 예제 구성 파일에서 queue1와 컴퓨팅 리소스 c5xlarge의 특정 노드는 호스트 이름 queue1-dy-c5xlarge-1을 가집니다.

호스트 이름과 FQDN(정규화된 도메인 이름)은 모두 Amazon Route 53 호스팅 영역을 사용하여 생성됩니다. FQDN은 \$HOSTNAME.\$CLUSTERNAME.pcluster입니다. 여기서 \$CLUSTERNAME는 클러스터의 이름입니다.

Slurm노드 이름에도 동일한 형식이 사용된다는 점에 유의하십시오.

사용자는 에서 사용하는 기본 호스트 이름 형식 대신 컴퓨팅 노드를 구동하는 인스턴스의 기본 Amazon EC2 호스트 이름을 사용하도록 선택할 수 있습니다. AWS ParallelCluster [UseEc2Hostnames](#) 파라미터를 true로 설정하여 이 작업을 수행할 수 있습니다. 하지만 Slurm 노드 이름은 계속해서 기본 AWS ParallelCluster 형식을 사용합니다.

## 다중 대기열 모드를 위한 Slurm 가이드

여기서는 큐 (파티션) 노드를 Slurm 관리하는 방법 AWS ParallelCluster 및 큐와 노드 상태를 모니터링하는 방법을 알아볼 수 있습니다.

### 개요

확장 아키텍처는 Slurm의 [클라우드 스케줄링 가이드](#) 및 절전 플러그인을 기반으로 합니다. 절전 플러그인에 대한 자세한 내용은 [Slurm 절전 가이드](#)를 참조하세요. 아키텍처에서 클러스터에 사용할 수 있는 리소스는 일반적으로 Slurm 구성에서 클라우드 노드로 미리 정의됩니다.

## 클라우드 노드 수명 주기

클라우드 노드는 수명 주기 내내 POWER\_SAVING, POWER\_UP(pow\_up), ALLOCATED(alloc), POWER\_DOWN(pow\_dn) 상태 중 여러 개 또는 전부로 전환됩니다. 경우에 따라 클라우드 노드가 OFFLINE 상태로 전환될 수 있습니다. 다음 목록은 클라우드 노드 수명 주기에서 이러한 상태의 여러 측면을 자세히 설명합니다.

- POWER\_SAVING 상태의 노드는 `sinfo`에서 ~ 접미사(예: `idle~`)와 함께 표시됩니다. 이 상태에서는 노드를 지원하는 EC2 인스턴스가 없습니다. 하지만 Slurm은 여전히 노드에 작업을 할당할 수 있습니다.
- POWER\_UP 상태로 전환되는 노드는 `sinfo`에서 # 접미사(예: `idle#`)와 함께 표시됩니다. Slurm이 POWER\_SAVING 상태의 노드에 작업을 할당하면 노드는 자동으로 POWER\_UP 상태로 전환됩니다.

또는 다음 명령을 사용하여 `su` 루트 사용자로서 노드를 수동으로 POWER\_UP 상태로 전환할 수 있습니다.

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

이 단계에서는 ResumeProgram이 호출되고, EC2 인스턴스가 시작 및 구성되고, 노드가 POWER\_UP 상태로 전환됩니다.

- 현재 사용할 수 있는 노드는 `sinfo`에서 접미사(예: `idle`)가 없는 것으로 나타납니다. 노드가 설정되고 클러스터에 가입되면 작업을 실행할 수 있게 됩니다. 이 단계에서는 노드가 적절하게 구성되고 사용할 준비가 됩니다.

일반적으로 Amazon EC2 인스턴스 수는 사용 가능한 노드 수와 같게 하는 것이 좋습니다. 대부분의 경우 정적 노드는 클러스터를 생성한 후에 사용할 수 있습니다.

- POWER\_DOWN 상태로 전환되는 노드는 `sinfo`에서 % 접미사(예: `idle%`)와 함께 표시됩니다. 동적 노드는 `ScaledownIdletime` 이후 POWER\_DOWN 상태가 자동으로 입력됩니다. 반면 정적 노드는 대부분의 경우 전원이 꺼지지 않습니다. 하지만 다음 명령을 사용하여 수동으로 `su` 루트 사용자로 노드를 POWER\_DOWN 상태에 배치할 수 있습니다.

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

이 상태에서는 노드와 연결된 인스턴스가 종료되고 노드는 다시 해당 POWER\_SAVING 상태로 설정되며 `ScaledownIdletime` 이후에 사용할 수 있습니다.

[ScaledownIdletime](#) 설정이 Slurm 구성 `SuspendTimeout` 설정에 저장됩니다.

- 오프라인 상태인 노드는 `sinfo`에서 \* 접미사(예:down\*)와 함께 나타납니다. Slurm 컨트롤러가 노드에 연결할 수 없거나 정적 노드가 비활성화되고 지원 인스턴스가 종료되면 노드는 오프라인 상태가 됩니다.

다음 `sinfo` 예제에 표시된 노드 상태를 고려해 보세요.

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

`spot-st-spotcompute2-[1-2]` 및 `efa-st-efacompute1-1` 노드에는 이미 백업 인스턴스가 설정되어 있고 사용할 수 있습니다. `ondemand-dy-ondemandcompute1-[1-2]` 노드는 현재 `POWER_UP` 상태이며 몇 분 이내에 사용할 수 있습니다. `gpu-dy-gpucompute1-1` 노드는 현재 `POWER_DOWN` 상태이고 [ScaledownIdletime](#) 이후 `POWER_SAVING` 상태로 전환됩니다(기본값 10 분).

다른 모든 노드는 이를 지원하는 EC2 인스턴스가 없는 `POWER_SAVING` 상태입니다.

사용 가능한 노드로 작업하기

사용 가능한 노드는 Amazon EC2 인스턴스에서 지원됩니다. 기본적으로 노드 이름을 사용하여 인스턴스에 직접 SSH로 연결할 수 있습니다(예: `ssh efa-st-efacompute1-1`). 다음 명령을 사용하여 인스턴스의 프라이빗 IP 주소를 검색할 수 있습니다.

```
$ scontrol show nodes nodename
```

반환된 `NodeAddr` 필드에서 IP 주소를 확인합니다.

사용할 수 없는 노드의 경우 `NodeAddr` 필드는 실행 중인 Amazon EC2 인스턴스를 가리키면 안 됩니다. 그보다는 노드 이름과 동일해야 합니다.

## 작업 상태 및 제출

대부분의 경우 제출된 작업은 시스템의 노드에 즉시 할당되거나, 모든 노드가 할당되면 보류 상태로 전환됩니다.

작업에 할당된 노드에 특정 POWER\_SAVING 상태의 노드가 포함된 경우 작업은 CF 또는 CONFIGURING 상태로 시작됩니다. 이때 작업은 POWER\_SAVING 상태의 노드가 해당 POWER\_UP 상태로 전환되어 사용 가능한 상태가 될 때까지 기다립니다.

작업에 할당된 모든 노드를 사용할 수 있게 되면 작업은 RUNNING(R) 상태가 됩니다.

기본적으로 모든 작업은 기본 대기열(Slurm에서 파티션이라고 함)에 제출됩니다. 이는 대기열 이름 뒤에 \* 접미사가 붙는 것으로 표시됩니다. -p 작업 제출 옵션을 사용하여 대기열을 선택할 수 있습니다.

모든 노드는 작업 제출 명령에 사용할 수 있는 다음 기능으로 구성됩니다.

- 인스턴스 유형(예: c5.xlarge)
- 노드 유형(dynamic 또는 static)

다음 명령을 사용하여 특정 노드의 특성을 확인할 수 있습니다.

```
$ scontrol show nodes nodename
```

반환할 때는 AvailableFeatures 목록을 확인하세요.

sinfo 명령을 실행하여 확인할 수 있는 클러스터의 초기 상태를 고려해 보세요.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4    idle~ efa-dy-efacompute1-[1-4]
efa        up    infinite   1    idle  efa-st-efacompute1-1
gpu        up    infinite  10    idle~ gpu-dy-gpucompute1-[1-10]
ondemand   up    infinite  20    idle~ ondemand-dy-ondemandcompute1-
[1-10],ondemand-dy-ondemandcompute2-[1-10]
spot*      up    infinite  13    idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*      up    infinite   2    idle  spot-st-spotcompute2-[1-2]
```

spot이 기본 대기열이라는 점에 유의하세요. \* 접미사로 표시됩니다.



기본 대기열(spot)에 있는 정적 노드 하나에 작업을 제출합니다.

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

EFA 대기열에 있는 한 동적 노드에 작업을 제출합니다.

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

ondemand 대기열에 있는 8개의 c5.2xlarge 노드와 2개의 t2.xlarge 노드에 작업을 제출하세요.

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

gpu 대기열에 있는 GPU 노드 하나에 작업을 제출하세요.

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

squeue 명령을 사용하여 작업 상태를 살펴보세요.

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

작업 7 및 8(spot 및 efa 대기열에 있음)은 이미 실행 중입니다(R). 작업 12와 13은 여전히 구성 중이며(CF), 아마도 인스턴스를 사용할 수 있을 때까지 기다리고 있을 것입니다.

```
# Nodes states corresponds to state of running jobs
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
efa up infinite 3 idle~ efa-dy-efacompute1-[2-4]
efa up infinite 1 mix efa-dy-efacompute1-1
efa up infinite 1 idle efa-st-efacompute1-1
gpu up infinite 1 mix~ gpu-dy-gpucompute1-1
gpu up infinite 9 idle~ gpu-dy-gpucompute1-[2-10]
ondemand up infinite 10 mix# ondemand-dy-ondemandcompute1-[1-8],ondemand-
dy-ondemandcompute2-[1-2]
```

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

## 노드 상태 및 특성

대부분의 경우 노드 상태는 이 주제 앞부분에서 설명한 클라우드 노드 수명 주기의 특정 프로세스에 AWS ParallelCluster 따라 완전히 관리됩니다.

그러나, DOWN 및 DRAINED 상태의 비정상 노드와 비정상 지원 인스턴스가 있는 노드를 대체하거나 AWS ParallelCluster 종료하기도 합니다. 자세한 정보는 [clustermgtd](#)을 참조하세요.

## 파티션 상태

AWS ParallelCluster 다음과 같은 파티션 상태를 지원합니다. Slurm 파티션은 AWS ParallelCluster의 대기열입니다.

- UP: 파티션이 활성 상태임을 나타냅니다. 이것은 파티션의 기본 상태입니다. 이 상태에서는 파티션의 모든 노드가 활성화되어 사용할 수 있습니다.
- INACTIVE: 파티션이 비활성 상태임을 나타냅니다. 이 상태에서는 비활성 파티션의 노드를 지원하는 모든 인스턴스가 종료됩니다. 비활성 파티션의 노드에 대해서는 새 인스턴스가 시작되지 않습니다.

pcluster는 update-compute-fleet

- 컴퓨팅 플릿 중지 - 다음 명령을 실행하면 모든 파티션이 INACTIVE 상태로 전환되고 AWS ParallelCluster 프로세스는 파티션을 INACTIVE 상태로 유지합니다.

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

- 컴퓨팅 플릿 시작 - 다음 명령을 실행하면 모든 파티션이 처음에 UP 상태로 전환됩니다. 하지만 AWS ParallelCluster 프로세스는 파티션을 일정한 상태로 유지하지 않습니다. UP 파티션 상태를 수동으로 변경해야 합니다. 몇 분 후 모든 고정 노드를 사용할 수 있습니다. 참고로 파티션을 UP으로 설정해도 동적 용량은 증가하지 않습니다.

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

```
--region eu-west-1 --status START_REQUESTED
```

update-compute-fleet가 실행되면 pcluster describe-compute-fleet 명령을 실행하고 Status를 확인하여 클러스터의 상태를 확인할 수 있습니다. 다음과 같은 잠재적인 상태가 있습니다.

- STOP\_REQUESTED: 컴퓨팅 플릿 중지 요청이 클러스터로 전송됩니다.
- STOPPING: pcluster 프로세스에서 현재 컴퓨팅 플릿을 중지하고 있습니다.
- STOPPED: pcluster 프로세스가 중지 프로세스를 완료했고, 모든 파티션이 INACTIVE 상태에 있으며, 모든 컴퓨팅 인스턴스가 종료되었습니다.
- START\_REQUESTED: 컴퓨팅 플릿 시작 요청이 클러스터로 전송됩니다.
- STARTING: pcluster 프로세스가 현재 클러스터를 시작하는 중입니다.
- RUNNING: pcluster 프로세스가 시작 프로세스를 마쳤고, 모든 파티션이 UP 상태에 있으며, 몇 분 후에 정적 노드를 사용할 수 있습니다.
- PROTECTED: 이 상태는 일부 파티션에서 일관된 부트스트랩 오류가 발생했음을 나타냅니다. 영향을 받는 파티션은 비활성 상태입니다. 문제를 조사한 다음 update-compute-fleet 실행하여 플릿을 다시 활성화하세요.

## 대기열 수동 제어

클러스터의 노드 또는 대기열(Slurm 파티션이라고 함)을 수동으로 제어해야 하는 경우도 있습니다. scontrol 명령을 사용하여 다음과 같은 일반적인 절차를 통해 클러스터의 노드를 관리할 수 있습니다.

- **POWER\_SAVING** 상태에 있는 동적 노드의 전원을 켜세요.

su 루트 사용자로 명령을 실행합니다.

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

특정 수의 노드를 요청하는 플레이스홀더 sleep 1 작업을 제출한 다음 Slurm에 의존하여 필요한 수의 노드에 전원을 공급할 수도 있습니다.

- [ScaledownIdletime](#) 전에 먼저 동적 노드의 전원을 끄세요.

해당 명령을 사용하여 su 루트 사용자로서 동적 노드를 DOWN으로 설정하는 것이 좋습니다.

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

AWS ParallelCluster 다운된 동적 노드를 자동으로 종료하고 재설정합니다.

일반적으로 `scontrol update nodename=nodename state=power_down` 명령을 사용하여 노드를 POWER\_DOWN으로 직접 설정하지 않는 것이 좋습니다. AWS ParallelCluster 가 전원 차단 프로세스를 자동으로 처리하기 때문입니다.

- 대기열(파티션)을 비활성화하거나 특정 파티션의 모든 정적 노드를 중지합니다.

해당 명령을 사용하여 su 루트 사용자로서 특정 대기열을 INACTIVE로 설정합니다.

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

이렇게 하면 파티션의 노드를 지원하는 모든 인스턴스가 종료됩니다.

- 대기열(파티션) 활성화

해당 명령을 사용하여 su 루트 사용자로서 특정 대기열을 UP로 설정합니다.

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

## 규모 조정 동작 및 조정

다음은 일반적인 규모 조정 워크플로의 예입니다.

- 스케줄러는 두 개의 노드가 필요한 작업을 수신합니다.
- 스케줄러는 두 노드를 POWER\_UP 상태로 전환하고 노드 이름(예: `queue1-dy-spotcompute1-[1-2]`)을 사용하여 `ResumeProgram`을 호출합니다.
- `ResumeProgram` Amazon EC2 인스턴스 2개를 시작하고 프라이빗 IP 주소와 호스트 이름을 할당하고 대기합니다 `ResumeTimeout` (기본 기간은 30분 후 노드 재설정). `queue1-dy-spotcompute1-[1-2]`
- 인스턴스가 구성되어 클러스터에 들어갑니다. 인스턴스에서 작업이 실행되기 시작합니다.
- 작업이 완료되고 실행이 중지됩니다.
- 구성된 `SuspendTime`이 경과한 후([ScaledownIdletime](#)로 설정되어 있음), 스케줄러는 인스턴스를 POWER\_SAVING 상태로 설정합니다. 그러면 스케줄러가 `queue1-dy-spotcompute1-[1-2]`를 POWER\_DOWN 상태를 설정하고 노드 이름으로 `SuspendProgram`을 호출합니다.
- 두 노드에 대해 `SuspendProgram`이 호출됩니다. 예를 들어 노드는 `SuspendTimeout`[기본 120 초(2분)]을 위해 `idle%`로 남아 있음으로써 POWER\_DOWN 상태로 유지됩니다. 노드의 전원이 꺼

지고 있음을 `clustermgtd`가 감지하면 지원 인스턴스가 종료됩니다. 그런 다음 `queue1-dy-spotcompute1-[1-2]`을 유휴 상태로 전환하고, 향후 작업을 위해 준비를 마칠 수 있도록 사설 IP 주소와 호스트 이름을 재설정합니다.

문제가 발생하여 특정 노드의 인스턴스를 어떤 이유로 시작할 수 없는 경우 다음과 같은 상황이 발생합니다.

- 스케줄러는 두 개의 노드가 필요한 작업을 수신합니다.
- 스케줄러는 두 개의 클라우드 버스팅 노드를 `POWER_UP` 상태로 전환하고 노드 이름을 사용하여 `ResumeProgram`를 호출합니다(예: `queue1-dy-spotcompute1-[1-2]`).
- `ResumeProgram` Amazon EC2 인스턴스 1개만 시작하고 인스턴스 1개로 `queue1-dy-spotcompute1-1` 구성하지만 시작에 실패합니다. `queue1-dy-spotcompute1-2`
- `queue1-dy-spotcompute1-1`는 영향을 받지 않으며 `POWER_UP` 상태에 도달하면 온라인 상태가 됩니다.
- `queue1-dy-spotcompute1-2`은 `POWER_DOWN` 상태로 전환되고 Slurm이 노드 장애를 감지하므로 작업이 자동으로 대기열에 추가됩니다.
- `queue1-dy-spotcompute1-2`은 `SuspendTimeout` 이후에 사용할 수 있습니다[기본 120초(2분)]. 그동안에는 작업이 대기되며 다른 노드에서 실행을 시작할 수 있습니다.
- 장애가 발생하지 않고 사용 가능한 노드에서 작업을 실행할 수 있을 때까지 위 프로세스가 반복됩니다.

필요한 경우 두 가지 타이밍 매개 변수를 조정할 수 있습니다.

- `ResumeTimeout`(기본값은 30분): `ResumeTimeout`은 Slurm가 대기하는 시간을 제어한 후 노드를 다운 상태로 전환합니다.
  - 사전/사후 설치 프로세스가 그렇게 오래 걸린다면 `ResumeTimeout`을 연장하는 것이 유용할 수 있습니다.
  - `ResumeTimeout`은 AWS ParallelCluster 가 문제가 있는 경우 노드를 교체하거나 재설정하기 전에 대기하는 최대 시간이기도 합니다. 시작 또는 설정 중에 오류가 발생하면 컴퓨팅 노드가 자동으로 종료됩니다. AWS ParallelCluster 종료된 인스턴스가 감지되면 프로세스가 노드를 교체합니다.
- `SuspendTimeout`[기본 120초(2분)]: `SuspendTimeout`이 노드를 시스템에 다시 배치하고 다시 사용할 준비가 되는 시간을 제어합니다.
  - `SuspendTimeout`이 짧을수록 노드가 더 빨리 재설정되고, Slurm는 인스턴스를 더 자주 시작하려고 할 수 있습니다.

- SuspendTimeout이 길수록 장애가 발생한 노드의 재설정 속도가 느려집니다. 그러는 동안 Slurm은 다른 노드를 사용하려고 합니다. SuspendTimeout이 몇 분 이상이면 Slurm가 시스템의 모든 노드를 순환하려 합니다. 대규모 시스템(1,000개 이상의 노드)에서는 Slurm가 받는 스트레스를 줄이기 위해 장애가 발생한 작업을 다시 대기열에 추가하려고 할 때 더 긴 SuspendTimeout을 사용하는 것이 유용할 수 있습니다.
- 단, 노드의 지원 인스턴스가 종료되는 데 걸리는 AWS ParallelCluster 대기 시간을 SuspendTimeout 의미하지는 않습니다. POWER\_DOWN 노드의 백업 인스턴스는 즉시 종료됩니다. 종료 프로세스는 일반적으로 몇 분 이내에 완료됩니다. 하지만 이 기간 동안에는 노드가 POWER\_DOWN 상태를 유지하며 스케줄러가 사용할 수 없습니다.

## 아키텍처 로그

다음 목록 키 로그를 포함합니다. Amazon CloudWatch Logs에 사용되는 로그 스트림 이름의 `{hostname}.{instance_id}.{logIdentifier}` 형식은 다음과 같습니다. 여기서 `LogIdentifier#` 로그 이름 뒤에 옵니다.

- 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)

일반적인 문제 및 디버그 방법:

시작, 전원 켜기 또는 클러스터 조인에 실패한 노드

- 동적 노드:
  - ResumeProgram 로그를 확인하여 노드와 함께 ResumeProgram 호출되었는지 확인하세요. 그렇지 않은 경우 slurmctld 로그를 확인하여 Slurm가 노드로 ResumeProgram을 호출하려 했는지 확인하세요. ResumeProgram 권한이 올바르지 않으면 로그가 자동으로 실패할 수 있다는 점에 유의하세요.
  - ResumeProgram가 호출되면 해당 노드에 대한 인스턴스가 시작되었는지 확인하세요. 인스턴스가 시작되지 않은 경우 인스턴스 시작 실패 이유에 대한 명확한 오류 메시지가 표시되어야 합니다.

- 인스턴스가 시작된 경우 부트스트랩 프로세스 중에 문제가 발생했을 수 있습니다. ResumeProgram 로그에서 해당 프라이빗 IP 주소와 인스턴스 ID를 찾고 Logs에서 특정 인스턴스에 해당하는 부트스트랩 로그를 확인합니다. CloudWatch
- 고정 노드:
  - clustermgtd 로그를 확인하여 해당 노드의 인스턴스가 시작되었는지 확인합니다. 인스턴스가 시작되지 않았다면 인스턴스 시작 실패 이유에 대한 명확한 오류가 있을 것입니다.
  - 인스턴스가 시작된 경우 부트스트랩 프로세스에 문제가 있는 것입니다. 로그에서 해당 사설 IP와 인스턴스 ID를 찾고 clustermgtd 로그에서 특정 인스턴스에 해당하는 부트스트랩 로그를 확인합니다. CloudWatch

노드가 여기치 않게 교체되거나 종료되었으며, 노드 장애가 발생했습니다.

- 노드가 여기치 않게 교체/종료됨:
  - 대부분의 경우 clustermgtd가 모든 노드 유지 관리 작업을 처리합니다. clustermgtd가 노드를 교체하거나 종료했는지를 확인하려면 clustermgtd 로그를 확인하세요.
  - clustermgtd가 노드를 교체하거나 종료한 경우 작업 이유를 나타내는 메시지가 표시되어야 합니다. 이유가 스케줄러와 관련된 경우(예: 노드가 DOWN이었음) slurmctld 로그에서 자세한 내용을 확인하세요. Amazon EC2와 관련된 이유인 경우 Amazon CloudWatch 또는 Amazon EC2 콘솔, CLI 또는 SDK와 같은 도구를 사용하여 해당 인스턴스의 상태 또는 로그를 확인하십시오. 예를 들어 인스턴스에 예약된 이벤트가 있는지 또는 Amazon EC2 상태 확인에 실패했는지 확인할 수 있습니다.
  - clustermgtd가 노드를 종료하지 않은 경우 computemgtd가 노드를 종료했는지 또는 EC2가 스팟 인스턴스를 회수하기 위해 인스턴스를 종료했는지 확인하세요.
- 노드 장애:
  - 대부분의 경우 노드에 장애가 발생하면 작업이 자동으로 대기열에 추가됩니다. slurmctld 로그에서 작업이나 노드에 장애가 발생한 이유를 확인하고 거기에서 상황을 평가하세요.

인스턴스 교체 또는 종료 시 실패, 노드 전원 차단 시 실패

- 일반적으로 clustermgtd가 모든 예상 인스턴스 종료 작업을 처리합니다. clustermgtd 로그에서 노드 교체 또는 종료에 실패한 이유를 확인하세요.
- 동적 노드가 [ScaledownIdletime](#)에 실패한 경우 SuspendProgram 로그에서 slurmctld 프로세스가 특정 노드를 인수로 사용하여 호출했는지 확인하세요. SuspendProgram는 실제로 특정 작업을 수행하지 않습니다. 그보다는 호출될 때만 로그를 기록합니다. 모든 인스턴스 종료 및

NodeAddr 재설정은 clustermgtd가 완료합니다. Slurm는 SuspendTimeout 후 노드를 IDLE로 전환합니다.

기타 문제:

- AWS ParallelCluster 작업 할당이나 규모 조정 결정을 내리지 않습니다. Slurm의 지침에 따라 리소스를 시작, 종료 및 유지 관리하려고 시도할 뿐입니다.

작업 할당, 노드 할당 및 규모 조정 결정과 관련된 문제는 slurmctld 로그에서 오류를 확인하세요.

## Slurm 클러스터 보호 모드

보호 모드가 활성화된 상태에서 클러스터를 실행하면 컴퓨팅 노드가 시작될 때 컴퓨팅 노드 부트스트랩 장애를 AWS ParallelCluster 모니터링하고 추적합니다. 이는 이러한 장애가 지속적으로 발생하는지 여부를 감지하기 위한 것입니다.

대기열(파티션)에서 다음이 감지되면 클러스터는 보호 상태로 전환됩니다.

1. 연이은 컴퓨팅 노드 부트스트랩 장애가 계속 발생하며 컴퓨팅 노드가 성공적으로 시작되지 않습니다.
2. 실패 수가 사전 정의된 임계값에 도달했습니다.

클러스터가 보호 상태로 전환되면 사전 정의된 임계값 이상에서 장애가 발생한 대기열을 AWS ParallelCluster 비활성화합니다.

Slurm클러스터 보호 모드는 버전 3.0.0에 추가되었습니다. AWS ParallelCluster

보호 모드를 사용하면 컴퓨팅 노드 부트스트랩 장애 사이클링에 소요되는 시간과 리소스를 줄일 수 있습니다.

보호 모드 파라미터

### **protected\_failure\_count**

protected\_failure\_count는 클러스터 보호 상태를 활성화하는 대기열(파티션)의 연속 실패 수를 지정합니다.

기본값 protected\_failure\_count는 10이며 보호 모드가 활성화되어 있습니다.

protected\_failure\_count가 0보다 크면 보호 모드가 활성화됩니다.



protected\_failure\_count가 0보다 작거나 같으면 보호 모드가 비활성화됩니다.

HeadNode의 /etc/parallelcluster/slurm\_plugin/parallelcluster\_clustermgtd.conf에 있는 clustermgtd 구성 파일에 파라미터를 추가하여 protected\_failure\_count 값을 변경할 수 있습니다.

이 파라미터는 언제든지 업데이트할 수 있으며 이를 위해 컴퓨팅 플릿을 중지할 필요가 없습니다. 실패 횟수가 protected\_failure\_count에 도달하기 전에 대기열에서 시작이 성공하면 실패 횟수가 0으로 재설정됩니다.

보호된 상태에서 클러스터 상태를 확인합니다.

클러스터가 보호 상태인 경우 컴퓨팅 플릿 상태와 노드 상태를 확인할 수 있습니다.

### 컴퓨팅 플릿 상태

컴퓨팅 플릿의 상태는 보호 상태로 실행 중인 클러스터에 있는 PROTECTED입니다.

```
$ pcluster describe-compute-fleet --cluster-name <cluster-name> --region <region-id>
{
  "status": "PROTECTED",
  "lastStatusUpdatedTime": "2022-04-22T00:31:24.000Z"
}
```

### 노드 상태

보호 상태를 활성화한 부트스트랩 장애가 발생한 대기열(파티션)을 알아보려면 클러스터에 로그인하고 sinfo 명령을 실행하세요. 부트스트랩 실패 횟수가 protected\_failure\_count 이상인 파티션은 현재 INACTIVE 상태입니다. 부트스트랩 실패 횟수가 protected\_failure\_count 이상이 아닌 파티션은 UP 상태이며 예상대로 작동합니다.

PROTECTED 상태는 실행 중인 작업에 영향을 주지 않습니다. 부트스트랩 실패 횟수가 protected\_failure\_count 이상인 파티션에서 작업이 실행되는 경우 해당 파티션은 실행 중인 작업이 완료된 후 INACTIVE로 설정됩니다.

다음 예제에 표시된 노드 상태를 고려해 보세요.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*  inact infinite 10  down% queue1-dy-c5xlarge-[1-10]
```

```
queue1* inact infinite 3490 idle~ queue1-dy-c5xlarge-[11-3500]
queue2 up infinite 10 idle~ queue2-dy-c5xlarge-[1-10]
```

파티션 queue1이 INACTIVE인 이유는 10개의 연속적인 컴퓨팅 노드 부트스트랩 장애가 감지되었기 때문입니다.

노드 뒤에 있는 인스턴스가 queue1-dy-c5xlarge-[1-10]를 시작했지만 비정상 상태로 인해 클러스터에 조인하지 못했습니다.

클러스터가 보호 상태에 있습니다.

queue1의 부트스트랩 실패는 파티션 queue2에 영향을 주지 않습니다. UP 상태이며 여전히 작업을 실행할 수 있습니다.

보호 상태를 비활성화하는 방법

부트스트랩 오류가 해결된 후 다음 명령을 실행하여 클러스터를 보호 상태에서 해제할 수 있습니다.

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

보호 상태를 활성화하는 부트스트랩 장애

보호 상태를 활성화하는 부트스트랩 오류는 다음 세 가지 유형으로 세분됩니다. 유형 및 문제를 식별하기 위해 로그가 AWS ParallelCluster 생성되었는지 확인할 수 있습니다. 로그가 생성된 경우 해당 로그에서 오류 세부 정보를 확인할 수 있습니다. 자세한 정보는 [로그 검색 및 보존](#)을 참조하세요.

1. 부트스트랩 오류로 인해 인스턴스가 자체 종료됩니다.

부트스트랩 프로세스 초기에 인스턴스가 실패합니다. 예를 들어

[SlurmQueues/CustomActions/OnNodeStart/OnNodeConfigured](#) 스크립트의 오류로 인해 자체 종료되는 인스턴스가 있습니다.

동적 노드의 경우 다음과 비슷한 오류가 있는지 확인하세요.

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

정적 노드의 경우 clustermgtd 로그(/var/log/parallelcluster/clustermgtd)에서 다음과 비슷한 오류가 있는지 확인하세요.

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

## 2. 노드 resume\_timeout 또는 node\_replacement\_timeout 만료.

인스턴스는 resume\_timeout(동적 노드의 경우) 또는 node\_replacement\_timeout(정적 노드의 경우) 내에서 클러스터에 조인할 수 없습니다. 타임아웃 전에는 자체 종료되지 않습니다. 예를 들어 클러스터에 네트워킹이 제대로 설정되지 않고 노드가 제한 시간 만료 후 Slurm에 의해 DOWN 상태로 설정됩니다.

동적 노드의 경우 다음과 비슷한 오류가 있는지 확인하세요.

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

정적 노드의 경우 clustermgtd 로그(/var/log/parallelcluster/clustermgtd)에서 다음과 비슷한 오류가 있는지 확인하세요.

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

## 3. 노드 상태 확인에 실패.

노드 뒤에 있는 인스턴스가 Amazon EC2 상태 점검 또는 예정된 이벤트 상태 점검에 실패하고 해당 노드는 부트스트랩 장애 노드로 간주됩니다. 이 경우 제어할 수 없는 이유로 인스턴스가 종료됩니다. AWS ParallelCluster

clustermgtd 로그(/var/log/parallelcluster/clustermgtd)에서 다음과 비슷한 오류가 있는지 확인하세요.

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

## 4. 컴퓨팅 노드가 Slurm 등록에 실패.

Slurm 제어 대몬(daemon)(slurmctld)을 통한 slurmd 대몬(daemon) 등록이 실패하고 컴퓨팅 노드 상태가 INVALID\_REG 상태로 변경됩니다. 잘못 구성된 Slurm 컴퓨팅 노드로 인해 이 오류가 발생할 수 있습니다(예: [CustomSlurmSettings](#) 컴퓨팅 노드 사양 오류로 구성된 컴퓨팅 노드).

헤드 노드의 slurmctld 로그 파일(/var/log/slurmctld.log) 또는 장애가 발생한 컴퓨팅 노드의 slurmd 로그 파일(/var/log/slurmd.log)에서 다음과 유사한 오류가 있는지 확인하세요.

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

## 보호 모드를 디버깅하는 방법

클러스터가 보호 상태이고 문제가 있는 컴퓨팅 노드에서 HeadNode 및 cloud-init-output 로그가 AWS ParallelCluster 생성된 clustermgtd 경우 로그에서 오류 세부 정보를 확인할 수 있습니다. 로그 검색에 대한 자세한 내용은 [로그 검색 및 보존](#) 섹션을 참조하세요.

헤드 노드의 clustermgtd 로그(/var/log/parallelcluster/clustermgtd)

로그 메시지에선 부트스트랩 오류가 발생한 파티션과 해당 부트스트랩 실패 수가 표시됩니다.

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

clustermgtd 로그에서 Found the following bootstrap failure nodes를 검색하여 부트스트랩에 실패한 노드를 찾으세요.

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

clustermgtd 로그에서 Node bootstrap error를 검색하여 실패 원인을 찾으세요.

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

컴퓨팅 노드의 cloud-init-output로그(/var/log/cloud-init-output.log)

clustermgtd 로그에서 부트스트랩 장애 노드 프라이빗 IP 주소를 획득한 후 컴퓨팅 노드에 로그인하거나 지침에 따라 [로그 검색 및 보존](#)에서 로그를 검색하여 해당 컴퓨팅 노드 로그를 찾을 수 있습니다. 대부분의 경우 문제가 있는 노드의 /var/log/cloud-init-output 로그에는 컴퓨팅 노드 부트스트랩 장애를 일으킨 단계가 나와 있습니다.

## Slurm 클러스터 빠른 용량 부족 장애 조치

AWS ParallelCluster 버전 3.2.0부터 클러스터는 빠른 용량 부족 장애 복구 모드가 기본적으로 활성화된 상태로 실행됩니다. 이렇게 하면 Amazon EC2의 용량 부족 오류가 감지될 때 작업을 대기열에 재시도하는 데 소요되는 시간을 최소화할 수 있습니다. 이는 여러 종류의 인스턴스 유형으로 클러스터를 구성할 때 특히 효과적입니다.

Amazon EC2에서 용량 부족 장애를 감지했습니다.

- `InsufficientInstanceCapacity`
- `InsufficientHostCapacity`
- `InsufficientReservedInstanceCapacity`
- `MaxSpotInstanceCountExceeded`
- `SpotMaxPriceTooLow`: 스팟 요청 가격이 필요한 최소 스팟 요청 이행 가격보다 낮습니다.
- `Unsupported`: 특정 AWS 리전인스턴스에서 지원되지 않는 인스턴스 유형을 사용하여 활성화되었습니다.

빠른 용량 부족 장애 복구 모드에서 작업이 [SlurmQueuescompute\\_resource/](#)에 할당될 때 용량 부족 오류가 감지되면 다음을 AWS ParallelCluster 수행합니다.

1. 미리 정의된 기간 동안 컴퓨팅 리소스를 비활성화(DOWN) 상태로 설정합니다.
2. 컴퓨팅 리소스 장애가 발생한 노드 작업을 취소하고 장애가 발생한 노드를 일시 중단하기 위해 `POWER_DOWN_FORCE`를 사용합니다. 장애가 발생한 노드를 IDLE 및 `POWER_DOWN (!)` 상태로 설정한 다음, `POWERING_DOWN (%)`로 설정합니다.
3. 작업을 다른 컴퓨팅 리소스에 대기열에 추가합니다.

비활성화된 컴퓨팅 리소스의 정적 노드와 전원이 켜진 노드는 영향을 받지 않습니다. 이러한 노드에서 작업을 완료할 수 있습니다.

이 주기는 작업이 컴퓨팅 리소스 노드 또는 여러 노드에 성공적으로 할당될 때까지 반복됩니다. 노드 상태에 대한 내용은 [다중 대기열 모드를 위한 Slurm 가이드](#) 섹션을 참조하세요.

작업을 실행할 컴퓨팅 리소스가 없는 경우 작업은 미리 정의된 기간이 경과할 때까지 `PENDING` 상태로 설정됩니다. 이 경우 다음 섹션에 설명된 대로 사전 정의된 기간을 수정할 수 있습니다.

용량 제한 시간 초과 파라미터 부족

### **`insufficient_capacity_timeout`**

`insufficient_capacity_timeout` 용량 부족 오류가 감지된 경우 컴퓨팅 리소스가 비활성화(down) 상태로 유지되는 기간(초)을 지정합니다.

기본값은 `insufficient_capacity_timeout`가 활성화됩니다.

`insufficient_capacity_timeout`의 기본값은 600초(10분)입니다.

`insufficient_capacity_timeout` 값이 0보다 작거나 같으면 빠른 용량 부족 장애 조치 모드가 비활성화됩니다.

HeadNode의 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf`에 있는 `clustermgtd` 구성 파일에 파라미터를 추가하여 `insufficient_capacity_timeout` 값을 변경할 수 있습니다.

컴퓨팅 플릿을 중지하지 않고도 언제든지 파라미터를 업데이트할 수 있습니다.

예:

- `insufficient_capacity_timeout=600`:

용량 부족 오류가 감지되면 컴퓨팅 리소스가 비활성화(DOWN)로 설정됩니다. 10분 후 장애가 발생한 노드는 `idle~(POWER_SAVING)` 상태로 설정됩니다.

- `insufficient_capacity_timeout=60`:

용량 부족 오류가 감지되면 컴퓨팅 리소스는 비활성화(DOWN) 상태가 됩니다. 1분 후 장애가 발생한 노드는 `idle~` 상태로 설정됩니다.

- `insufficient_capacity_timeout=0`:

빠른 용량 부족 장애 조치 모드가 비활성화되었습니다. 컴퓨팅 리소스는 비활성화되지 않았습니다.

#### Note

용량 부족 오류로 인해 노드에 장애가 발생하는 시간과 클러스터 관리 대몬(daemon)이 노드 장애를 감지하는 시간 사이에는 최대 1분의 지연이 있을 수 있습니다. 이는 클러스터 관리 대몬(daemon)이 노드 용량 부족 장애를 확인하고 1분 간격으로 컴퓨팅 리소스를 down 상태로 설정하기 때문입니다.

### 빠른 용량 부족 장애 조치 모드 상태

클러스터가 빠른 용량 부족 장애 조치 모드에 있는 경우 클러스터의 상태와 노드 상태를 확인할 수 있습니다.

#### 노드 상태

컴퓨팅 리소스 동적 노드에 작업이 제출되고 용량 부족 오류가 감지되면 해당 노드는 이유가 있는 `down#` 상태로 전환됩니다.

```
(Code:InsufficientInstanceCapacity)Failure when resuming nodes.
```

그러면 전원이 꺼진 노드(idle~ 상태의 노드)가 이유가 있는 down~로 설정됩니다.

```
(Code:InsufficientInstanceCapacity)Temporarily disabling node due to insufficient capacity.
```

작업은 대기열에 있는 다른 컴퓨팅 리소스로 대기열에 추가됩니다.

컴퓨팅 리소스 정적 노드 및 UPI 빠른 용량 부족 장애 조치 모드의 영향을 받지 않는 노드입니다.

다음 예제에 표시된 노드 상태를 고려해 보세요.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

노드 하나가 필요한 작업을 queue1에 제출합니다.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    1   down# queue1-dy-c-1-1
queue1*   up    infinite    15  idle~ queue1-dy-c-2-[1-15]
queue1*   up    infinite    14  down~ queue1-dy-c-1-[2-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

작업을 실행하기 위해 노드 queue1-dy-c-1-1이 시작됩니다. 그러나 용량 부족 오류로 인해 인스턴스가 시작되지 않았습니다. 노드 queue1-dy-c-1-1이 down으로 설정되었습니다. 컴퓨팅 리소스(queue2-dy-c-1) 내의 전원이 꺼진 동적 노드는 down으로 설정되어 있습니다.

scontrol show nodes로 노드 이유를 확인할 수 있습니다.

```
$ scontrol show nodes queue1-dy-c-1-1
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Failure when resuming nodes
[root@2022-03-10T22:17:50]
```

```
$ scontrol show nodes queue1-dy-c-1-2
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Temporarily disabling node due to
insufficient capacity [root@2022-03-10T22:17:50]
```

작업이 대기열 컴퓨팅 리소스 내의 다른 인스턴스 유형에 대기됩니다.

`insufficient_capacity_timeout`이 경과하면 컴퓨팅 리소스의 노드가 `idle~` 상태로 재설정됩니다.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

`insufficient_capacity_timeout`이 경과하고 컴퓨팅 리소스의 노드가 해당 `idle~` 상태로 재설정되면, Slurm 스케줄러는 노드에 더 낮은 우선 순위를 부여합니다. 스케줄러는 다음 중 하나가 발생하지 않는 한 가중치가 더 높은 다른 대기열 컴퓨팅 리소스에서 노드를 계속 선택합니다.

- 작업 제출 요구 사항은 복구된 컴퓨팅 리소스와 일치합니다.
- 용량이 충분하기 때문에 다른 컴퓨팅 리소스를 사용할 수 없습니다.
- `slurmctld`가 다시 시작됩니다.
- AWS ParallelCluster 컴퓨팅 플릿을 중지한 다음 모든 노드의 전원을 끄고 다시 켜기 시작합니다.

## 관련 로그

용량 부족 오류 및 빠른 용량 부족 장애 조치 모드와 관련된 로그는 Slurm의 `resume` 로그 및 헤드 노드의 `clustermgtd` 로그에서 확인할 수 있습니다.

## Slurm `resume (/var/log/parallelcluster/slurm_resume.log)`

용량이 부족하여 노드 시작에 실패할 때 나타나는 오류 메시지.

```
[slurm_plugin.instance_manager:_launch_ec2_instances] - ERROR - Failed RunInstances
request: dcd0c252-90d4-44a7-9c79-ef740f7ecd87
```



```
[slurm_plugin.instance_manager:add_instances_for_nodes] - ERROR - Encountered
exception when launching instances for nodes (x1) ['queue1-dy-c-1-1']: An error
occurred
(InsufficientInstanceCapacity) when calling the RunInstances operation (reached max
retries: 1): We currently do not have sufficient p4d.24xlarge capacity in the
Availability Zone you requested (us-west-2b). Our system will be working on
provisioning additional capacity. You can currently get p4d.24xlarge capacity by
not
specifying an Availability Zone in your request or choosing us-west-2a, us-west-2c.
```

## Slurm `clustermgtd (/var/log/parallelcluster/clustermgtd)`

용량이 부족하여 queue1의 컴퓨팅 리소스 c-1이 비활성화되었습니다.

```
[slurm_plugin.clustermgtd:_reset_timeout_expired_compute_resources] - INFO - The
following compute resources are in down state
due to insufficient capacity: {'queue1': {'c-1':
ComputeResourceFailureEvent(timestamp=datetime.datetime(2022, 4, 14, 23, 0, 4,
769380, tzinfo=datetime.timezone.utc),
error_code='InsufficientInstanceCapacity')}}}, compute resources are reset after
insufficient capacity timeout (600 seconds) expired
```

용량 부족 제한 시간이 만료되면 컴퓨팅 리소스가 재설정되고 컴퓨팅 리소스 내의 노드는 `idle`로 설정됩니다.

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the
following compute resources because insufficient capacity
timeout expired: {'queue1': ['c-1']}
```

## Slurm 메모리 기반 스케줄링

버전 3.2.0부터 [SlurmSettings/EnableMemoryBasedScheduling](#) cluster 구성 매개 변수를 사용하여 Slurm 메모리 기반 스케줄링을 AWS ParallelCluster 지원합니다.

### Note

AWS ParallelCluster [버전 3.7.0부터 인스턴스에서 여러 인스턴스 유형을 구성하면 EnableMemoryBasedScheduling 활성화할 수 있습니다.](#)

AWS ParallelCluster 버전 3.2.0~3.6의 경우 **x**, [인스턴스에서 여러 인스턴스 유형을 구성한 경우 활성화할 EnableMemoryBasedScheduling 수 없습니다.](#)

**⚠ Warning**

EnableMemoryBasedScheduling이 활성화된 상태에서 Slurm 대기열 컴퓨팅 리소스에 여러 인스턴스 유형을 지정하는 경우 RealMemory 값은 모든 인스턴스 유형에 사용할 수 있는 최소 메모리 양입니다. 메모리 용량이 매우 다른 인스턴스 유형을 지정하는 경우 이로 인해 사용되지 않은 메모리가 상당량 발생할 수 있습니다.

EnableMemoryBasedScheduling: true를 사용하면, Slurm 스케줄러가 각 노드에서 각 작업에 필요한 메모리 양을 추적합니다. 그런 다음 Slurm 스케줄러는 이 정보를 사용하여 동일한 컴퓨팅 노드에서 여러 작업을 스케줄링합니다. 노드에서 작업에 필요한 총 메모리 양은 사용 가능한 노드 메모리보다 클 수 없습니다. 스케줄러는 작업이 제출될 때 요청된 것보다 더 많은 메모리를 작업이 사용하지 않도록 합니다.

EnableMemoryBasedScheduling: false를 사용하면 공유 노드의 메모리를 놓고 작업이 경쟁하여 작업 실패 및 out-of-memory 이벤트가 발생할 수 있습니다.

**⚠ Warning**

Slurm은 레이블에 2의 거듭제곱 표기법(예: MB 또는 GB)을 사용합니다. 이 레이블을 각각 MiB 및 GiB로 읽습니다.

**Slurm 구성 및 메모리 기반 스케줄링**

EnableMemoryBasedScheduling: true를 사용하면 Slurm는 다음과 같은 Slurm 구성 파라미터를 설정합니다.

- slurm.conf의 [SelectTypeParameters=CR\\_CPU\\_Memory](#). 이 옵션은 노드 메모리를 Slurm에서 사용 가능한 리소스로 구성합니다.
- Slurm cgroup.conf의 [ConstrainRAMSpace=yes](#). 이 옵션을 사용하면 작업이 제출될 때 요청한 메모리 양만큼 작업의 메모리 액세스가 제한됩니다.

**i Note**

이 두 옵션이 설정된 경우 다른 여러 Slurm 구성 파라미터가 Slurm 스케줄러 및 리소스 관리자의 동작에 영향을 줄 수 있습니다. 자세한 내용은 [Slurm 설명서](#)를 참조하세요.

## Slurm 스케줄러 및 메모리 기반 스케줄링

### **EnableMemoryBasedScheduling: false**(기본값)

기본적으로 `EnableMemoryBasedScheduling`는 `false`로 설정됩니다. `false`인 경우, Slurm은 스케줄링 알고리즘에 메모리를 리소스로 포함하지 않으며 작업에서 사용하는 메모리를 추적하지 않습니다. 사용자는 작업에 필요한 노드당 최소 메모리 양을 설정하는 `--mem MEM_PER_NODE` 옵션을 지정할 수 있습니다. 이렇게 하면 스케줄러는 작업을 예약할 때 `RealMemory` 값이 최소한 `MEM_PER_NODE`인 노드를 선택해야 합니다.

예를 들어, 한 사용자가 `--mem=5GB`를 사용하여 두 개의 작업을 제출한다고 가정해 보겠습니다. CPU 또는 GPU와 같은 요청된 리소스를 사용할 수 있는 경우 8GiB 메모리가 있는 노드에서 작업을 동시에 실행할 수 있습니다. 이 두 작업은 `RealMemory` 5GiB 미만의 컴퓨팅 노드에서는 스케줄되지 않습니다.

#### Warning

메모리 기반 스케줄링이 비활성화되면 Slurm은 작업에서 사용하는 메모리 양을 추적하지 않습니다. 동일한 노드에서 실행되는 작업은 메모리 리소스를 놓고 경쟁하여 다른 작업이 실패할 수 있습니다.

메모리 기반 스케줄링을 사용하지 않도록 설정한 경우에는 사용자가 [--mem-per-cpu](#) 또는 [--mem-per-gpu](#) 옵션을 지정하지 않는 것이 좋습니다. 이러한 옵션으로 인해 [Slurm 설명서](#)에 설명된 것과 다른 동작이 발생할 수 있습니다.

### **EnableMemoryBasedScheduling: true**

`EnableMemoryBasedScheduling`를 `true`로 설정하면 Slurm은 각 작업의 메모리 사용량을 추적하여 `--mem` 제출 옵션에서 요청한 것보다 많은 메모리를 작업에 사용하지 않도록 합니다.

이전 예제를 사용하면 한 사용자가 `--mem=5GB`를 사용하여 두 개의 작업을 제출합니다. 메모리가 8GiB인 노드에서는 작업을 동시에 실행할 수 없습니다. 필요한 총 메모리 양이 노드에서 사용 가능한 메모리보다 많기 때문입니다.

메모리 기반 스케줄링이 활성화된 상태에서 `--mem-per-cpu` 및 `--mem-per-gpu`는 Slurm 설명서에 설명된 것과 일관되게 작동합니다. 예를 들어, 작업이 `--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB`로 제출됩니다. 이 경우 Slurm은 각 노드에 총 4GiB를 작업에 할당합니다.

**⚠ Warning**

메모리 기반 스케줄링이 활성화된 경우 사용자가 작업을 제출할 때 `--mem` 사양을 포함하는 것이 좋습니다. 에 포함된 기본 Slurm AWS ParallelCluster 구성에서는 메모리 옵션이 포함되지 않은 경우 (`--mem`, `--mem-per-cpu`, 또는 `--mem-per-gpu`) 할당된 노드의 전체 메모리를 작업에 Slurm 할당합니다. 이는 CPU 또는 GPU와 같은 다른 리소스의 일부만 요청하는 경우에도 마찬가지입니다. 이렇게 하면 다른 작업에 사용할 수 있는 메모리가 없으므로 작업이 완료될 때까지 노드 공유를 효과적으로 방지할 수 있습니다. 이는 작업 제출 시 메모리 사양이 제공되지 않은 경우 Slurm이 작업의 노드당 메모리를 `DefMemPerNode`로 설정하기 때문에 발생합니다. 이 파라미터의 기본값은 0이며 노드 메모리에 대한 무제한 액세스를 지정합니다. 동일한 대기열에 메모리 양이 다른 여러 유형의 컴퓨팅 리소스가 있는 경우 메모리 옵션 없이 제출된 작업에는 노드마다 다른 양의 메모리가 할당될 수 있습니다. 이는 스케줄러가 작업에 사용할 수 있는 노드에 따라 달라집니다. 사용자는 Slurm 구성 파일의 클러스터 또는 파티션 수준에서 `DefMemPerNode` 또는 `DefMemPerCPU` 같은 옵션에 대한 사용자 지정 값을 정의하여 이러한 동작을 방지할 수 있습니다.

**Slurm RealMemory 그리고 AWS ParallelCluster SchedulableMemory**

함께 AWS ParallelCluster 제공되는 Slurm 구성을 사용하면 작업에 사용할 수 `RealMemory` 있는 노드당 메모리 양으로 Slurm 해석됩니다. 버전 3.2.0부터는 기본적으로 Amazon [EC2 인스턴스 유형에 나열된 메모리의 95% RealMemory](#) 로 AWS ParallelCluster 설정되며 Amazon EC2 API에서 반환됩니다. [DescribeInstanceTypes](#)

메모리 기반 스케줄링이 비활성화되면 Slurm 스케줄러는 사용자가 `RealMemory`를 사용해서 `--mem`이 지정된 작업을 제출할 때 노드를 필터링합니다.

메모리 기반 스케줄링이 활성화된 경우 Slurm 스케줄러는 `RealMemory`를 컴퓨팅 노드에서 실행 중인 작업에 사용할 수 있는 최대 메모리 양으로 해석합니다.

기본 설정은 모든 인스턴스 유형에 적합하지 않을 수 있습니다.

- 이 설정은 노드가 실제로 액세스할 수 있는 메모리 양보다 많을 수 있습니다. 이는 컴퓨팅 노드가 작은 인스턴스 유형일 때 발생할 수 있습니다.
- 이 설정은 노드가 실제로 액세스할 수 있는 메모리 양보다 적을 수 있습니다. 이는 컴퓨팅 노드가 인스턴스 유형이 크고 사용되지 않은 메모리가 상당히 많을 때 발생할 수 있습니다.

[SlurmQueuesComputeResourcesSchedulableMemory](#)를 사용하여 컴퓨팅 노드의 RealMemory 구성 기준 값을 미세 조정할 수 있습니다. AWS ParallelCluster 기본값을 재정의하려면 클러스터 구성에 맞게 SchedulableMemory의 사용자 지정 값을 정의하세요.

컴퓨팅 노드의 실제 사용 가능한 메모리를 확인하려면 노드에서 `/opt/slurm/sbin/slurmd -C` 명령을 실행합니다. 이 명령은 [RealMemory](#) 값을 포함하여 노드의 하드웨어 구성을 반환합니다. 자세한 정보는 [slurmd -C](#)을 참조하세요.

컴퓨팅 노드의 운영 체제 프로세스에 충분한 메모리가 있는지 확인하세요. 이렇게 하려면 SchedulableMemory 값을 slurmd -C 명령이 반환한 RealMemory 값보다 낮게 설정하여 작업에 사용할 수 있는 메모리를 제한하세요.

## Slurm을 사용하여 여러 인스턴스 유형 할당

AWS ParallelCluster 버전 3.3.0부터 컴퓨팅 리소스의 정의된 인스턴스 유형 집합에서 할당하도록 클러스터를 구성할 수 있습니다. 할당은 Amazon EC2 플릿 저가 또는 최적의 용량 전략을 기반으로 할 수 있습니다.

정의된 인스턴스 유형 집합은 모두 동일한 수의 vCPU를 가져야 하며, 멀티스레딩이 비활성화된 경우 코어 수가 같아야 합니다. 또한 인스턴스 유형 집합은 동일한 제조업체의 액셀러레이터 수가 같아야 합니다. [Efa/Enabled](#)가 true로 설정된 경우 인스턴스는 EFA를 지원해야 합니다. 요구 사항에 대한 자세한 내용은 [Scheduling/SlurmQueues/AllocationStrategy](#) 및 [ComputeResources/Instances](#) 섹션을 참조하세요.

구성에 capacity-optimized 따라 lowest-price 또는 [CapacityType](#) 구성에 [AllocationStrategy](#) 따라 설정할 수 있습니다.

[Instances](#)에서 인스턴스 유형 집합을 구성할 수 있습니다.

### Note

AWS ParallelCluster [버전 3.7.0부터 인스턴스에서 여러 인스턴스 유형을 구성하면 EnableMemoryBasedScheduling 활성화할 수 있습니다.](#)

AWS ParallelCluster 버전 3.2.0~3.6의 경우, [x, 인스턴스에서 여러 인스턴스 유형을 구성한 경우 활성화할 EnableMemoryBasedScheduling 수 없습니다.](#)

다음 예에서는 vCPU, EFA 지원 및 아키텍처의 인스턴스 유형을 쿼리하는 방법을 보여줍니다.

96개의 vCPU 및 x86\_64 아키텍처를 사용하여 InstanceTypes을 쿼리합니다.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-vcpus,Values=96" "Name=processor-info.supported-
architecture,Values=x86_64" \
  --query "sort_by(InstanceTypes[*].
{InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
&InstanceType})" \
  --output table
```

64코어, EFA 지원 및 arm64 아키텍처를 사용하여 InstanceTypes을 쿼리합니다.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-cores,Values=64" "Name=processor-
info.supported-architecture,Values=arm64" "Name=network-info.efa-
supported,Values=true" --query "sort_by(InstanceTypes[*].
{InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
&InstanceType})" \
  --output table
```

다음 예제 클러스터 구성 조각은 이러한 InstanceType과 AllocationStrategy 속성을 사용하는 방법을 보여줍니다.

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue-1
      CapacityType: ONDEMAND
      AllocationStrategy: lowest-price
      ...
    ComputeResources:
      - Name: computeresource1
        Instances:
          - InstanceType: r6g.2xlarge
          - InstanceType: m6g.2xlarge
          - InstanceType: c6g.2xlarge
        MinCount: 0
        MaxCount: 500
      - Name: computeresource2
        Instances:
          - InstanceType: m6g.12xlarge
          - InstanceType: x2gd.12xlarge
        MinCount: 0
```

```
MaxCount: 500
```

```
...
```

## 동적 노드의 클러스터 스케일링

ParallelCluster Slurm의 절전 플러그인을 사용하여 클러스터를 동적으로 확장하는 Slurm의 메서드를 지원합니다. 자세한 내용은 Slurm 설명서의 [클라우드 일정 가이드](#) 및 [Slurm 절전 가이드](#)를 참조하세요.

ParallelCluster 버전 3.8.0부터 작업 수준 재개 또는 작업 수준 조정을 기본 동적 노드 할당 전략으로 ParallelCluster 사용하여 클러스터를 확장합니다. 즉, 각 작업의 요구 사항, 작업에 할당된 노드 수, 재개해야 하는 노드를 기반으로 클러스터를 확장합니다. ParallelCluster ParallelCluster SLURM\_RESUME\_FILE 환경 변수에서 이 정보를 가져옵니다.

동적 노드의 확장은 EC2 인스턴스를 시작하고 시작된 Amazon EC2 인스턴스를 노드에 할당하는 Slurm 과정을 포함하는 2단계 프로세스입니다. 이 두 단계는 각각 all-or-nothing 또는 최선형 로직을 사용하여 수행할 수 있습니다.

Amazon EC2 인스턴스를 시작하는 경우:

- all-or-nothing 총 대상 용량과 동일한 최소 대상을 지정하여 Amazon EC2 시작 API를 호출합니다.
- best-effort: 최소 목표 용량이 1이고 총 목표 용량이 요청된 용량과 동일한 상태에서 Amazon EC2 API 시작을 호출합니다.

Amazon EC2 인스턴스를 노드에 할당하는 Slurm 경우:

- all-or-nothing 요청된 모든 노드에 Amazon EC2 인스턴스를 Slurm 할당할 수 있는 경우에만 Amazon EC2 인스턴스를 노드에 할당합니다.
- best-effort: 요청된 모든 노드가 Amazon EC2 인스턴스 Slurm 용량을 지원하지 않는 경우에도 Amazon EC2 인스턴스를 노드에 할당합니다.

위 전략의 가능한 조합은 시작 전략으로 해석됩니다. ParallelCluster

### Example

<caption>The available ParallelCluster 출시 전략 that can be set into the [ScalingStrategy](#) cluster configuration to be used with 직무 수준 규모 조정 are:</caption>

all-or-nothing 스케일링:

이 전략에는 요청된 컴퓨팅 노드를 성공적으로 AWS ParallelCluster 시작하는 데 필요한 모든 인스턴스가 필요한 각 작업에 대해 Amazon EC2 시작 인스턴스 API 호출을 시작하는 것이 포함됩니다. 이렇게 하면 작업당 필요한 용량을 사용할 수 있을 때만 클러스터를 확장하여 조정 프로세스 종료 시 유휴 인스턴스가 남아 있지 않도록 할 수 있습니다.

이 전략은 각 작업 플러스의 Amazon EC2 인스턴스 시작 로직과 Amazon EC2 인스턴스를 노드에 할당하는 all-or-nothing로직을 사용합니다. all-or-nothingSlurm

이 전략은 시작 요청을 요청된 컴퓨팅 리소스당 하나씩, 각 노드를 최대 500개까지 일괄 처리하여 그룹화합니다. 여러 컴퓨팅 리소스에 걸쳐 있거나 500개 이상의 노드를 초과하는 요청의 경우 여러 배치를 ParallelCluster 순차적으로 처리합니다.

단일 리소스의 일괄 처리에 장애가 발생하면 관련 미사용 용량이 모두 종료되므로 조정 프로세스가 끝날 때 유휴 인스턴스가 남아 있지 않습니다.

### 제한 사항

- 확장에 걸리는 시간은 Slurm 재개 프로그램 실행당 제출된 작업 수에 정비례합니다.
- 조정 작업은 기본적으로 1000개 인스턴스로 설정된 RunInstances 자원 계정 한도에 따라 제한됩니다. 이 제한은 AWS의 EC2 API 제한 정책을 따릅니다. 자세한 내용은 [Amazon EC2 API 제한 설명서](#)를 참조하십시오.
- 여러 가용 영역에 걸친 대기열에 있는 단일 인스턴스 유형의 컴퓨팅 리소스에서 작업을 제출하면 단일 가용 영역에서 모든 용량을 제공할 수 있는 경우에만 all-or-nothingEC2 시작 API 호출이 성공합니다.
- 단일 가용 영역이 있는 대기열에 있는 여러 인스턴스 유형이 있는 컴퓨팅 리소스에서 작업을 제출하면 단일 인스턴스 유형으로 모든 용량을 제공할 수 있는 경우에만 all-or-nothingAmazon EC2 시작 API 호출이 성공합니다.
- 여러 가용 영역에 걸친 대기열에 있는 여러 인스턴스 유형이 있는 컴퓨팅 리소스에서 작업을 제출하는 경우 all-or-nothingAmazon EC2 시작 API 호출은 지원되지 않으며 대신 최선의 조정을 수행합니다.

### greedy-all-or-nothing규모 조정:

이 all-or-nothing 전략을 변형하면 여전히 작업당 필요한 용량을 사용할 수 있을 때만 클러스터를 확장하여 확장 프로세스 종료 시 유휴 인스턴스를 피할 수 있지만, 최소 목표 용량 1을 목표로 ParallelCluster 하는 Amazon EC2 시작 인스턴스 API 호출을 시작하여 요청된 용량까지 시작된 노드 수를 최대화하는 것이 포함됩니다. 이 전략에서는 모든 작업에 대해 EC2 인스턴스를 시작하기 위한 최



선의 로직과 각 작업의 노드에 Amazon EC2 인스턴스를 Slurm 할당하는 all-or-nothing 로직을 사용합니다.

전략 그룹은 요청 시작 요청을 요청된 컴퓨팅 리소스당 하나씩, 각 노드를 최대 500개까지 일괄 처리합니다. 여러 컴퓨팅 리소스에 걸쳐 있거나 500개 이상의 노드를 초과하는 요청의 경우 Parallelcluster는 여러 배치를 순차적으로 처리합니다.

확장 프로세스 중에 일시적인 오버스케일링을 감수하고 처리량을 극대화함으로써 확장 프로세스가 끝날 때 유휴 인스턴스가 남지 않도록 합니다.

#### 제한 사항

- 일시적인 오버스케일링이 가능하므로 규모 조정 완료 전에 실행 상태로 전환되는 인스턴스에는 추가 비용이 발생합니다.
- AWS의 RunInstances 자원 계정 한도에 따라 all-or-nothing 전략과 동일한 인스턴스 한도가 적용됩니다.

#### 최대한의 규모 조정:

이 전략은 최소 용량 1을 목표로 하고 요청된 용량을 모두 사용할 수 없는 경우 조정 프로세스 실행 후 유휴 인스턴스를 남겨두는 대신 요청된 총 용량을 달성하는 것을 목표로 하여 Amazon EC2 시작 인스턴스 API 호출을 호출합니다. 이 전략에서는 모든 작업에 대해 Amazon EC2 인스턴스를 시작하는 최선의 로직과 각 작업의 Slurm 노드에 Amazon EC2 인스턴스를 할당하는 최선 로직을 사용합니다.

이 전략은 시작 요청을 요청된 컴퓨팅 리소스당 하나씩, 각 노드를 최대 500개까지 일괄 처리하여 그룹화합니다. 여러 컴퓨팅 리소스에 걸쳐 있거나 500개 이상의 노드를 초과하는 요청의 경우 여러 배치를 ParallelCluster 순차적으로 처리합니다.

이 전략을 사용하면 여러 조정 프로세스에서 유휴 인스턴스를 유지하는 대신 여러 조정 프로세스 실행에 대해 기본 1,000개 인스턴스 한도를 훨씬 초과하여 확장할 수 있습니다.

#### 제한 사항

- 작업에서 요청한 모든 노드를 할당할 수 없는 경우, 조정 프로세스 종료 시 유휴 실행 인스턴스가 발생할 수 있습니다.

다음은 다양한 ParallelCluster 시작 전략을 사용하여 동적 노드 확장이 어떻게 작동하는지 보여주는 예제입니다. 동일한 유형의 총 40개 노드에 대해 각각 20개의 노드를 요청하는 두 개의 작업을 제출했지만 EC2에서 요청된 용량을 처리할 수 있는 Amazon EC2 인스턴스가 30개뿐이라고 가정해 보겠습니다.

### all-or-nothing 규모 조정:

- 첫 번째 작업에서는 all-or-nothing Amazon EC2 시작 인스턴스 API가 호출되어 20개의 인스턴스를 요청합니다. 호출이 성공하면 20개의 인스턴스가 시작됩니다.
- all-or-nothing 첫 번째 작업을 위해 시작된 인스턴스 20개를 Slurm 노드에 성공적으로 할당했습니다.
- 또 다른 all-or-nothing Amazon EC2 시작 인스턴스 API가 호출되어 두 번째 작업에 사용할 20개의 인스턴스를 요청합니다. 다른 10개의 인스턴스를 위한 용량만 있기 때문에 호출이 성공하지 못했습니다. 지금은 인스턴스가 시작되지 않았습니다.

### greedy-all-or-nothing 스케일링:

- 최선을 다해 Amazon EC2 시작 인스턴스 API를 호출하여 모든 작업에서 요청한 총 용량인 40개의 인스턴스를 요청합니다. 그 결과 30개의 인스턴스가 시작됩니다.
- 첫 번째 작업을 위해 시작된 인스턴스 20개를 Slurm 노드에 성공적으로 all-or-nothing 할당했습니다.
- 두 번째 작업을 위해 나머지 시작된 인스턴스를 Slurm 노드에 다시 all-or-nothing 할당하려고 시도하지만 작업에서 요청한 총 20개 중 사용 가능한 인스턴스가 10개뿐이므로 할당에 실패했습니다.
- 할당되지 않은 시작 인스턴스 10개가 종료됩니다.

### 최선의 규모 조정:

- 최선을 다해 Amazon EC2 시작 인스턴스 API를 호출하여 모든 작업에서 요청한 총 용량인 40개의 인스턴스를 요청합니다. 그 결과 30개의 인스턴스가 시작됩니다.
- 첫 번째 작업을 위해 시작된 인스턴스 20개를 Slurm 노드에 최선을 다해 할당하면 성공합니다.
- 요청된 총 용량이 20인 경우에도 나머지 10개의 시작된 인스턴스를 두 번째 작업을 위해 Slurm 노드에 최선을 다해 할당하는 데 성공했습니다. 그러나 작업이 20개 노드를 요청했고 그 중 10개 노드에 Amazon EC2 인스턴스를 할당할 수 있었기 때문에 나중에 확장 프로세스를 호출할 때 누락된 10개의 인스턴스를 시작할 수 있을 만큼 충분한 용량을 찾거나 스케줄러가 이미 실행 중인 다른 컴퓨팅 노드에 작업을 스케줄링할 때까지 작업을 시작할 수 없고 인스턴스는 유휴 상태로 남아 있습니다.

### Slurm 버전 3.7.x의 동적 노드 할당 전략

ParallelCluster 두 가지 유형의 동적 노드 할당 전략을 사용하여 클러스터를 확장합니다.

- 사용 가능한 요청 노드 정보를 기반으로 한 할당:
  - 모든 노드 재개 또는 노드 목록 규모 조정:

ParallelCluster 가 ResumeProgram 실행될 때 Slurm 요청된 노드 목록 이름만을 기반으로 Slurm 클러스터를 확장합니다. 노드 이름으로만 노드에 컴퓨팅 리소스를 할당합니다. 노드 이름 목록은 여러 작업에 걸쳐 있을 수 있습니다.

- 직무 수준 재개 또는 직무 수준 규모 조정:

ParallelCluster 각 작업의 요구 사항, 작업에 할당된 현재 노드 수, 재개해야 하는 노드를 기반으로 클러스터를 확장합니다. ParallelCluster SLURM\_RESUME\_FILE 환경 변수에서 이 정보를 가져옵니다.

- Amazon EC2 출시 전략을 사용한 할당:

- 최선의 규모 조정:

ParallelCluster 요청된 노드를 지원하는 데 필요한 일부 인스턴스 (반드시 필요하지는 않음) 를 시작하기 위해 최소 목표 용량이 1인 Amazon EC2 시작 인스턴스 API 호출을 사용하여 클러스터를 확장합니다.

- 규모 All-or-nothing 조정:

ParallelCluster 요청된 노드를 지원하는 데 필요한 모든 인스턴스가 시작된 경우에만 성공하는 Amazon EC2 시작 인스턴스 API 호출을 사용하여 클러스터를 확장합니다. 이 경우 요청된 총 용량과 동일한 최소 목표 용량을 사용하여 Amazon EC2 시작 인스턴스 API를 호출합니다.

기본적으로 는 최선의 Amazon EC2 시작 전략과 함께 노드 목록 확장을 ParallelCluster 사용하여 요청된 노드를 지원하는 데 필요한 일부 인스턴스를 시작하지만 반드시 전부는 아닙니다. 제출된 워크로드를 처리하기 위해 최대한 많은 용량을 프로비저닝하려고 합니다.

ParallelCluster 버전 3.7.0부터 전용 모드로 제출된 작업에 대해 all-or-nothing EC2 시작 전략을 통한 작업 수준 조정을 ParallelCluster 사용합니다. 단독 모드에서 작업을 제출하면 작업은 할당된 노드에 독점적으로 액세스할 수 있습니다. 자세한 내용은 Slurm 설명서의 [단독](#)을 참조하세요.

단독 모드에서 작업을 제출하려면:

- 클러스터에 Slurm 작업을 제출할 때 단독 플래그를 전달하세요. 예를 들어 `sbatch ... --exclusive`입니다.

OR

- [JobExclusiveAllocation](#)이 true로 설정된 상태로 구성된 클러스터 대기열에 작업을 제출합니다.

단독 모드에서 작업을 제출하는 경우:

- ParallelCluster 현재 최대 500개의 노드를 포함하도록 시작 요청을 일괄 처리하고 있습니다. 작업에서 500개 이상의 노드를 요청하는 경우 각 500개 노드 집합에 대해 all-or-nothing 시작 요청을 하고 나머지 노드 집합에 대해 추가 시작 요청을 합니다. ParallelCluster
- 노드 할당이 단일 컴퓨팅 리소스에 있는 경우 500개 노드 집합마다 all-or-nothing 시작 요청을 보내고 나머지 노드 집합에 대해서는 추가 시작 요청을 합니다. ParallelCluster 시작 요청이 실패하면 모든 시작 요청에서 생성된 미사용 용량이 ParallelCluster 종료됩니다.
- 노드 할당이 여러 컴퓨팅 리소스에 걸친 경우 각 컴퓨팅 리소스에 대해 all-or-nothing 시작 요청을 ParallelCluster 해야 합니다. 이러한 요청도 일괄 처리됩니다. 컴퓨팅 리소스 중 하나에 대한 시작 요청이 실패하면 모든 컴퓨팅 리소스 시작 요청에서 생성된 미사용 용량이 ParallelCluster 종료됩니다.

all-or-nothing 시작 전략에 따른 작업 수준 조정 (알려진 제한 사항 포함):

- 여러 가용 영역에 걸친 대기열에 있는 단일 인스턴스 유형의 컴퓨팅 리소스에서 작업을 제출하면 단일 가용 영역에서 모든 용량을 제공할 수 있는 경우에만 all-or-nothing EC2 시작 API 호출이 성공합니다.
- 단일 가용 영역이 있는 대기열에 있는 여러 인스턴스 유형이 있는 컴퓨팅 리소스에서 작업을 제출하면 단일 인스턴스 유형으로 모든 용량을 제공할 수 있는 경우에만 all-or-nothing Amazon EC2 시작 API 호출이 성공합니다.
- 여러 가용 영역에 걸친 대기열에 있는 여러 인스턴스 유형이 있는 컴퓨팅 리소스에서 작업을 제출하는 경우 all-or-nothing Amazon EC2 시작 API 호출은 지원되지 않으며 대신 최선의 조정을 수행합니다.

Slurm 버전 3.6.x 및 이전 버전의 동적 노드 할당 전략

AWS ParallelCluster 다음 한 가지 유형의 동적 노드 할당 전략만 사용하여 클러스터를 확장합니다.

- 사용 가능한 요청 노드 정보를 기반으로 한 할당:
  - 모든 노드 재개 또는 노드 목록 ParallelCluster 확장: 가 실행될 때 Slurm 요청된 노드 목록 이름만을 기반으로 Slurm 클러스터를 확장합니다. ResumeProgram 노드 이름으로만 노드에 컴퓨팅 리소스를 할당합니다. 노드 이름 목록은 여러 작업에 걸쳐 있을 수 있습니다.
- Amazon EC2 출시 전략을 사용한 할당:
  - 최대한의 ParallelCluster 확장: 최소 목표 용량을 1로 설정한 Amazon EC2 시작 인스턴스 API 호출을 사용하여 클러스터를 확장하여 요청된 노드를 지원하는 데 필요한 일부 인스턴스를 시작하지만 반드시 필요한 것은 아닙니다.

ParallelCluster 최선의 Amazon EC2 시작 전략과 함께 노드 목록 확장을 사용하여 요청된 노드를 지원하는 데 필요한 일부 인스턴스를 시작하지만 반드시 필요한 것은 아닙니다. 제출된 워크로드를 처리하기 위해 최대한 많은 용량을 프로비저닝하려고 합니다.

### 제한 사항

- 확장 프로세스 종료 시 (작업에서 요청한 모든 노드를 할당할 수 없는 경우) 유휴 실행 인스턴스가 발생할 수 있습니다.

## Slurm회계 담당자: AWS ParallelCluster

버전 3.3.0부터 클러스터 구성 [SlurmSettings매개변수/데이터베이스](#)를 사용한 Slurm 어카운팅을 AWS ParallelCluster 지원합니다.

버전 3.10.0부터 클러스터 구성 매개 변수 /를 사용하여 외부 Slurmdbd를 사용한 Slurm 어카운팅을 AWS ParallelCluster 지원합니다. [SlurmSettingsExternalSlurmdbd](#) 여러 클러스터가 동일한 데이터베이스를 공유하는 경우 외부 Slurmdbd를 사용하는 것이 좋습니다.

Slurm 회계를 사용하면 외부 계정 데이터베이스를 통합하여 다음 작업을 수행할 수 있습니다.

- 클러스터 사용자 또는 사용자 그룹 및 기타 엔터티를 관리합니다. 이 기능을 사용하면 리소스 제한 적용, 공정 공유 및 QoS와 같은 Slurm의 고급 기능을 사용할 수 있습니다.
- 작업을 실행한 사용자, 작업 기간, 사용한 리소스와 같은 작업 데이터를 수집하고 저장합니다. sacct 유틸리티를 사용하여 저장된 데이터를 볼 수 있습니다.

### Note

AWS ParallelCluster [Slurm지원되는 MySQL 데이터베이스](#) 서버의 Slurm 계정을 지원합니다.

AWS ParallelCluster v3.10.0 Slurmdbd 이상에서 외부를 사용하여 Slurm 계정 작업을 수행합니다.

Slurm계정을 구성하려면 먼저 기존 외부 데이터베이스 서버에 연결되는 기존 외부 Slurmdbd 데이터베이스 서버가 있어야 합니다.

이를 구성하려면 다음을 정의하십시오.

- [ExternalSlurmdbd/Host의](#) 외부 Slurmdbd 서버 주소. 서버가 존재하고 헤드 노드에서 연결할 수 있어야 합니다.

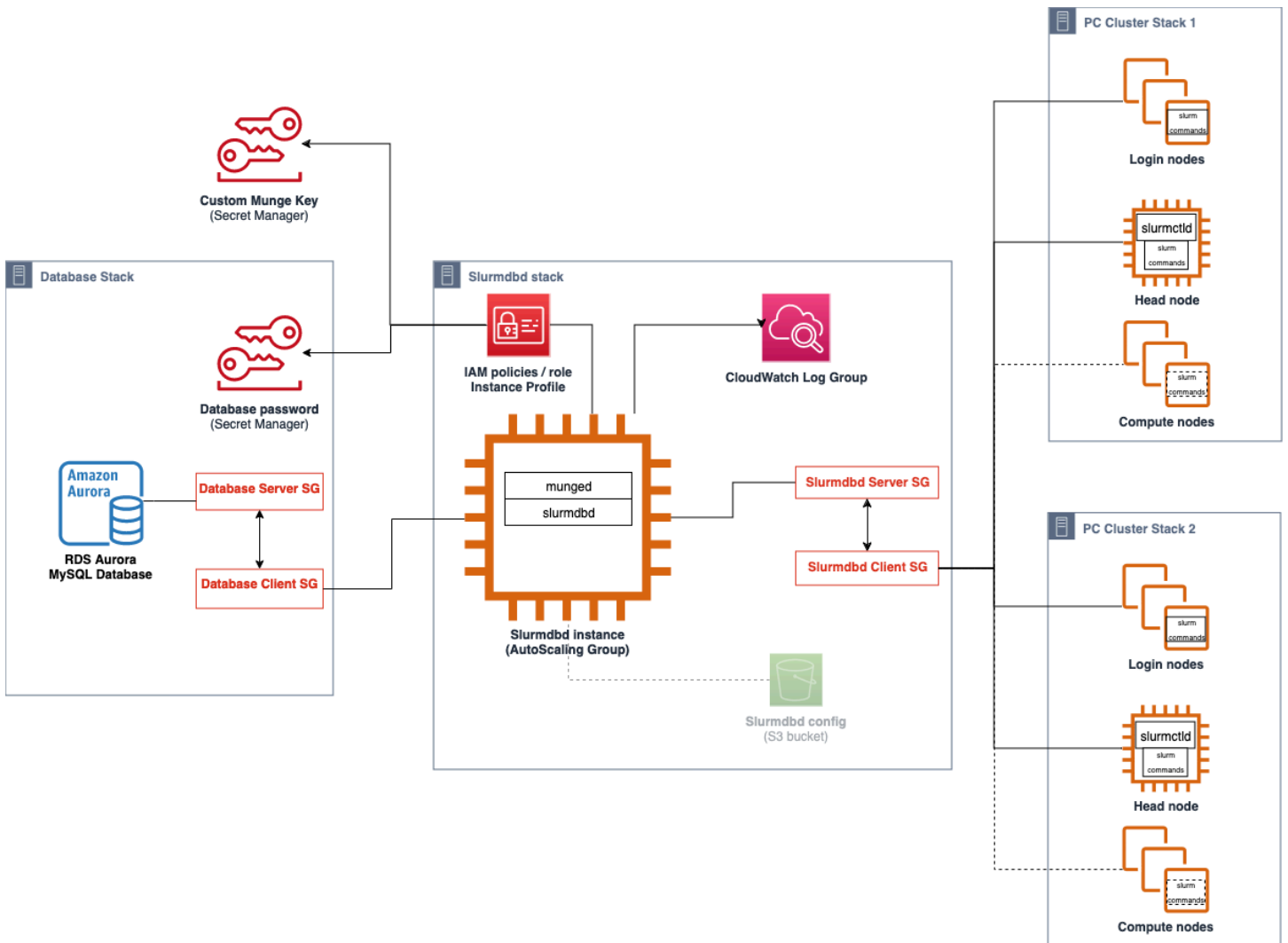
- 에서 외부 Slurmdbd 서버와 통신하기 위한 Munge 키입니다. [MungeKeySecretArn](#)

튜토리얼을 단계별로 진행하려면 [외부 Slurmdbd 어카운팅을 사용하여 클러스터 생성](#)을 참조하세요.

**Note**

Slurm데이터베이스 회계 엔티티를 관리할 책임은 귀하에게 있습니다.

AWS ParallelCluster 외부 SlurmDB 지원 기능의 아키텍처를 통해 여러 클러스터가 SlurmDB 동일하고 동일한 데이터베이스를 공유할 수 있습니다.



**⚠ Warning**

외부 AWS ParallelCluster 트래픽과 외부 SlurmDB 트래픽은 암호화되지 않습니다. 클러스터와 외부 네트워크는 신뢰할 수 SlurmDB 있는 네트워크에서 실행하는 것이 좋습니다.

AWS ParallelCluster v3.3.0 Slurmdbd 이상에서 헤드 노드를 사용한 Slurm 어카운팅 작업

Slurm 회계를 구성하기 전에 mysql 프로토콜을 사용하는 기존 외부 데이터베이스 서버 및 데이터베이스가 있어야 합니다.

를 사용하여 Slurm 어카운팅을 AWS ParallelCluster 구성하려면 다음을 정의해야 합니다.

- [Database/Uri](#)에 있는 외부 데이터베이스 서버의 URI입니다. 서버가 존재하고 헤드 노드에서 연결할 수 있어야 합니다.
- [데이터베이스/PasswordSecretArn 및 데이터베이스 /에 정의된 외부 데이터베이스에 액세스하기 위한 UserName 자격 증명](#) AWS ParallelCluster 이 정보를 사용하여 헤드 노드에서 Slurm 레벨의 어카운팅과 slurmdbd 서비스를 구성합니다. slurmdbd 클러스터와 데이터베이스 서버 간의 통신을 관리하는 데몬입니다.

튜토리얼을 단계별로 진행하려면 [Slurm 회계를 사용하여 클러스터 생성](#)을 참조하세요.

**i Note**

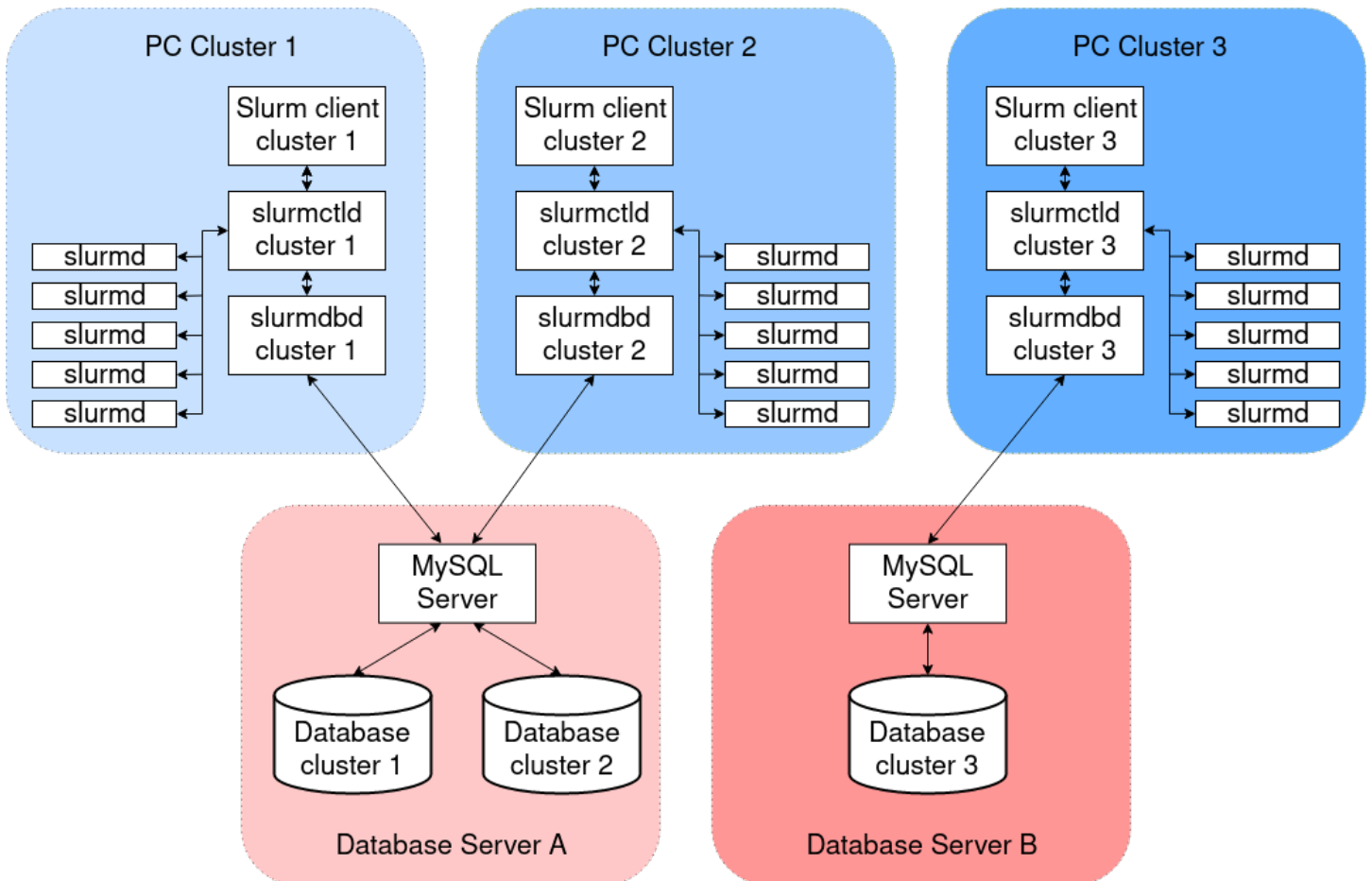
AWS ParallelCluster 데이터베이스의 기본 클러스터 사용자를 데이터베이스 관리자로 설정하여 Slurm 계정 데이터베이스의 기본 부트스트랩을 수행합니다. Slurm AWS ParallelCluster 계정 데이터베이스에 다른 사용자를 추가하지 않습니다. 고객은 Slurm 데이터베이스의 회계 엔터티를 관리할 책임이 있습니다.

AWS ParallelCluster 클러스터가 데이터베이스 서버에 자체 Slurm 데이터베이스를 [slurmdbd](#) 갖도록 구성합니다. 여러 클러스터에서 동일한 데이터베이스 서버를 사용할 수 있지만 각 클러스터에는 별도의 데이터베이스가 있습니다. AWS ParallelCluster 클러스터 이름을 사용하여 slurmdbd 구성 파일 [StorageLoc](#) 매개 변수에 데이터베이스 이름을 정의합니다. 다음 상황을 고려하세요. 데이터베이스 서버에 있는 데이터베이스에는 활성 클러스터 이름에 매핑되지 않는 클러스터 이름이 포함되어 있습니다. 이 경우 해당 클러스터 이름으로 새 클러스터를 만들어 해당 데이터베이스에 매핑할 수 있습니다. Slurm은 데이터베이스를 새 클러스터에 재사용합니다.

**Warning**

- 한 번에 같은 데이터베이스를 사용하기 위해 두 개 이상의 클러스터를 설정하지 않는 것이 좋습니다. 이렇게 하면 성능 문제가 발생하거나 데이터베이스 교착 상태가 발생할 수 있습니다.
- 클러스터의 헤드 노드에서 Slurm 회계를 활성화한 경우 강력한 CPU, 더 많은 메모리, 더 높은 네트워크 대역폭을 갖춘 인스턴스 유형을 사용하는 것이 좋습니다. Slurm 회계는 클러스터의 헤드 노드에 부담을 가중시킬 수 있습니다.

AWS ParallelCluster Slurm계정 기능의 현재 아키텍처에서는 다음 다이어그램 예제 구성에서 볼 수 있듯이 각 클러스터에는 자체 slurmdbd 데몬 인스턴스가 있습니다.



클러스터 환경에 사용자 지정 Slurm 다중 클러스터 또는 페더레이션 기능을 추가하는 경우 모든 클러스터가 동일한 slurmdbd 인스턴스를 참조해야 합니다. 이 대안의 경우 한 클러스터에서 AWS



ParallelCluster Slurm 어카운팅을 활성화하고 다른 클러스터는 첫 번째 클러스터에서 호스팅되는 클러스터에 연결하도록 수동으로 구성하는 것이 좋습니다. `slurmdbd`

버전 3.3.0 이전 AWS ParallelCluster 버전을 사용하는 경우 이 [HPC 블로그 게시물에](#) 설명된 Slurm 어카운팅을 구현하는 대체 방법을 참조하십시오.

## Slurm 회계 고려 사항

### 서로 다른 VPC의 데이터베이스 및 클러스터

Slurm 회계를 활성화하려면 `slurmdbd` 대몬(daemon)이 수행하는 읽기 및 쓰기 작업을 위한 백엔드 역할을 하는 데이터베이스 서버가 필요합니다. Slurm 회계를 활성화하도록 클러스터가 생성되거나 업데이트되기 전에 헤드 노드가 데이터베이스 서버에 연결할 수 있어야 합니다.

클러스터가 사용하는 VPC가 아닌 다른 VPC에 데이터베이스 서버를 배포해야 하는 경우 다음 사항을 고려하세요.

- 클러스터 측의 `slurmdbd`와 데이터베이스 서버 간의 통신을 활성화하려면 두 VPC 간의 연결을 설정해야 합니다. 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 [VPC 피어링](#)을 참조하세요.
- 클러스터의 VPC에 있는 헤드 노드에 연결할 보안 그룹을 만들어야 합니다. 두 VPC가 피어링된 후에는 데이터베이스 측과 클러스터 측 보안 그룹 간의 상호 연결을 사용할 수 있습니다. 자세한 내용을 알아보려면 Amazon Virtual Private Cloud 사용 설명서의 [보안 그룹 규칙](#)을 참조하세요.

### 데이터베이스 서버와 `slurmdbd` 간의 TLS 암호화 구성

를 AWS ParallelCluster 제공하는 기본 Slurm 계정 구성을 사용하면 서버가 TLS 암호화를 지원하는 경우 데이터베이스 서버에 대한 TLS 암호화 연결을 `slurmdbd` 설정합니다. AWS Amazon RDS와 같은 데이터베이스 서비스는 기본적으로 TLS 암호화를 Amazon Aurora 지원합니다.

데이터베이스 서버에서 `require_secure_transport` 파라미터를 설정하여 서버 측의 보안 연결을 요구할 수 있습니다. 이는 제공된 CloudFormation 템플릿에서 구성됩니다.

최상의 보안을 위해 `slurmdbd` 클라이언트에서 서버 ID 확인도 활성화하는 것이 좋습니다. 이렇게 하려면 [StorageParameters](#)에서 를 구성하십시오 `slurmdbd.conf`. 서버 CA 인증서를 클러스터의 헤드 노드에 업로드합니다. 그런 다음 [StorageParameters](#)의 [SSL\\_CA](#) 옵션을 헤드 노드의 서버 CA 인증서 `slurmdbd.conf` 경로로 설정합니다. 이렇게 하면 `slurmdbd` 측의 서버 ID 확인이 가능해집니다. 이러한 변경을 수행한 후에는 `slurmdbd` 서비스를 다시 시작하여 ID 검증이 활성화된 상태에서 데이터베이스 서버와의 연결을 다시 설정하세요.

## 데이터베이스 보안 인증 업데이트

[Database/UserNamePasswordSecretArn](#)의 값을 업데이트하려면 먼저 컴퓨팅 플릿을 중지해야 합니다. 시크릿에 저장된 시크릿 값은 변경되고 ARN은 변경되지 않았다고 가정해 봅시다. AWS Secrets Manager 이 경우 클러스터는 데이터베이스 비밀번호를 새 값으로 자동 업데이트하지 않습니다. 새 암호 값에 맞게 클러스터를 업데이트하려면 헤드 노드에서 다음 명령을 실행합니다.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

### Warning

계정 데이터 손실을 방지하려면 컴퓨팅 플릿이 중지된 경우에만 데이터베이스 비밀번호를 변경하는 것이 좋습니다.

## 데이터베이스 모니터링

AWS 데이터베이스 서비스의 모니터링 기능을 활성화하는 것이 좋습니다. 자세한 내용은 [Amazon RDS 모니터링](#) 또는 [Amazon Aurora 모니터링](#) 설명서를 참조하세요.

## Slurm 구성 사용자 지정

AWS ParallelCluster 버전 3.6.0부터 AWS ParallelCluster 클러스터 구성의 `slurm.conf` Slurm 구성을 사용자 지정할 수 있습니다.

클러스터 구성에서 다음 클러스터 구성 설정을 사용하여 Slurm 구성 파라미터를 사용자 지정할 수 있습니다.

- [SlurmSettings/CustomSlurmSettings](#) 또는 [CustomSlurmSettingsIncludeFile](#) 파라미터를 사용하여 전체 클러스터의 Slurm 파라미터를 사용자 지정합니다. 둘 다 지정하면 AWS ParallelCluster가 실패합니다.
- [SlurmQueues/CustomSlurmSettings](#)(Slurm 파티션에 매핑됨)를 사용하여 대기열의 Slurm 파라미터를 사용자 지정합니다.
- [SlurmQueues/ComputeResources/CustomSlurmSettings](#)(Slurm 노드에 매핑됨)를 사용하여 컴퓨팅 리소스의 Slurm 파라미터를 사용자 지정합니다.

## Slurm 구성 사용자 지정 제한 및 AWS ParallelCluster 사용 시 고려 사항

- CustomSlurmSettings 및 CustomSlurmSettingsIncludeFile 설정의 경우 클러스터를 구성하는 데 사용하는 AWS ParallelCluster 버전에서 지원하는 [Slurm 버전](#)에 포함된 slurm.conf 파라미터만 지정하고 업데이트할 수 있습니다.
- CustomSlurmSettings 파라미터에 사용자 지정 Slurm 구성을 지정하는 경우 AWS ParallelCluster가 유효성 검사를 수행하고 AWS ParallelCluster 로직과 충돌하는 Slurm 구성 파라미터를 설정하거나 업데이트하지 못하도록 합니다. AWS ParallelCluster와 충돌하는 것으로 알려진 Slurm 구성 파라미터는 거부 목록에서 식별됩니다. 다른 Slurm 특능이 추가되면 향후 AWS ParallelCluster 버전에서 거부 목록이 변경될 수 있습니다. 자세한 내용은 [CustomSlurmSettings을 위한 거부 목록에 등록된 Slurm 구성 파라미터](#) 항목을 참조하세요.
- AWS ParallelCluster는 파라미터가 거부 목록에 있는지 여부만 확인합니다. AWS ParallelCluster는 사용자 지정 Slurm 구성 파라미터 구문이나 의미 체계를 검증하지 않습니다. 사용자 지정 Slurm 구성 파라미터의 유효성을 검사하는 것은 사용자의 책임입니다. 잘못된 사용자 지정 Slurm 구성 파라미터로 인해 Slurm 대몬(daemon) 장애가 발생하여 클러스터 생성 및 업데이트 실패로 이어질 수 있습니다.
- CustomSlurmSettingsIncludeFile에서 사용자 지정 Slurm 구성을 지정하면 AWS ParallelCluster가 검증을 수행하지 않습니다.
- 컴퓨팅 플릿을 중지하고 시작하지 않고도 CustomSlurmSettings 및 CustomSlurmSettingsIncludeFile을 업데이트할 수 있습니다. 이 경우 AWS ParallelCluster는 slurmctld 대몬(daemon)을 다시 시작하고 scontrol reconfigure 명령을 실행합니다.

일부 Slurm 구성 파라미터에는 변경 내용이 전체 클러스터에 등록되기 전에 다른 작업이 필요할 수 있습니다. 예를 들어 클러스터의 모든 대몬(daemon)을 다시 시작해야 할 수 있습니다. 업데이트 중에 사용자 지정 Slurm 구성 파라미터 설정을 전파하기에 AWS ParallelCluster 작업이 충분한지 확인하는 것은 사용자의 책임입니다. AWS ParallelCluster작업이 충분하지 않다고 판단되는 경우 [Slurm 설명서](#)의 권장 사항에 따라 업데이트된 설정을 전파하는 데 필요한 추가 조치를 제공하는 것은 사용자의 책임입니다.

### CustomSlurmSettings을 위한 거부 목록에 등록된 Slurm 구성 파라미터

다음 표에는 버전 3.6.0부터 사용을 거부하는 AWS ParallelCluster 버전의 파라미터가 나열되어 있습니다. CustomSlurmSettings는 버전 3.6.0 이전의 AWS ParallelCluster 버전에서 지원되지 않습니다.

클러스터 수준에서 거부 목록에 등록된 파라미터 목록:

Slurm 파라미터	AWS ParallelCluster 버전별 거부 목록
CommunicationParameters	3.6.0
Epilog	3.6.0
GresTypes	3.6.0
LaunchParameters	3.6.0
Prolog	3.6.0
ReconfigFlags	3.6.0
ResumeFailProgram	3.6.0
ResumeProgram	3.6.0
ResumeTimeout	3.6.0
SlurmctldHost	3.6.0
SlurmctldLogFile	3.6.0
SlurmctldParameters	3.6.0
SlurmdLogfile	3.6.0
SlurmUser	3.6.0
SuspendExcNodes	3.6.0
SuspendProgram	3.6.0
SuspendTime	3.6.0
TaskPlugin	3.6.0
TreeWidth	3.6.0

클러스터 구성에서 [네이티브 Slurm 회계 통합](#)이 구성된 경우 클러스터 수준에서 거부 목록에 있는 파라미터:

Slurm 파라미터	AWS ParallelCluster 버전별 거부 목록
AccountingStorageType	3.6.0
AccountingStorageHost	3.6.0
AccountingStoragePort	3.6.0
AccountingStorageUser	3.6.0
JobAcctGatherType	3.6.0

AWS ParallelCluster가 관리하는 대기열의 대기열(파티션) 수준에서 거부 목록에 등록된 파라미터:

Slurm 파라미터	AWS ParallelCluster 버전별 거부 목록
노드	3.6.0
PartitionName	3.6.0
ResumeTimeout	3.6.0
State	3.6.0
SuspendTime	3.6.0

AWS ParallelCluster가 관리하는 컴퓨팅 리소스의 컴퓨팅 리소스(노드) 수준에서 거부 목록에 등록된 파라미터:

Slurm 파라미터	AWS ParallelCluster 버전 및 이후 버전에서는 거부 목록에 등록되어 있습니다.
CPU	3.6.0
특성	3.6.0
Gres	3.6.0

Slurm 파라미터	AWS ParallelCluster 버전 및 이후 버전에서는 거부 목록에 등록되어 있습니다.
NodeAddr	3.6.0
NodeHostname	3.6.0
NodeName	3.6.0
Weight	3.7.0

## Slurmprolog 및 epilog

AWS ParallelCluster 버전 3.6.0부터 배포되는 Slurm 구성에는 Prolog 및 Epilog 구성 AWS ParallelCluster 매개변수가 포함됩니다.

```
# PROLOG AND EPILOG
Prolog=/opt/slurm/etc/scripts/prolog.d/*
Epilog=/opt/slurm/etc/scripts/epilog.d/*
SchedulerParameters=nohold_on_prolog_fail
BatchStartTimeout=180
```

자세한 내용은 Slurm 설명서의 [Prolog 및 Epilog 안내서](#)를 참조하세요.

AWS ParallelCluster 다음과 같은 프로로그 및 에필로그 스크립트가 포함되어 있습니다.

- 90\_plcluster\_health\_check\_manager(Prolog 폴더 내)
- 90\_pcluster\_noop(Epilog 폴더 내)

### Note

Prolog 및 Epilog 폴더는 각각 적어도 하나 이상의 파일을 포함해야 합니다.

각 해당 Prolog 및 Epilog 폴더에 사용자 지정 prolog 또는 epilog 스크립트를 추가하여 사용할 수 있습니다.

**⚠ Warning**

Slurm은 폴더에 있는 모든 스크립트를 알파벳 역순으로 실행합니다.

prolog 및 epilog 스크립트의 실행 시간은 작업을 실행하는 데 필요한 시간에 영향을 줍니다. 여러 스크립트 또는 장기 실행 prolog 스크립트를 실행하는 경우 BatchStartTimeout 구성 설정을 업데이트하세요. 기본값은 3분입니다.

사용자 지정 prolog 및 epilog 스크립트를 사용하는 경우 각 Prolog 및 Epilog 폴더에서 스크립트를 찾으세요. 모든 사용자 지정 스크립트보다 먼저 실행되는 90\_plcluster\_health\_check\_manager 스크립트를 유지하는 것이 좋습니다. 자세한 내용은 [Slurm 구성 사용자 지정](#)(를) 참조하세요.

## 클러스터 용량 크기 및 업데이트

클러스터의 용량은 클러스터가 확장할 수 있는 컴퓨팅 노드의 수로 정의됩니다. 컴퓨팅 노드는 AWS ParallelCluster (Scheduling/SlurmQueues/[ComputeResources](#)) 구성의 컴퓨팅 리소스 내에 정의된 Amazon EC2 인스턴스에 의해 지원되며 파티션에 1:1 로 (Scheduling/[SlurmQueues](#)) 매핑되는 대기열로 구성됩니다. Slurm

[컴퓨팅 리소스 내에서 클러스터에서 항상 실행되어야 하는 최소 컴퓨팅 노드 \(인스턴스\) 수 \(\) 와 컴퓨팅 리소스가 \(3MinCount\) 까지 확장할 수 있는 최대 인스턴스 수를 구성할 수 있습니다. MaxCount](#)

클러스터 생성 시 또는 클러스터 업데이트 시 클러스터에 정의된 각 컴퓨팅 리소스 (Scheduling/SlurmQueues/ [ComputeResources](#) ) 에 MinCount 대해 구성된 수만큼 Amazon EC2 인스턴스를 AWS ParallelCluster 시작합니다. 클러스터의 컴퓨팅 리소스에 필요한 최소한의 노드를 포함하도록 시작된 인스턴스를 정적 노드라고 합니다. 정적 노드는 일단 시작되면 클러스터에서 영구 유지되며 특정 이벤트나 상황이 발생하지 않는 한 시스템에서 종료되지 않습니다. 이러한 이벤트에는 Amazon EC2 상태 점검 실패, Slurm 노드 상태가 DRAIN Slurm 또는 DOWN으로 변경되는 경우가 포함됩니다.

클러스터의 증가된 부하를 **1** 처리하기 위해 '**MaxCount - MinCount**' 온디맨드로 시작되는 **MaxCount ~**마이너스 **MinCount**) 범위의 Amazon EC2 인스턴스를 동적 노드라고 합니다. 그 특성은 일시적이며 보류 중인 작업을 처리하기 위해 시작되며 클러스터 Scheduling/SlurmSettings/[ScaledownIdletime](#) 구성에서 정의한 기간 (기본값: 10분) 동안 유휴 상태를 유지하면 종료됩니다.

정적 노드와 동적 노드는 다음 이름 지정 스키마를 준수합니다.

- 정적 노드는 다음과 같습니다 <Queue/Name>-st-<ComputeResource/Name>-<num>. <num> = 1..ComputeResource/MinCount
- 동적 노드, <Queue/Name>-dy-<ComputeResource/Name>-<num> 위치 <num> = 1.. (ComputeResource/MaxCount - ComputeResource/MinCount)

예를 들어 다음과 같은 AWS ParallelCluster 구성이 제공됩니다.

```
Scheduling:
  Scheduler: Slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

에 다음과 같은 노드가 정의됩니다. Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up      infinite  100    idle queue1-st-c5xlarge-[1-100]
```

컴퓨팅 리소스에 리소스가 MinCount == MaxCount 있는 경우 해당하는 모든 컴퓨팅 노드는 정적이며 클러스터 생성/업데이트 시 모든 인스턴스가 시작되고 계속 실행됩니다. 예:

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
```



```
MinCount: 100
MaxCount: 100
```

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite   100   idle queue1-st-c5xlarge-[1-100]
```

## 클러스터 용량 업데이트

클러스터 용량 업데이트에는 대기열 추가 또는 제거, 컴퓨팅 리소스, 컴퓨팅 리소스 변경이 포함됩니다. MinCount/MaxCount AWS ParallelCluster 버전 3.9.0부터 대기열 크기를 줄이려면 클러스터 업데이트를 실행하기 전에 컴퓨팅 풀릿을 중지하거나 TERMINATE로 [QueueUpdateStrategy](#) 설정해야 합니다. 다음과 같은 경우에는 컴퓨팅 풀릿을 중지하거나 TERMINATE로 설정할 [QueueUpdateStrategy](#) 필요가 없습니다.

- 스케줄링에 새 대기열 추가/ [SlurmQueues](#)
- 대기열에 새 컴퓨팅 리소스 추가 Scheduling/SlurmQueues/[ComputeResources](#)
- 컴퓨팅 리소스 증가 [MaxCount](#)
- 컴퓨팅 리소스 증가 MaxCount 및 동일 컴퓨팅 리소스 최소 양 증가 MinCount

## 고려 사항 및 제한

이 섹션은 클러스터 용량 크기를 조정할 때 고려해야 하는 중요한 요소, 제약 조건 또는 제한 사항을 간략하게 설명하기 위한 것입니다.

- 이름이 있는 Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues 모든 컴퓨팅 노드에서 대기열을 제거하면 정적 및 동적 이름이 <Queue/Name>-\* Slurm 구성에서 제거되고 해당 Amazon EC2 인스턴스가 종료됩니다.
- Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues-ComputeResources> ComputeResources 대기열에서 컴퓨팅 리소스를 제거하면 정적 및 동적 이름을 <Queue/Name>-\*-<ComputeResource/Name>-\* 가진 모든 컴퓨팅 노드가 Slurm 구성에서 제거되고 해당 Amazon EC2 인스턴스가 종료됩니다.

컴퓨팅 리소스의 MinCount 파라미터를 변경할 때 두 가지 시나리오를 구분할 수 있습니다.

MaxCount 하나는 같게 유지되고 MinCount (정적 용량만 해당) 이면 MinCount (정적 용량과 동적 용량이 혼합된 경우 MaxCount) 보다 크다는 것입니다.

정적 노드에서만 용량이 변경됩니다.

- 증가 MinCount (및 MaxCount) 시 정적 노드 수를 새 값까지 확장하여 클러스터를 구성하고 시스템은 새로운 필수 정적 용량을 충족하기 위해 Amazon EC2 인스턴스를 계속 시작하려고 시도합니다. MinCount == MaxCount MinCount < Queue/Name>-st-<ComputeResource/Name>-<new\_MinCount>
- 양 N을 줄이면 MinCount (및 MaxCount) 마지막 N개의 정적 노드를 <Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount - N>...<old\_MinCount>] 제거하여 클러스터가 구성되고 시스템은 해당 Amazon EC2 인스턴스를 종료합니다. MinCount == MaxCount
- 초기 상태 MinCount = MaxCount = 100

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   100   idle queue1-st-c5xlarge-[1-100]
```

- MinCount 및 -30 에서 업데이트 MaxCount: MinCount = MaxCount = 70

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite    70   idle queue1-st-c5xlarge-[1-70]
```

## 혼합 노드의 용량 변경

N만큼 MinCount 증가하면 MinCount < MaxCount (변경되지 않은 상태로 유지된다고 가정 MaxCount) 정적 노드 수가 새 값인 MinCount (old\_MinCount + N) :까지 확장되어 클러스터가 <Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N> 구성되고 시스템은 새로운 필수 정적 용량을 충족하기 위해 Amazon EC2 인스턴스를 계속 시작하려고 시도합니다. 또한 컴퓨팅 리소스의 MaxCount 용량을 유지하기 위해 마지막 N개의 동적 노드를 제거하여 클러스터 구성을 업데이트합니다 <Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount -

old\_MinCount - N>...<MaxCount - old\_MinCount>]. 그러면 시스템이 해당 Amazon EC2 인스턴스를 종료합니다.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle  queue1-st-c5xlarge-[1-100]
```

- +30으로 업데이트 MinCount : MinCount = 130 (MaxCount = 150)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*    up    infinite  130    idle  queue1-st-c5xlarge-[1-130]
```

같은 양 N을 MinCount 늘리면 정적 노드 수를 새 값인 MinCount (old\_MinCount + N) <Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N> :까지 확장하여 클러스터를 구성하고 시스템은 새로운 필수 정적 용량을 충족하기 위해 Amazon EC2 인스턴스를 계속 시작하려고 시도합니다. MinCount < MaxCount MaxCount 또한 새 노드를 적용하기 위해 동적 노드 수는 변경되지 않습니다.

MaxCount USD 상당.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle  queue1-st-c5xlarge-[1-100]
```

- +30으로 업데이트 MinCount : MinCount = 130 (MaxCount = 180)

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

N개의 양을 MinCount 줄이면 (변경되지 않은 상태로 유지된다고 가정 MaxCount) 마지막 N개의 정적 노드 정적 노드를 <Queue/Name>-st-<ComputeResource/Name>-[<old\_MinCount - N>...<old\_MinCount> 제거하여 클러스터를 구성하고 시스템은 해당 Amazon EC2 인스턴스를 종료합니다. MinCount < MaxCount 또한 컴퓨팅 리소스의 MaxCount 용량을 늘리기 위해 동적 노드 수를 늘려 격차를 메우는 MaxCount - new\_MinCount: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new\_MinCount>] 방식으로 클러스터 구성을 업데이트합니다. 이 경우 동적 노드는 동적 노드이므로 스케줄러가 새 노드에 보류 중인 작업이 없는 한 새 Amazon EC2 인스턴스가 시작되지 않습니다.

- 초기 상태: MinCount = 100; MaxCount = 150

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 업데이트 -30 켜기 MinCount : MinCount = 70 (MaxCount = 120)

- ```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   80    idle~ queue1-dy-c5xlarge-[1-80]
queue1*   up    infinite   70    idle queue1-st-c5xlarge-[1-70]
```

N개를 MinCount 줄이면 마지막 N개의 정적 노드를 <Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount - N>...<oldMinCount>] 제거하여 클러스터를 구성하고 시스템은 해당 Amazon EC2 인스턴스를 종료합니다. MinCount < MaxCount MaxCount

또한 새 MaxCount 값을 적용하기 위한 동적 노드 수는 변경되지 않습니다.

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 업데이트 -30 켜기 MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   80    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite   70    idle queue1-st-c5xlarge-[1-70]
```

N개의 양을 줄이면 MaxCount (변경되지 않은 상태로 유지된다고 가정MinCount) 마지막 N개의 동적 노드를 <Queue/Name>-dy-<ComputeResource/Name>-<old\_MaxCount - N...<oldMaxCount>] 제거하여 클러스터를 구성하고 시스템은 해당 Amazon EC2 인스턴스가 실행 중일 때 해당 Amazon EC2 인스턴스를 종료합니다. 정적 노드에는 영향이 없을 것으로 예상됩니다. MinCount < MaxCount

- 초기 상태: MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- 업데이트 -30 켜기 MaxCount : MinCount = 100 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up      infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up      infinite  100    idle  queue1-st-c5xlarge-[1-100]
```

## 작업에 미치는 영향

노드가 제거되고 Amazon EC2 인스턴스가 종료되는 모든 경우, 작업 요구 사항을 충족하는 다른 노드가 없는 한 제거된 노드에서 실행 중인 sbatch 작업은 대기열에 다시 추가됩니다. 마지막 경우에는 작업이 NODE\_FAIL 상태로 실패하고 대기열에서 사라집니다. 이 경우 수동으로 다시 제출해야 합니다.

클러스터 크기 조정 업데이트를 수행하려는 경우 계획된 업데이트 중에 제거될 노드에서 작업이 실행되지 않도록 할 수 있습니다. 이는 유지 관리 시 제거될 노드를 설정하여 가능합니다. 유지 관리 중인 노드를 설정해도 결국 해당 노드에서 이미 실행 중인 작업에는 영향을 미치지 않는다는 점에 유의하십시오.

계획된 클러스터 크기 조정 업데이트를 통해 해당 노드를 queue-st-computeresource-[9-10] 제거한다고 가정해 보겠습니다.] 다음 명령으로 Slurm 예약을 생성할 수 있습니다.

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queue-st-
computeresource-[9-10]
```

그러면 maint\_for\_update 노드에 이름이 지정된 Slurm 예약이 생성됩니다 queue-st-computeresource-[9-10]. 예약이 생성된 시점부터 더 이상 노드에 작업이 실행될 수 없습니다 queue-st-computeresource-[9-10]. 예약을 한다고 해서 결국 노드에 작업이 할당되는 것을 막지는 못한다는 점에 queue-st-computeresource-[9-10] 유의하십시오.

클러스터 크기 조정 업데이트 후 크기 조정 업데이트 중에 제거된 노드에만 Slurm 예약이 설정된 경우 유지 관리 예약은 자동으로 삭제됩니다. 대신 클러스터 크기 조정 업데이트 후에도 여전히 존재하는 노드에 대한 Slurm 예약을 생성한 경우, 크기 조정 업데이트를 수행한 후 다음 명령을 사용하여 노드의 유지 관리 예약을 제거할 수 있습니다.

```
sudo -i scontrol delete ReservationName=maint_for_update
```

Slurm [예약에 대한 자세한 내용은 여기에서 공식 SchedMD 문서를 참조하십시오.](#)

용량 변경에 대한 클러스터 업데이트 프로세스

스케줄러 구성이 변경되면 클러스터 업데이트 프로세스 중에 다음 단계가 실행됩니다.

- 중지 AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- 구성에서 업데이트된 Slurm 파티션 AWS ParallelCluster 구성 생성
- 재시작 `slurmctld` (Chef 서비스 레시피를 통해 완료)
- `slurmctld` 상태 확인 (`systemctl is-active --quiet slurmctld.service`)
- 구성 다시 불러오기 Slurm (`scontrol reconfigure`)
- `clustermgtd` (`supervisorctl start clustermgtd`) 시작

## AWS Batch (**awsbatch**)

AWS Batch에 대한 자세한 내용은 [AWS Batch](#) 섹션을 참조하세요. 설명서는 [AWS Batch 사용 설명서](#)를 참조하세요.

AWS Batch에 대한 AWS ParallelCluster CLI 명령

`awsbatch` 스케줄러를 사용하는 경우 AWS Batch에 대한 AWS ParallelCluster CLI 명령이 AWS ParallelCluster 헤드 노드에 자동으로 설치됩니다. CLI는 AWS Batch API 작업을 사용하며 다음 작업을 허용합니다.

- 작업 제출 및 관리
- 작업, 대기열 및 호스트 모니터링
- 기존 스케줄러 명령 미러링

### Important

AWS ParallelCluster는 AWS Batch에 대한 GPU 작업은 지원하지 않습니다. 자세한 내용은 [GPU 작업을 참조하세요.](#)

이 CLI는 별도의 패키지로 배포됩니다. 자세한 내용은 [스케줄러 지원](#) 항목을 참조하세요.

주제

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)
- [awsbqueues](#)
- [awsbhosts](#)

## awsbsub

작업을 클러스터의 작업 대기열에 제출합니다.

```
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 에 대한 AWS Batch GPU 작업은 지원하지 않습니다. 자세한 내용은 [GPU 작업을 참조하세요](#).

위치 인수

### **command**

작업을 제출(지정된 명령이 컴퓨팅 인스턴스에서 사용 가능해야 함)하거나 전송할 파일 이름을 지정합니다. --command-file 섹션도 참조하십시오.

### **arguments**

(선택 사항) 명령 또는 명령 파일의 인수를 지정합니다.



이름 지정된 인수

**-jn *JOB\_NAME*, --job-name *JOB\_NAME***

작업 이름을 지정합니다. 첫 번째 자리는 문자 또는 숫자여야 합니다. 작업 이름은 최대 128자까지 포함할 수 있으며, 대문자와 소문자, 숫자, 하이픈(-), 밑줄(\_)을 포함할 수 있습니다.

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

사용할 클러스터를 지정합니다.

**-cf, --command-file**

명령이 컴퓨팅 인스턴스로 전송될 파일임을 나타냅니다.

기본값: False

**-w *WORKING\_DIR*, --working-dir *WORKING\_DIR***

작업의 작업 디렉터리로 사용할 폴더를 지정합니다. 작업 디렉터리가 지정되지 않으면 작업이 사용자의 홈 디렉터리에 있는 `job-<AWS_BATCH_JOB_ID>` 하위 폴더에서 실행됩니다. 이 파라미터 또는 `--parent-working-dir` 파라미터를 사용할 수 있습니다.

**-pw *PARENT\_WORKING\_DIR*, --parent-working-dir *PARENT\_WORKING\_DIR***

작업의 작업 디렉터리에서 상위 폴더를 지정합니다. 상위 작업 디렉터리가 지정되지 않은 경우, 사용자의 홈 디렉터리가 기본적으로 지정됩니다. 상위 작업 디렉터리에 `job-<AWS_BATCH_JOB_ID>`라는 하위 폴더가 만들어집니다. 이 파라미터 또는 `--working-dir` 파라미터를 사용할 수 있습니다.

**-if *INPUT\_FILE*, --input-file *INPUT\_FILE***

작업의 작업 디렉터리에서 컴퓨팅 인스턴스로 전송할 파일을 지정합니다. 여러 입력 파일 파라미터를 지정할 수 있습니다.

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

컨테이너를 위해 예약할 vCPU 개수를 지정합니다. `-nodes`와 함께 사용할 경우 노드당 vCPU 수를 식별합니다.

기본값: 1

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

작업에 제공할 메모리의 하드 제한(MiB)을 지정합니다. 작업에서 여기서 지정된 메모리 제한을 초과하려고 하면 해당 작업이 종료됩니다.

기본값: 128

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

작업 환경으로 내보낼 환경 변수 이름의 목록을 쉼표로 구분하여 지정합니다. 모든 환경 변수를 내보내려면 'all'을 지정하세요. `-env-blacklist` 파라미터에 나열된 변수, 또는 `PCLUSTER_*`나 `AWS_*`로 시작하는 변수는 'all' 환경 변수 목록에 포함되지 않습니다.

**-eb ENV\_DENYLIST, --env-blacklist ENV\_DENYLIST**

작업 환경으로 내보내지 않을 환경 변수 이름의 목록을 쉼표로 구분하여 지정합니다. 기본적으로, HOME, PWD, USER, PATH, LD\_LIBRARY\_PATH, TERM 및 TERMCAP은 내보내지 않습니다.

**-r RETRY\_ATTEMPTS, --retry-attempts RETRY\_ATTEMPTS**

작업을 RUNNABLE 상태로 전환하는 횟수를 지정합니다. 1부터 10까지 시도 횟수를 지정할 수 있습니다. 시도 횟수가 1보다 큰 경우 작업이 실패하면 RUNNABLE 상태로 전환될 때까지 지정된 횟수만큼 다시 시도됩니다.

기본값: 1

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

작업이 완료되지 않은 경우 작업이 AWS Batch 종료되는 기간을 초 단위로 지정합니다 (작업 시도의 `startedAt` 타임스탬프에서 측정). 제한 시간 값은 60초 이상이어야 합니다.

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

작업을 위해 예약할 노드 수를 지정합니다. 다중 노드 병렬 제출을 사용하려면 이 파라미터의 값을 지정합니다.

#### Note

[Scheduler/AwsBatchQueues/CapacityType](#) 매개 변수를 SPOT으로 설정하면 다중 노드 병렬 작업이 지원되지 않습니다. 또한 사용자 계정에는 `AWSServiceRoleForEC2Spot` 서비스 연결 역할이 있어야 합니다. 다음 AWS CLI 명령으로 이 역할을 생성할 수 있습니다.

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

자세한 내용은 Linux [인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 스팟 인스턴스 요청에 대한 서비스 연결 역할](#)을 참조하십시오.

**-a *ARRAY\_SIZE*, --array-size *ARRAY\_SIZE***

배열의 크기를 지정합니다. 2~10,000 범위의 값을 지정할 수 있습니다. 작업에 배열 속성을 지정하면 배열 작업이 됩니다.

**-d *DEPENDS\_ON*, --depends-on *DEPENDS\_ON***

작업에 대해 세미콜론으로 구분된 종속성 목록을 지정합니다. 작업은 최대 20개의 작업에 종속될 수 있습니다. 배열 작업의 작업 ID를 지정하지 않고 SEQUENTIAL 유형의 종속성을 지정할 수 있습니다. 순차 종속성을 사용하면 각 하위 배열 작업을 인덱스 0부터 순차적으로 완료할 수 있습니다. 배열 작업의 작업 ID로 N\_TO\_N 유형의 종속성을 지정할 수도 있습니다. N\_TO\_N 종속성이란 이 작업의 각 인덱스 하위 항목은 각 종속성의 해당 인덱스 하위 항목이 완료될 때까지 기다린 후에만 시작할 수 있다는 의미입니다. 이 파라미터의 구문은 "jobId=<string>,type=<string>;..."입니다.

**awsbstat**

클러스터의 작업 대기열에 제출된 작업을 표시합니다.

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

위치 인수

***job\_ids***

출력에 표시할 작업 ID 목록을 공백으로 구분하여 지정합니다. 작업이 작업 배열이면 모든 하위 작업이 표시됩니다. 단일 작업이 요청되면 자세한 버전으로 표시됩니다.

이름 지정된 인수

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

사용할 클러스터를 지정합니다.

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

포함할 작업 상태 목록을 쉼표로 구분하여 지정합니다. 기본 작업 상태는 “활성”입니다. 허용되는 값: SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING, SUCCEEDED, FAILED 및 ALL

기본값: “SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING”

**-e, --expand-children**

하위 항목(배열 및 다중 노드 병렬 모두)을 사용하여 작업을 확장합니다.

기본값: False

## **-d, --details**

작업 세부 정보를 표시합니다.

기본값: False

## **awsbout**

지정된 작업의 출력을 표시합니다.

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

위치 인수

### ***job\_id***

작업 ID를 지정합니다.

이름 지정된 인수

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

사용할 클러스터를 지정합니다.

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

작업 출력의 첫 번째 *HEAD* 행을 가져옵니다.

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

작업 출력의 마지막 <tail> 행을 가져옵니다.

### **-s, --stream**

작업 출력을 가져온 다음 추가 출력이 생성될 때까지 대기합니다. 이 인수는 -tail과 함께 사용하여 작업 출력의 마지막 <tail> 행에서 시작할 수 있습니다.

기본값: False

### **-sp *STREAM\_PERIOD*, --stream-period *STREAM\_PERIOD***

스트리밍 기간을 설정합니다.

기본값: 5

## awsbkill

클러스터에 제출된 작업을 취소하거나 종료합니다.

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

위치 인수

### *job\_ids*

작업을 취소하거나 종료할 작업 ID 목록을 공백으로 구분하여 지정합니다.

이름 지정된 인수

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

사용할 클러스터의 이름을 나타냅니다.

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

작업에 첨부할 메시지를 표시하고 취소 이유를 설명합니다.

기본값: "Terminated by the user"

## awsbqueues

클러스터와 연관된 작업 대기열을 표시합니다.

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

위치 인수

### *job\_queues*

표시할 대기열 이름의 목록을 공백으로 구분하여 지정합니다. 단일 대기열이 요청되면 자세한 버전으로 표시됩니다.

이름 지정된 인수

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

사용할 클러스터의 이름을 지정합니다.

**-d, --details**

대기열의 세부 정보 표시 여부를 나타냅니다.

기본값: False

## awsbhosts

클러스터의 컴퓨팅 환경에 속한 호스트를 표시합니다.

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

위치 인수

***instance\_ids***

공백으로 구분된 인스턴스 ID 목록을 지정합니다. 단일 인스턴스가 요청되면 자세한 버전으로 표시됩니다.

이름 지정된 인수

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

사용할 클러스터의 이름을 지정합니다.

**-d, --details**

호스트의 세부 정보 표시 여부를 나타냅니다.

기본값: False

## 공유 스토리지

AWS ParallelCluster [는 Amazon EBS, ONTAP용 FSx, OpenZFS 공유 스토리지 볼륨의 경우 FSx, Lustre 공유 스토리지 파일 시스템용 Amazon EFS 및 FSx 또는 파일 캐시를 사용할 수 있습니다.](#) [AWS](#)

[Well-Architected Framework 신뢰성 요소](#) 지침을 따르고 볼륨과 파일 시스템을 백업하는 것이 좋습니다.

HPC 애플리케이션 I/O 요구 사항을 충족하는 스토리지 시스템을 선택하세요. 특정 사용 사례에 따라 각 파일 시스템을 최적화할 수 있습니다. 자세한 내용은 [스토리지 옵션 개요](#)를 참조하세요.

Amazon EBS 볼륨은 헤드 노드에 연결되고 NFS를 통해 컴퓨팅 노드와 공유됩니다. 이 옵션은 비용 효율적일 수 있지만 성능은 스토리지 요구 사항 규모 조정에 따른 헤드 노드 리소스에 따라 달라집니다. 클러스터에 더 많은 컴퓨팅 노드가 추가되고 처리량 수요가 증가하면 병목 현상이 발생할 수 있습니다.

Amazon EFS 파일 시스템은 스토리지 요구 사항의 변화에 따라 규모 조정됩니다. 다양한 사용 사례에 맞게 이러한 파일 시스템을 구성할 수 있습니다. Amazon EFS 파일 시스템을 사용하여 클러스터에서 병렬 처리되고 지연 시간에 민감한 애플리케이션을 실행할 수 있습니다.

FSx for Lustre 파일 시스템은 초당 최대 수백 기가바이트의 처리량, 수백만 IOPS 및 1밀리초 미만의 지연 시간으로 대규모 데이터 세트를 처리할 수 있습니다. 까다로운 고성능 컴퓨팅 환경을 위해 FSx for Lustre 파일 시스템을 사용하세요.

에서 외부 또는 [SharedStorage 섹션](#) 관리형 스토리지를 정의할 수 있습니다. AWS ParallelCluster

- 외부 스토리지는 관리하는 기존 볼륨 또는 파일 시스템을 말합니다. AWS ParallelCluster 이 스토리지를 만들거나 삭제하지 않습니다.
- 관리형 스토리지는 AWS ParallelCluster 생성하여 삭제할 수 있는 볼륨 또는 파일 시스템을 말합니다.

## 외부 스토리지

클러스터를 만들거나 업데이트할 때 외부 스토리지를 클러스터에 AWS ParallelCluster 연결하도록 구성할 수 있습니다. 마찬가지로 클러스터가 삭제되거나 업데이트될 때 외부 스토리지를 클러스터에서 분리하도록 구성할 수 있습니다. 데이터는 보존되며 클러스터 수명 주기 외의 기간에도 장기간 영구 공유 스토리지로 사용할 수 있습니다.

### Note

3.8 AWS ParallelCluster 이전 버전에서는 외부에서 관리되는 파일 시스템을 마운트할 수 없습니다. /home 버전 3.8부터는 외부 관리 파일 시스템의 마운트 /home 지점으로 사용할 AWS ParallelCluster 수 있습니다. 에서 아래에 있는 매개 변수에 값을 /home /home 지정하여 외부 관리 파일 시스템을 마운트할 수 있습니다. [MountDirSharedStorage 섹션](#)  
Amazon File Cache는 시스템 /home 디렉터리로 사용하기에 적합하지 않으므로 현재 마운트가 지원되지 않습니다/home.

[SharedStorageType](#) 구성 옵션에서 /home 디렉터리를 지정하는 경우 [SharedStorage](#) 섹션이 옵션이 재정의됩니다. 즉, 에 있는 설정이 [SharedStorage](#) 섹션 대신 사용됩니다. 외부 파일 시스템을 /home 디렉토리에 마운트하면 외부 스토리지의 기존 파일을 덮어쓰지 않고 헤드 노드의 /home 내용이 외부 파일 시스템에 AWS ParallelCluster 복사됩니다. 여기에는 외부 파일 시스템에 없는 경우 기본 사용자를 위해 클러스터의 SSH 키를 전송하는 것도 포함됩니다. 자세한 내용은 [을 참조하십시오. AWS ParallelCluster 공유 스토리지 고려사항](#)

## AWS ParallelCluster 관리형 스토리지

AWS ParallelCluster 관리형 스토리지는 기본적으로 구성의 클러스터 수명 주기에 따라 달라집니다. 기본적으로 SharedStorage DeletionPolicy 구성 파라미터는 Delete로 설정됩니다.

기본적으로 다음 중 하나에 해당하는 경우 AWS ParallelCluster 관리되는 파일 시스템 또는 볼륨과 해당 데이터가 삭제됩니다.

- 사용자가 클러스터를 삭제합니다.
- 사용자가 관리형 공유 스토리지 구성 Name을 변경합니다.
- 사용자가 구성에서 관리형 공유 스토리지를 제거합니다.

관리형 공유 파일 시스템 또는 볼륨 및 데이터를 유지하도록 DeletionPolicy를 Retain으로 설정합니다. 데이터 손실을 방지하려면 데이터를 정기적으로 백업하는 것이 좋습니다. [AWS Backup](#)을 사용하여 모든 스토리지 옵션의 백업을 중앙에서 관리할 수 있습니다.

구성 설정을 사용하여 라이프 사이클 종속성을 제거할 수 있습니다. 자세한 정보는 [AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 전환합니다.](#)을 참조하세요.

공유 스토리지 할당량에 대한 자세한 내용은 [공유 스토리지 할당량](#) 섹션을 참조하세요.

공유 스토리지 및 새 AWS ParallelCluster 버전으로의 전환에 대한 자세한 내용은 [을 참조하십시오. 모범 사례: 클러스터를 새 AWS ParallelCluster 마이너 또는 패치 버전으로 이동.](#)

클러스터를 만들거나 업데이트할 때 외부 스토리지를 클러스터에 AWS ParallelCluster 연결하도록 구성할 수 있습니다. 마찬가지로 클러스터가 삭제되거나 업데이트될 때 클러스터에서 외부 스토리지를 분리하도록 구성할 수 있습니다. 데이터는 보존되며 클러스터 수명 주기에 구애받지 않는 장기 영구 공유 스토리지 솔루션에 사용할 수 있습니다.



기본적으로 관리형 스토리지는 클러스터의 수명 주기에 따라 달라집니다. [AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 전환합니다.](#)에 설명된 구성 설정을 사용하여 종속성을 제거할 수 있습니다.

특정 설정을 통해 지원되는 각 스토리지 솔루션을 사용 사례에 맞게 최적화할 수 있습니다.

공유 스토리지 할당량에 대해서는 [공유 스토리지 할당량](#) 섹션을 참조하세요.

공유 스토리지 및 새 AWS ParallelCluster 버전으로의 전환에 대한 자세한 내용은 [을 참조하십시오 모범 사례: 클러스터를 새 AWS ParallelCluster 마이너 또는 패치 버전으로 이동.](#)

주제

- [공유된 스토리지 구성](#)
- [에서 공유 스토리지로 작업하기 AWS ParallelCluster](#)
- [공유 스토리지 할당량](#)

## 공유된 스토리지 구성

클러스터의 공유 스토리지를 정의하는 데 사용할 수 있는 구성 설정에 대해 알아봅니다.

주제

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx for Lustre](#)
- [FSx for ONTAP, FSx for OpenZFS 및 파일 캐시 공유 스토리지를 구성합니다.](#)

## Amazon Elastic Block Store

클러스터 수명 주기에 구매받지 않는 장기 영구 스토리지에 기존 외부 Amazon EBS 볼륨을 사용하려면 [EbsSettings/VolumeId](#)를 지정하세요.

[VolumeId](#)를 지정하지 않으면 기본적으로 클러스터가 생성될 때 [EbsSettings](#)에서 AWS ParallelCluster 가 관리형 EBS 볼륨을 생성합니다. AWS ParallelCluster 또한 클러스터가 삭제되거나 클러스터 구성에서 볼륨이 제거될 때 볼륨과 데이터를 삭제합니다.

AWS ParallelCluster 관리형 EBS 볼륨의 경우 [EbsSettingsDeletionPolicy/](#)를 사용하여 명령을 AWS ParallelCluster 내리거나 Snapshot 클러스터가 삭제되거나 볼륨이 클러스터 구성에서 제거될

때 볼륨을 사용할 수 있습니다. Delete Retain 기본적으로 DeletionPolicy는 Delete로 설정됩니다.

#### Warning

AWS ParallelCluster 관리형 공유 스토리지의 경우 DeletionPolicy 이 Delete 기본적으로 설정됩니다.

즉, 다음 중 하나에 해당하는 경우 관리형 볼륨과 해당 데이터가 삭제됩니다.

- 사용자가 클러스터를 삭제합니다.
- 관리형 공유 스토리지 구성 [SharedStorage/Name](#)을 변경합니다.
- 사용자가 구성에서 관리형 공유 스토리지를 제거합니다.

데이터 손실을 방지하려면 스냅샷을 사용하여 공유 스토리지를 정기적으로 백업하는 것이 좋습니다. Amazon EBS 스냅샷 생성에 대한 자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 [Amazon EBS 스냅샷](#)을 참조하세요. 전체에서 데이터 백업을 관리하는 방법을 AWS 서비스알아보려면 AWS Backup 개발자 안내서의AWS [Backup](#)을 참조하십시오.

## Amazon Elastic File System

클러스터 수명 주기 외의 장기간 영구 스토리지에 기존 외부 Amazon EFS 파일 시스템을 사용하려면 클러스터를 AWS ParallelCluster 생성할 [EfsSettings](#)때부터 기본적으로 [EfsSettingsFileSystemId](#)/를 지정하여 관리형 Amazon EFS 파일 시스템을 생성합니다. AWS ParallelCluster 또한 클러스터가 삭제되거나 클러스터 구성에서 파일 시스템이 제거될 때 파일 시스템과 데이터를 삭제합니다.

AWS ParallelCluster 관리형 Amazon EFS 파일 시스템의 경우 [EfsSettingsDeletionPolicy](#)/를 사용하여 클러스터를 Delete Retain 삭제하거나 클러스터 구성에서 파일 시스템을 AWS ParallelCluster 제거하도록 지시할 수 있습니다. 기본적으로 DeletionPolicy는 Delete로 설정됩니다.

#### Warning

AWS ParallelCluster 관리형 공유 스토리지의 경우 DeletionPolicy 이 Delete 기본적으로 설정됩니다.

즉, 다음 중 하나에 해당하는 경우 관리형 파일 시스템과 해당 데이터가 삭제됩니다.

- 사용자가 클러스터를 삭제합니다.
- 관리형 공유 스토리지 구성 [SharedStorage/Name](#)을 변경합니다.
- 사용자가 구성에서 관리형 공유 스토리지를 제거합니다.

데이터 손실을 방지하려면 공유 스토리지를 정기적으로 백업하는 것이 좋습니다. 개별 Amazon EFS 볼륨을 백업하는 방법에 대한 자세한 내용은 Amazon Elastic File System 사용 설명서의 [Amazon EFS 파일 시스템 백업](#)을 참조하세요. 전체에서 데이터 백업을 관리하는 방법을 AWS 서비스알아보려면 AWS Backup 개발자 안내서의AWS [Backup](#)을 참조하십시오.

## Amazon FSx for Lustre

클러스터 수명 주기 외의 장기간 영구 스토리지에 기존 외부 FSx for Lustre 파일 시스템을 사용하려면 [FsxLustreSettings/FileSystemId](#)를 지정하세요.

[FsxLustreSettingsFileSystemId](#)/를 지정하지 않으면 기본적으로 클러스터를 생성할 때 FSx for Lustre 파일 시스템이 관리되는 FSx for Lustre 파일 [FsxLustreSettings](#)시스템이 AWS ParallelCluster 생성됩니다. AWS ParallelCluster 또한 클러스터가 삭제되거나 클러스터 구성에서 파일 시스템이 제거될 때 파일 시스템과 데이터를 삭제합니다.

AWS ParallelCluster 관리형 FSx for Lustre 파일 시스템의 경우 클러스터가 삭제되거나 클러스터 [DeletionPolicy](#)구성에서 파일 시스템이 제거될 때 AWS ParallelCluster /를 [FsxLustreSettings](#)사용하여 파일 시스템에 Delete 명령을 Retain 내릴 수 있습니다. 기본적으로 DeletionPolicy는 Delete로 설정됩니다.

### Warning

AWS ParallelCluster 관리형 공유 스토리지의 경우 DeletionPolicy 이 기본적으로 로 설정됩니다. Delete

즉, 다음 중 하나에 해당하는 경우 관리형 파일 시스템과 해당 데이터가 삭제됩니다.

- 사용자가 클러스터를 삭제합니다.
- 관리형 공유 스토리지 구성 [SharedStorage/Name](#)을 변경합니다.
- 사용자가 구성에서 관리형 공유 스토리지를 제거합니다.

데이터 손실을 방지하려면 공유 스토리지를 정기적으로 백업하는 것이 좋습니다. [SharedStorage/FsxLustreSettings/AutomaticBackupRetentionDays](#) 및 [DailyAutomaticBackupStartTime](#)를 사용하여 클러스터에서 백업을 정의할 수 있습니다. 전체에서 데이터 백업을 관리하는 방법을 AWS 서비스알아보려면 AWS Backup 개발자 안내서의 [AWS Backup](#)을 참조하십시오.

FSx for ONTAP, FSx for OpenZFS 및 파일 캐시 공유 스토리지를 구성합니다.

FSx for ONTAP, FSx for OpenZFS 및 파일 캐시의 경우

[FsxOntapSettings/VolumeId](#), [FsxOpenZfsSettingsVolumeId](#) 및 [FileCacheSettings/FileCacheId](#)를 사용하여 클러스터에 대한 외부 기존 볼륨 또는 파일 캐시 탭재를 지정할 수 있습니다.

AWS ParallelCluster 관리형 공유 스토리지는 ONTAP용 FSX, OpenZFS용 FSX 및 파일 캐시에는 지원되지 않습니다.

## 에서 공유 스토리지로 작업하기 AWS ParallelCluster

공유 스토리지 사용 AWS ParallelCluster 및 공유 스토리지에 대해 알아보세요.

주제

- [AWS ParallelCluster 공유 스토리지 고려사항](#)
- [AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 전환합니다.](#)

## AWS ParallelCluster 공유 스토리지 고려사항

AWS ParallelCluster에서 공유된 스토리지로 작업할 때는 다음 사항을 고려하세요.

- [AWS Backup](#) 또는 다른 방법을 사용하여 파일 시스템 데이터를 백업하여 모든 스토리지 시스템의 백업을 관리하세요.
- 공유 스토리지를 추가하려면 구성 파일에 공유 스토리지 섹션을 추가하고 클러스터를 생성하거나 업데이트합니다.
- 공유 스토리지를 제거하려면 구성 파일에서 공유 스토리지 섹션을 제거하고 클러스터를 업데이트합니다.

- 기존 AWS ParallelCluster 관리형 공유 스토리지를 새 관리형 스토리지로 교체하려면 [SharedStorageName](#)/의 값을 변경하고 클러스터를 업데이트하십시오.

#### Warning

기본적으로 새 Name 매개 변수로 클러스터 업데이트를 수행하면 기존 AWS ParallelCluster 관리형 스토리지와 데이터가 삭제됩니다. Name을 변경하고 기존의 관리형 공유 스토리지 데이터를 보존해야 하는 경우 클러스터를 업데이트하기 전에 DeletionPolicy를 Retain로 설정하거나 데이터를 백업해야 합니다.

- AWS ParallelCluster 관리형 스토리지 데이터를 백업하지 않는 경우 Delete, 클러스터가 삭제되거나 클러스터 구성에서 관리형 스토리지가 제거되고 클러스터가 업데이트되면 데이터가 삭제됩니다. DeletionPolicy
- AWS ParallelCluster 관리형 저장소 데이터를 백업하지 않는 경우 DeletionPolicy 클러스터가 Retain 삭제되기 전에 파일 시스템이 분리되어 다른 클러스터에 외부 파일 시스템으로 다시 연결할 수 있습니다. 데이터가 유지됩니다.
- 클러스터 구성에서 AWS ParallelCluster 관리형 스토리지가 제거된 경우 클러스터 데이터를 보존한 상태로 외부 파일 시스템으로 클러스터에 다시 연결할 수 있습니다. DeletionPolicy Retain
- AWS ParallelCluster 버전 3.4.0부터 [SharedStorageEfsSettings/EncryptionInTransit](#) 및 [IamAuthorization](#) 설정을 구성하여 Amazon EFS 파일 시스템 마운트의 보안을 강화할 수 있습니다.
- 외부 파일 시스템을 /home 디렉터리에 마운트하는 경우 헤드 노드의 /home 디렉터리 콘텐츠를 외부 파일 시스템에 AWS ParallelCluster 복사합니다. 외부 스토리지의 기존 파일이나 디렉터리를 덮어 쓰지 않고 /home 디렉터리의 기존 데이터를 복사합니다. 여기에는 기본 사용자를 위한 클러스터의 SSH 키가 포함되어 있습니다. 외부 파일 시스템에 SSH 키가 아직 없는 경우를 대비하기 위해서입니다. 따라서 동일한 외부 파일 시스템을 해당 /home 디렉터리에 마운트하는 다른 모든 클러스터도 클러스터의 기본 사용자에게 대해 동일한 SSH 키를 갖게 됩니다.
- 클러스터의 /home 디렉터리에 동일한 외부 파일 시스템을 마운트하는 다중 클러스터 환경에서, 헤드 노드에 생성된 컴퓨팅 노드에 대한 액세스 권한을 부여하는 SSH 키는 첫 번째 클러스터가 외부 파일 시스템을 /home에 마운트할 때 한 번만 생성됩니다. AWS ParallelCluster 다른 모든 클러스터는 동일한 SSH 키를 사용합니다. 따라서 이러한 공유 클러스터의 기본 사용자에게 대한 SSH 키를 소유한 사람은 누구나 모든 클러스터에 액세스할 수 있습니다. 모든 컴퓨팅 노드는 처음에 생성된 키를 사용하여 연결할 수 있습니다.

## AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 전환합니다.

AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 변환하는 방법을 알아보세요.

절차는 다음 예제 구성 파일 스니펫을 기반으로 합니다.

```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Delete
...
```

AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 전환하세요

1. 클러스터 구성 파일에서 DeletionPolicy를 Retain로 설정합니다.

```
...
- MountDir: /fsx
  Name: fsx
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: 1200
    DeletionPolicy: Retain
...
```

2. 다음 명령을 실행하여 DeletionPolicy 변경을 설정합니다.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

3. 클러스터 구성 파일에서 SharedStorage 섹션을 제거합니다.

```
...
...
```

4. 관리형 SharedStorage를 외부 SharedStorage로 변경하고 클러스터에서 분리하려면 다음 명령을 실행합니다.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. 공유 스토리지는 이제 외부에 있으며 클러스터와 분리되어 있습니다.
6. 외부 파일 시스템을 원래 클러스터나 다른 클러스터에 연결하려면 다음 단계를 따르세요.
  - a. FSx for Lustre 파일 시스템 ID를 가져옵니다.
    - i. 사용하려면 다음 명령을 AWS CLI 실행하여 원래 클러스터의 이름이 포함된 이름의 파일 시스템을 찾은 다음 파일 시스템 ID를 기록해 둡니다.

```
aws fsx describe-file-systems
```

- ii. 를 사용하려면 AWS Management Console로그인하고 <https://console.aws.amazon.com/fsx/>으로 이동하십시오. 파일 시스템 목록에서 원래 클러스터의 이름이 포함된 이름을 가진 파일 시스템을 찾아 파일 시스템 ID를 기록해 둡니다.
- b. 파일 시스템 및 클러스터 서브넷에 대한 액세스를 제공하도록 파일 시스템 보안 그룹 규칙을 업데이트하세요. Amazon FSx 콘솔에서 파일 시스템 보안 그룹 이름과 ID를 찾을 수 있습니다.

헤드 노드와 컴퓨팅 노드 IP CIDR 범위 또는 접두사를 오가는 인바운드 및 아웃바운드 TCP 트래픽을 허용하는 규칙을 파일 시스템 보안 그룹에 추가합니다. 인바운드 및 아웃바운드 TCP 트래픽에 대해 TCP 포트 988, 1021, 1022, 1023을 지정합니다.

자세한 내용은 버전 [2 AWS Command Line Interface 사용 설명서의 SharedStorageFsxLustreSettings//FileSystemId](#) 및 Amazon EC2용 보안 그룹 생성, 구성 및 삭제를 참조하십시오.

- c. SharedStorage 섹션을 클러스터 구성에 추가합니다.

```
...
- MountDir: /fsx
  Name: fsx-external
  StorageType: FsxLustre
  FsxLustreSettings:
    FileSystemId: fs-02e5b4b4abd62d51c
...
```

- d. 외부 공유 스토리지를 클러스터에 추가하려면 다음 명령을 실행합니다.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

## 공유 스토리지 할당량

기존 공유 파일 스토리지를 탑재하도록 클러스터 SharedStorage를 구성하고 다음 표에 나열된 할당량을 기반으로 새 공유 파일 스토리지를 생성합니다.

각 클러스터에 탑재된 파일 스토리지 할당량

| 파일 공유 스토리지 유형 | AWS ParallelCluster 관리형 스토리지 | 외부 스토리지 | 할당량 순 총액 |
|---------------|------------------------------|---------|----------|
| Amazon EBS    | 5                            | 5       | 5        |
| RAID          | 1                            | 0       | 1        |
| Amazon EFS    | 1                            | 20      | 21       |
| Amazon FSx †  | 1 FSx for Lustre             | 20      | 21       |

### Note

이 할당량 표는 AWS ParallelCluster 버전 3.2.0에 추가되었습니다.

† NetApp ONTAP용 기존 Amazon FSX, OpenZFS용 Amazon FSX 및 파일 캐시 시스템의 AWS ParallelCluster 마운트만 지원합니다. 새 FSx for ONTAP, FSx for OpenZFS 및 파일 캐시 시스템의 새 생성은 지원하지 않습니다.

### Note

AWS Batch 스케줄러로 사용하는 경우 FSx for Lustre는 클러스터 헤드 노드에서만 사용할 수 있습니다.

파일 캐시는 스케줄러를 지원하지 않습니다. AWS Batch

## AWS ParallelCluster 리소스 및 태깅

AWS ParallelCluster 를 사용하여 태그를 생성하여 AWS ParallelCluster 리소스를 추적하고 관리할 수 있습니다. AWS CloudFormation 생성하여 모든 클러스터 리소스에 전파하려는 태그를 클러스터 구성



파일에서 정의합니다. [Tags 섹션](#) 또한 AWS ParallelCluster 자동으로 생성되는 태그를 사용하여 리소스를 추적하고 관리할 수 있습니다.

클러스터를 생성하면 클러스터와 해당 리소스에 이 섹션에 정의된 AWS ParallelCluster 및 AWS 시스템 태그가 지정됩니다.

AWS ParallelCluster 클러스터 인스턴스, 볼륨 및 리소스에 태그를 적용합니다. 클러스터 스택을 식별하기 위해 클러스터 인스턴스에 AWS 시스템 태그를 AWS CloudFormation 적용합니다. Amazon EC2는 클러스터 Amazon EC2 시작 템플릿을 식별하기 위해 인스턴스에 시스템 태그를 적용합니다. 이러한 태그를 사용하여 리소스를 보고 관리할 AWS ParallelCluster 수 있습니다.

AWS 시스템 태그는 수정할 수 없습니다. AWS ParallelCluster 기능에 영향을 주지 않으려면 AWS ParallelCluster 태그를 수정하지 마세요.

다음은 AWS ParallelCluster 리소스에 대한 AWS 시스템 태그의 예입니다. 해당 태그는 수정할 수 없습니다.

```
"aws:cloudformation:stack-name"="clustername"
```

다음은 리소스에 적용된 AWS ParallelCluster 태그의 예입니다. 수정하지 마세요.

```
"parallelcluster:cluster-name"="clustername"
```

의 Amazon EC2 섹션에서 이러한 태그를 볼 수 있습니다. AWS Management Console

태그 보기

1. <https://console.aws.amazon.com/ec2/> 에서 Amazon EC2 콘솔을 탐색하십시오.
2. 모든 클러스터 태그를 보려면 탐색 창에서 태그를 선택합니다.
3. 인스턴스별로 클러스터 태그를 보려면 탐색 창에서 인스턴스를 선택합니다.
4. 클러스터 인스턴스를 선택합니다.
5. 인스턴스 세부 정보에서 태그 관리 탭을 선택하고 태그를 확인합니다.
6. 인스턴스 세부 정보에서 스토리지 탭을 선택합니다.
7. 볼륨 ID를 선택합니다.
8. 볼륨에서 볼륨을 선택합니다.
9. 볼륨 세부 정보에서 태그 탭을 선택하고 태그를 확인합니다.

## AWS ParallelCluster 헤드 노드 인스턴스 태그

| 키                                          | 태그 값                                                                                                                     |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>parallelcluster:cluster-name</code>  | <i>clustername</i>                                                                                                       |
| Name                                       | HeadNode                                                                                                                 |
| <code>aws:ec2launchtemplate:id</code>      | <i>lt-1234567890abcdef0</i>                                                                                              |
| <code>aws:ec2launchtemplate:version</code> | <i>1</i>                                                                                                                 |
| <code>parallelcluster:node-type</code>     | HeadNode                                                                                                                 |
| <code>aws:cloudformation:stack-name</code> | <i>clustername</i>                                                                                                       |
| <code>aws:cloudformation:logical-id</code> | HeadNode                                                                                                                 |
| <code>aws:cloudformation:stack-id</code>   | <i>arn:aws:cloudformation: <b>region-id</b> :ACCOUNTID :stack/clustername /1234abcd-12ab-12ab-12ab-1234567890abcdef0</i> |
| <code>parallelcluster:version</code>       | <i>3.7.0</i>                                                                                                             |

## AWS ParallelCluster 헤드 노드 루트 볼륨 태그

| 태그 키                                      | 태그 값               |
|-------------------------------------------|--------------------|
| <code>parallelcluster:cluster-name</code> | <i>clustername</i> |
| <code>parallelcluster:node-type</code>    | HeadNode           |
| <code>parallelcluster:version</code>      | <i>3.7.0</i>       |

## AWS ParallelCluster 컴퓨팅 노드 인스턴스 태그

| 키                                         | 태그 값               |
|-------------------------------------------|--------------------|
| <code>parallelcluster:cluster-name</code> | <i>clustername</i> |

| 키                                     | 태그 값                         |
|---------------------------------------|------------------------------|
| parallelcluster:compute-resource-name | <i>compute-resource-name</i> |
| aws:ec2launchtemplate:id              | <i>lt-1234567890abcdef0</i>  |
| aws:ec2launchtemplate:version         | <i>1</i>                     |
| parallelcluster:node-type             | Compute                      |
| parallelcluster:queue-name            | <i>queue-name</i>            |
| parallelcluster:version               | <i>3.7.0</i>                 |

#### AWS ParallelCluster 컴퓨팅 노드 루트 볼륨 태그

| 태그 키                                  | 태그 값                         |
|---------------------------------------|------------------------------|
| parallelcluster:cluster-name          | <i>clustername</i>           |
| parallelcluster:compute-resource-name | <i>compute-resource-name</i> |
| parallelcluster:node-type             | Compute                      |
| parallelcluster:queue-name            | <i>queue-name</i>            |
| parallelcluster:version               | <i>3.7.0</i>                 |

#### AWS ParallelCluster UI 태그

| 태그 키               | 태그 값 |
|--------------------|------|
| parallelcluster-ui | true |

## 모니터링 AWS ParallelCluster 및 로그

모니터링은 및 기타 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 AWS ParallelCluster 있어 중요한 부분입니다. AWS 문제 발생 시 이를 확인하고 보고하고 적절한 AWS ParallelCluster 경우 자동 조치를 취할 수 있는 다음과 같은 모니터링 도구를 제공합니다.

- Amazon은 실행 중인 AWS 리소스와 애플리케이션을 AWS 실시간으로 CloudWatch 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요할 때 새 인스턴스를 자동으로 시작할 수 있습니다. 자세한 내용은 [Amazon CloudWatch 사용 설명서](#)를 참조하십시오.
- Amazon CloudWatch Logs를 사용하면 Amazon EC2 인스턴스 및 기타 소스에서 로그 파일을 모니터링 CloudTrail, 저장 및 액세스할 수 있습니다. CloudWatch 로그는 로그 파일의 정보를 모니터링하고 특정 임계값이 충족되면 알려줄 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs 사용 설명서](#)를 참조하십시오.
- AWS CloudTrail은 직접 수행하거나 AWS 계정을 대신하여 수행한 API 호출 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 어떤 사용자 및 계정이 AWS를 호출했는지 어떤 소스 IP 주소에 호출이 이루어졌는지 언제 호출이 발생했는지 확인할 수 있습니다. 자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 참조하십시오.
- EventBridgeAmazon은 다양한 소스의 데이터에 애플리케이션을 쉽게 연결할 수 있게 해주는 서버리스 이벤트 버스 서비스입니다. EventBridge 자체 애플리케이션, oftware-as-a S-Service (SaaS) 애플리케이션 AWS 및 서비스에서 실시간 데이터 스트림을 제공하고 해당 데이터를 Lambda와 같은 대상으로 라우팅합니다. 이를 통해 서비스에서 발생하는 이벤트를 모니터링하고 이벤트 기반 아키텍처를 구축할 수 있습니다. 자세한 내용은 [Amazon EventBridge 사용 설명서](#)를 참조하십시오.

### 주제

- [Amazon CloudWatch Logs와 통합](#)
- [아마존 CloudWatch 대시보드](#)
- [클러스터 지표에 대한 Amazon CloudWatch 경보](#)
- [AWS ParallelCluster 로그 교체 구성](#)
- [pcluster CLI 로그](#)
- [Amazon EC2 콘솔 출력 로그](#)
- [AWS ParallelCluster UI 및 AWS ParallelCluster 런타임 로그 검색](#)

- [로그 검색 및 보존](#)

## Amazon CloudWatch Logs와 통합

CloudWatch Logs에 대한 자세한 내용은 [Amazon CloudWatch Logs 사용 설명서](#)를 참조하세요. CloudWatch Logs 통합을 구성하려면 [Monitoring](#) 섹션을 참조하세요. `append-config`를 사용하여 CloudWatch 구성에 사용자 지정 로그를 추가하는 방법을 알아보려면 Amazon CloudWatch 사용 설명서의 [다중 CloudWatch 에이전트 구성 파일](#)을 참조하세요.

### Amazon CloudWatch Logs 클러스터 로그

이름이 `/aws/parallelcluster/cluster-name-<timestamp>`인 각 클러스터에 대해 로그 그룹이 생성됩니다(예: `/aws/parallelcluster/testCluster-202202050215`). 각 노드의 각 로그(또는 경로에 \*가 포함된 경우 로그 집합)에는 `{hostname}.{instance_id}.{logIdentifier}`라는 로그 스트림이 있습니다. (예: `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`.) 로그 데이터는 모든 클러스터 인스턴스에서 root로 실행되는 [CloudWatch 에이전트](#)에 의해 CloudWatch로 전송됩니다.

Amazon CloudWatch 대시보드는 클러스터가 생성될 때 생성됩니다. 이 대시보드를 사용하면 CloudWatch Logs에 저장된 로그를 검토할 수 있습니다. 자세한 내용은 [아마존 CloudWatch 대시보드](#) 항목을 참조하세요.

이 목록에는 플랫폼, 스케줄러 및 노드에 사용할 수 있는 로그 스트림의 `logIdentifier` 및 경로가 포함되어 있습니다.

플랫폼, 스케줄러 및 노드에 사용할 수 있는 로그 스트림

| 플랫폼    | 스케줄러    | 노드     | 로그 스트림                                                                      |
|--------|---------|--------|-----------------------------------------------------------------------------|
| amazon | awsbatc | HeadNc | dcv-authenticator: /var/log/parallelcluster/pc_luster_dcv_authenticator.log |
| centos | slurm   |        | dcv-ext-authenticator: /var/log/parallelcluster/pc_luster_dcv_connect.log   |
| redhat |         |        | dcv-agent: /var/log/dcv/agent.*.log                                         |
| ubuntu |         |        | dcv-xsession: /var/log/dcv/dcv-xsession.*.log                               |

| 플랫폼                                  | 스케줄러             | 노드                      | 로그 스트림                                                                                                                                                                                                         |
|--------------------------------------|------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                      |                  |                         | dcv-server: /var/log/dcv/server.log<br>dcv-session-launcher: /var/log/dcv/sessionlauncher.log<br>Xdcv: /var/log/dcv/Xdcv.*.log<br>cfn-init: /var/log/cfn-init.log<br>chef-client: /var/log/chef-client.log     |
| amazon<br>centos<br>redhat<br>ubuntu | awsbatc<br>slurm | Comput<br>eet<br>HeadNc | cloud-init: /var/log/cloud-init.log<br>supervisord: /var/log/supervisord.log                                                                                                                                   |
| amazon<br>centos<br>redhat<br>ubuntu | slurm            | Comput<br>eet           | cloud-init-output: /var/log/cloud-init-output.log<br>computemgtd: /var/log/parallelcluster/computemgtd<br>slurmd: /var/log/slurmd.log<br>slurm_prolog_epilog: /var/log/parallelcluster/slurm_prolog_epilog.log |

| 플랫폼    | 스케줄러     | 노드          | 로그 스트림                                                                              |
|--------|----------|-------------|-------------------------------------------------------------------------------------|
| amazon | slurm    | HeadNode    | sssd: /var/log/sssds/sssds.log                                                      |
| centos |          |             | sssd_domain_default: /var/log/sssds/sssds_default.log                               |
| redhat |          |             | pam_ssh_key_generator: /var/log/parallelcluster/pam_ssh_key_generator.log           |
| ubuntu |          |             | clusterstatusmgtd: /var/log/parallelcluster/clusterstatusmgtd                       |
|        |          |             | clustermgtd: /var/log/parallelcluster/clustermgtd                                   |
|        |          |             | compute_console_output: /var/log/parallelcluster/compute_console_output             |
|        |          |             | slurm_resume: /var/log/parallelcluster/slurm_resume.log                             |
|        |          |             | slurm_suspend: /var/log/parallelcluster/slurm_suspend.log                           |
|        |          |             | slurmctld: /var/log/slurmctld.log                                                   |
|        |          |             | slurm_fleet_status_manager: /var/log/parallelcluster/slurm_fleet_status_manager.log |
| amazon | awsbatch | ComputeNode | system-messages: /var/log/messages                                                  |
| centos | slurm    | HeadNode    |                                                                                     |
| redhat |          |             |                                                                                     |
| ubuntu | awsbatch | ComputeNode | syslog: /var/log/syslog                                                             |
|        | slurm    | HeadNode    |                                                                                     |

AWS Batch를 사용하는 클러스터의 작업은 RUNNING, SUCCEEDED, 또는 FAILED 상태에 도달한 작업의 출력을 CloudWatch Logs에 저장합니다. 로그 그룹은 `/aws/batch/job`이며 로그 스트림 이름 형식은 `jobDefinitionName/default/ecs_task_id`입니다. 기본적으로 이러한 로그들은 만료되도록 설정하지 않지만 유지 기간을 수정할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs User Guide의 [CloudWatch에서 로그 데이터 보존 기간을 변경](#)을 참조하세요.

## Amazon CloudWatch Logs 빌드 이미지 로그

각 사용자 지정 빌드 이미지에 대해 이름이 `/aws/imagebuilder/ParallelClusterImage-<image-id>`인 로그 그룹이 생성됩니다. 이름이 `{pcluster-version}/1`인 고유한 로그 스트림에는 빌드 이미지 프로세스의 출력이 포함됩니다.

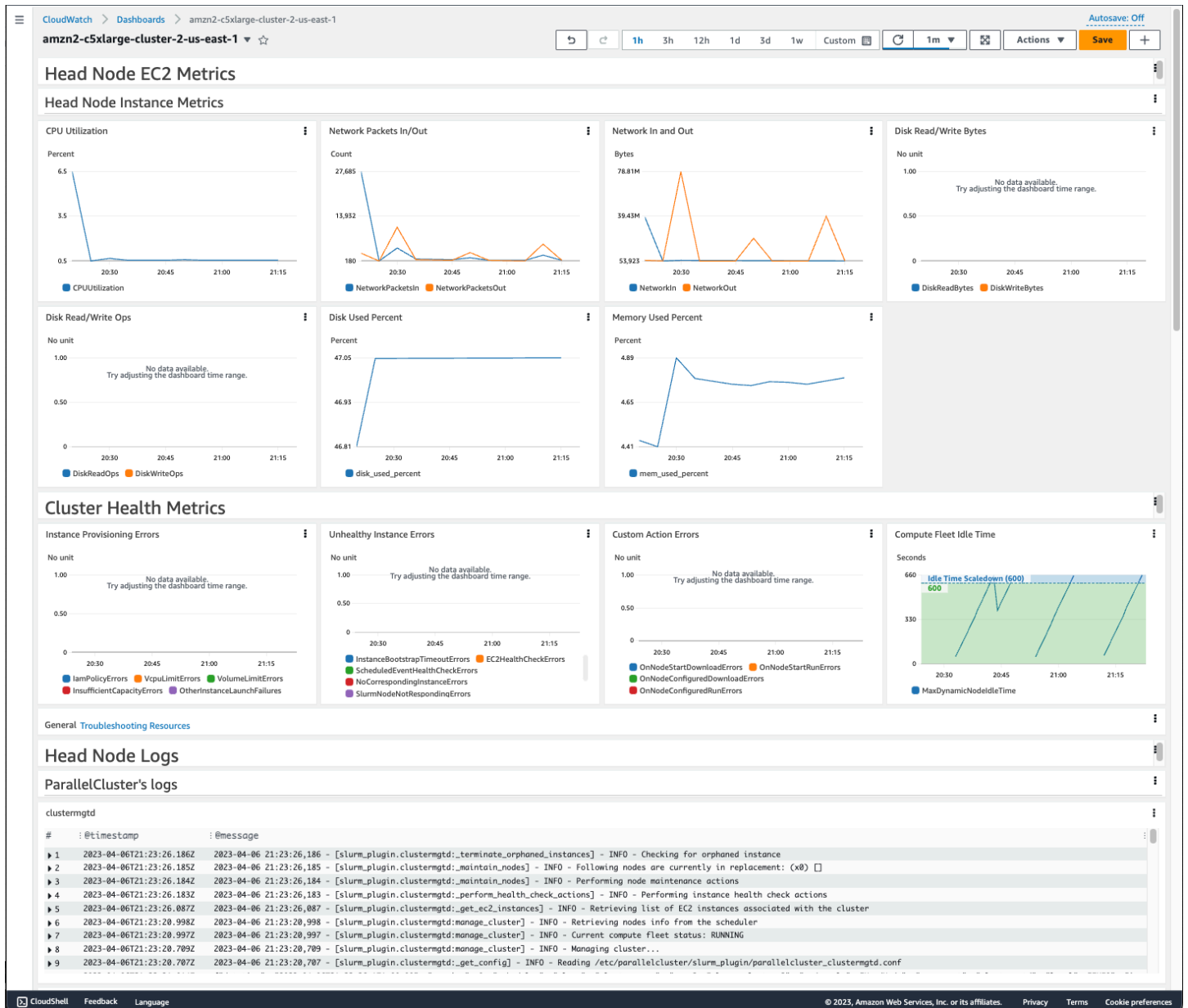
`pcluster` 이미지 명령을 사용하여 로그에 액세스할 수 있습니다. 자세한 내용은 [AWS ParallelCluster AMI 사용자 지정](#) 항목을 참조하세요.

## 아마존 CloudWatch 대시보드

Amazon CloudWatch 대시보드는 클러스터가 생성될 때 생성됩니다. 이렇게 하면 클러스터의 노드를 더 쉽게 모니터링하고 Amazon CloudWatch Logs에 저장된 로그를 볼 수 있습니다. 대시보드의 이름은 `ClusterName-Region`입니다. `ClusterName`는 클러스터의 이름이고 클러스터가 속한 `###` AWS 리전 클러스터의 이름입니다. 콘솔에서 또는 `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`를 열어 대시보드에 액세스할 수 있습니다.

다음 이미지는 클러스터의 예제 CloudWatch 대시보드를 보여줍니다.





## 헤드 노드 인스턴스 지표

대시보드의 첫 번째 섹션에는 헤드 노드 Amazon EC2 지표의 그래프가 표시됩니다.

클러스터에 공유 스토리지가 있는 경우 다음 섹션에는 공유 스토리지 지표가 표시됩니다.

## 클러스터 상태 지표

클러스터가 스케줄링에 Slurm을 사용하는 경우 클러스터 상태 지표 그래프에 실시간 클러스터 컴퓨팅 노드 오류가 표시됩니다. 자세한 정보는 [클러스터 상태 지표 문제 해결](#)을 참조하세요. 클러스터 상태 지표는 AWS ParallelCluster 버전 3.6.0부터 대시보드에 추가됩니다.

## 헤드 노드 로그

마지막 섹션에는 AWS ParallelCluster의 로그, 스케줄러 로그, NICE DCV 통합 로그 및 시스템 로그별로 그룹화된 헤드 노드 로그가 나열됩니다.

Amazon CloudWatch 대시보드에 대한 자세한 내용은 Amazon 사용 CloudWatch 설명서의 Amazon CloudWatch [대시보드 사용](#)을 참조하십시오.

Amazon CloudWatch 대시보드를 생성하지 않으려는 경우 [MonitoringDashboardsCloudWatchEnabled](#)를 설정하여 비활성화할 수 false 있습니다.

### Note

Amazon CloudWatch 대시보드 생성을 비활성화하면 클러스터의 Amazon CloudWatch `disk_used_percent` 및 `memory_used_percent` 경보도 비활성화됩니다. 자세한 정보는 [클러스터 지표에 대한 Amazon CloudWatch 경보](#)을 참조하세요.

`disk_used_percent` 및 `memory_used_percent` 경보는 AWS ParallelCluster 버전 3.6부터 추가되었습니다.

## 클러스터 지표에 대한 Amazon CloudWatch 경보

AWS ParallelCluster 버전 3.6부터 헤드 노드를 모니터링하기 위한 Amazon CloudWatch 경보를 사용하여 클러스터를 구성할 수 있습니다. 하나의 경보가 루트 볼륨 `disk_used_percent`을 모니터링합니다. 다른 경보는 `mem_used_percent` 지표를 모니터링합니다. 자세한 내용은 [Amazon CloudWatch 사용 설명서](#)의 CloudWatch 에이전트가 수집하는 지표를 참조하세요.

경보 이름은 다음과 같습니다.

- `cluster-name_DiskAlarm_HeadNode`
- `cluster-name_MemAlarm_HeadNode`

`cluster-name`은 클러스터의 이름입니다.

탐색 창에서 경보를 선택하여 CloudWatch 콘솔에서 경보에 액세스합니다. 다음 이미지는 클러스터의 디스크 사용량 경보 및 메모리 사용 경보를 보여줍니다.

CloudWatch > Alarms > test-disk-alarm-1\_DiskAlarm\_HeadNode

**Alarms (11)**

Any state ▼

Any type ▼

Any actions status ▼

Hide Auto Scaling alarms

< 1 >

test-disk-alarm-1\_DiskAlarm\_HeadNode  
Metric alarm  
 OK

test-disk-alarm-1\_MemAlarm\_HeadNode  
Metric alarm  
 OK

mytest  
Metric alarm  
 Insufficient data

test-disk-alarm-1\_DiskAlarm\_HeadNode  **Analyse** ▼ **Actions** ▼

### Graph

**disk\_used\_percent**  OK

disk\_used\_percent > 90 for 1 datapoints within 1 minute

Percent

90.00

67.67

45.35

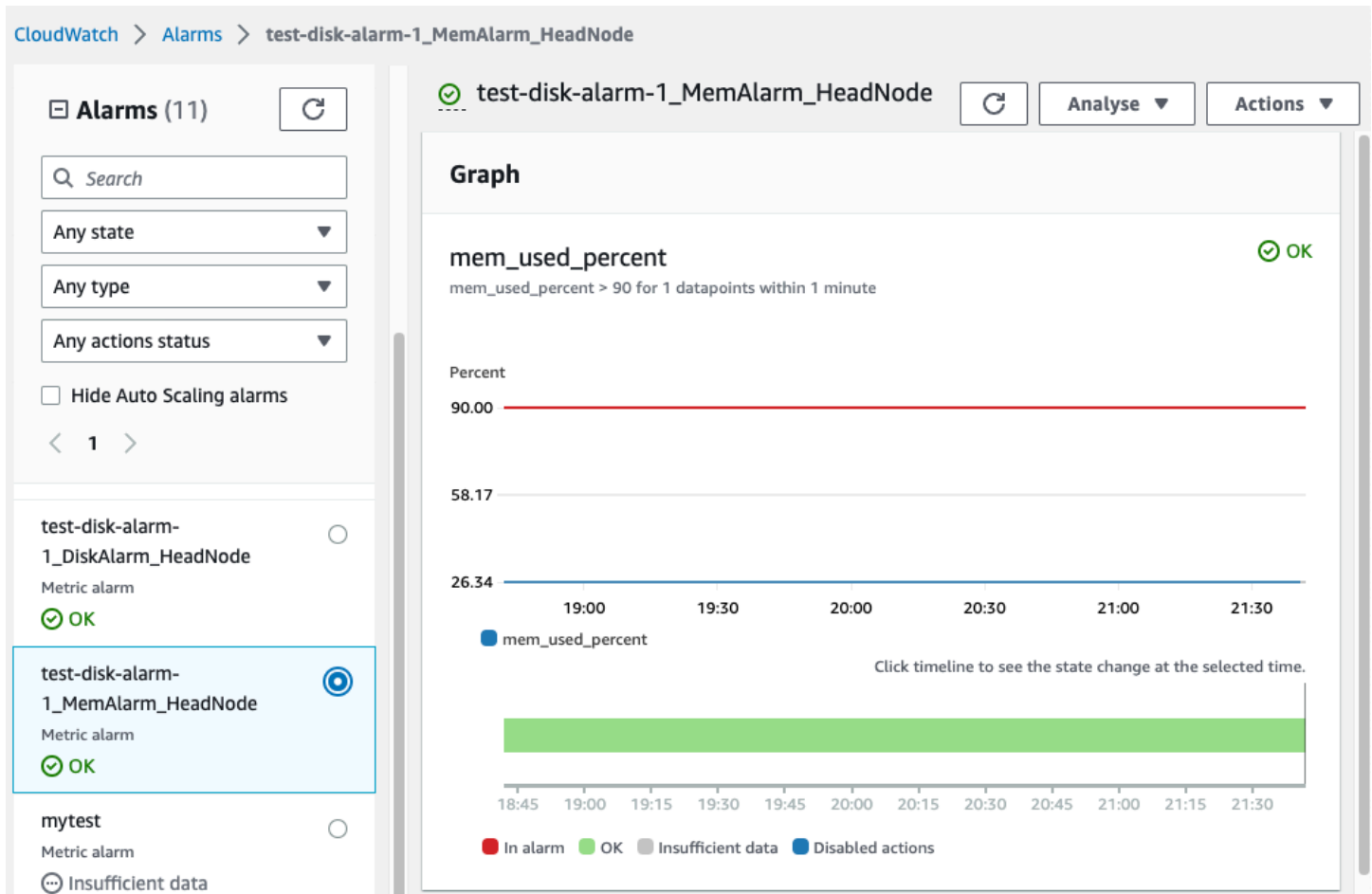
19:00 19:30 20:00 20:30 21:00 21:30

disk\_used\_percent

Click timeline to see the state change at the selected time.

18:45 19:00 19:15 19:30 19:45 20:00 20:15 20:30 20:45 21:00 21:15 21:30

In alarm  OK  Insufficient data  Disabled actions



디스크 사용량 경보는 1분 동안 데이터 포인트 1개의 디스크 사용률이 90%를 초과하는 ALARM 상태입니다.

메모리 사용량 경보는 1분 동안 데이터 포인트 1개의 메모리 사용률이 90%를 초과하는 ALARM 상태입니다.

### Note

기본적으로 AWS ParallelCluster는 경보 작업을 구성하지 않습니다. 경보 전송과 같은 경보 작업을 설정하는 방법에 대한 자세한 내용은 [경보 작업을 참조](#)하세요. Amazon CloudWatch 경보에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 [Amazon CloudWatch 경보 사용](#)을 참조하세요.

이러한 Amazon CloudWatch 경보를 생성하지 않으려면 클러스터 구성에서 [Monitoring/Dashboards/CloudWatch/Enabled](#)를 false로 설정하여 경보를 비활성화하세요. 이

렇게 하면 Amazon CloudWatch 대시보드 생성도 비활성화됩니다. 자세한 내용은 [아마존 CloudWatch 대시보드](#) 항목을 참조하세요.

### Note

Amazon CloudWatch 대시보드 생성을 비활성화하면 클러스터에 대한 Amazon CloudWatch `disk_used_percent` 및 `memory_used_percent` 경보도 비활성화됩니다.

## AWS ParallelCluster 로그 교체 구성

AWS ParallelCluster 로그 교체 구성은 `/etc/logrotate.d/parallelcluster_*_log_rotation` 파일에 있습니다. 구성된 로그가 교체되면 현재 로그 내용은 단일 백업에 보존되고 비워진 로그는 로깅을 재개합니다.

구성된 각 로그에 대해 백업은 1개만 유지됩니다.

AWS ParallelCluster는 빠르게 증가하는 로그의 크기가 50MB에 도달하면 교체하도록 구성합니다. 빠르게 증가하는 로그는 `/var/log/parallelcluster/clustermgtd`, `/var/log/parallelcluster/slurm_resume.log`, `/var/log/slurmctld.log`를 포함하여 크기 조정 및 Slurm과 관련이 있습니다.

AWS ParallelCluster는 느리게 증가하는 로그의 크기가 10MB에 도달하면 교체하도록 구성합니다.

CloudFormation 로깅이 활성화된 상태에서 클러스터 구성 [Logs/CloudWatch/RetentionInDays](#) 설정에 정의된 일수 동안 보존된 이전 로그를 볼 수 있습니다. `RetentionInDays` 설정에서 사용 사례에 따라 일수를 늘려야 하는지 확인하세요.

AWS ParallelCluster는 다음 로그를 구성하고 교체합니다.

### 헤드 노드 로그

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cfn-init.log
/var/log/chef-client.log
/var/log/dcv/server.log
/var/log/dcv/sessionlauncher.log
/var/log/dcv/agent.*.log
/var/log/dcv/dcv-xsession.*.log
/var/log/dcv/Xdcv.*.log
/var/log/parallelcluster/pam_ssh_key_generator.log
```

```

/var/log/parallelcluster/clustermgtd
/var/log/parallelcluster/clusterstatusmgtd
/var/log/parallelcluster/slurm_fleet_status_manager.log
/var/log/parallelcluster/slurm_resume.log
/var/log/parallelcluster/slurm_suspend.log
/var/log/slurmctld.log
/var/log/slurmdbd.log
/var/log/parallelcluster/compute_console_output.log

```

## 컴퓨팅 노드 로그

```

/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log

```

## 로그인 노드 로그

```

/var/log/cloud-init.log
/var/log/cloud-init.log
/var/log/cloud-init-output.log
/var/log/supervisord.log
/var/log/parallelcluster/pam_ssh_key_generator.log

```

## pcluster CLI 로그

pcluster CLI는 명령 로그를 `/home/user/.parallelcluster/` 내의 `pcluster.log.#` 파일에 기록합니다.

각 명령에 대한 로그에는 일반적으로 입력이 포함된 명령, 명령을 만드는 데 사용된 CLI API 버전 사본, 응답, 정보 및 오류 메시지가 포함됩니다. `create` and `build` 명령의 경우 로그에는 구성 파일, 구성 파일 검증 작업, CloudFormation 템플릿 및 스택 명령도 포함됩니다.

이러한 로그를 사용하여 오류, 입력, 버전 및 pcluster CLI 명령을 확인할 수 있습니다. 또한 명령이 언제 실행되었는지에 대한 기록으로도 사용할 수 있습니다.

## Amazon EC2 콘솔 출력 로그

정적 컴퓨팅 노드 인스턴스가 예기치 않게 종료되는 것을 AWS ParallelCluster 감지하면 일정 시간이 경과한 후 종료된 노드 인스턴스에서 Amazon EC2 콘솔 출력을 검색하려고 시도합니다. 이렇게 하면

컴퓨팅 노드가 CloudWatch Amazon과 통신할 수 없는 경우에도 노드가 종료된 이유에 대한 유용한 문제 해결 정보가 콘솔 출력에서 계속 검색될 수 있습니다. 이 콘솔 출력은 헤드 노드의 `/var/log/parallelcluster/compute_console_output` 로그에 기록됩니다. Amazon EC2 콘솔 출력에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 콘솔 출력을](#) 참조하십시오.

기본적으로 종료된 노드의 샘플 하위 AWS ParallelCluster 집합에서만 콘솔 출력을 검색합니다. 이렇게 하면 잦은 종료로 다수의 콘솔 출력 요청이 발생해 클러스터 헤드 노드가 과부하되는 것을 방지할 수 있습니다. 기본적으로 Amazon EC2가 노드에서 최종 콘솔 출력을 검색할 시간을 주기 위해 종료 감지와 콘솔 출력 검색 사이에 5분을 AWS ParallelCluster 기다립니다.

헤드 노드의 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` 파일에서 샘플 크기 및 대기 시간 파라미터 값을 편집할 수 있습니다.

이 기능은 버전 3.5.0에 추가되었습니다. AWS ParallelCluster

## Amazon EC2 콘솔 출력 파라미터

헤드 노드의 `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` 파일에서 다음 Amazon EC2 콘솔 출력 파라미터의 값을 편집할 수 있습니다.

### **compute\_console\_logging\_enabled**

콘솔 출력 로그 수집을 비활성화하려면 `compute_console_logging_enabled`를 `false`로 설정합니다. 기본값은 `true`입니다.

컴퓨팅 플릿을 중지하지 않고 언제든지 이 파라미터를 업데이트할 수 있습니다.

### **compute\_console\_logging\_max\_sample\_size**

`compute_console_logging_max_sample_size` 예상치 못한 종료를 감지할 때마다 콘솔 출력을 AWS ParallelCluster 수집하는 컴퓨팅 노드의 최대 수를 설정합니다. 이 값이 보다 1 작으면 종료된 모든 노드에서 콘솔 출력을 AWS ParallelCluster 검색합니다. 기본값은 1입니다.

컴퓨팅 플릿을 중지하지 않고 언제든지 이 파라미터를 업데이트할 수 있습니다.

### **compute\_console\_wait\_time**

`compute_console_wait_time` 노드 장애를 감지한 후 AWS ParallelCluster 해당 노드에서 콘솔 출력을 수집하는 데 걸리는 시간 (초) 을 설정합니다. Amazon EC2가 종료된 노드에서 최종 출력을 수집하는 데 시간이 더 필요하다고 판단되면 대기 시간을 늘릴 수 있습니다. 기본값은 300초(5분)입니다.

컴퓨팅 플릿을 중지하지 않고 언제든지 이 파라미터를 업데이트할 수 있습니다.

## AWS ParallelCluster UI 및 AWS ParallelCluster 런타임 로그 검색

문제 해결을 위해 AWS ParallelCluster UI 및 AWS ParallelCluster 런타임 로그를 검색하는 방법을 알아봅니다. 시작하려면 관련 AWS ParallelCluster UI와 AWS ParallelCluster 스택 이름을 찾아보세요. 스택 이름을 사용하여 설치 로그 그룹을 찾을 수 있습니다. 완료하려면 로그를 내보내세요. 이러한 로그는 AWS ParallelCluster 런타임에만 해당됩니다. 클러스터 로그는 [로그 검색 및 보존](#) 섹션을 참조하세요.

### 사전 조건

- AWS CLI가 설치되어 있습니다.
- AWS ParallelCluster UI가 켜져 있는 AWS 계정에 AWS CLI 명령을 실행할 보안 인증이 있습니다.
- AWS ParallelCluster UI가 켜져 있는 AWS 계정으로 Amazon CloudWatch 콘솔에 액세스할 수 있습니다.

### 1단계: 관련 스택의 스택 이름 찾기

다음 예에서는 빨간색으로 강조 표시된 텍스트를 실제 값으로 바꿉니다.

AWS ParallelCluster UI를 설치한 AWS 리전 위치를 사용하여 스택을 나열하세요.

```
$ aws cloudformation list-stacks --region aws-region-id
```

다음 스택의 스택 이름을 적어 둡니다.

- 계정에 AWS ParallelCluster UI를 배포한 스택의 이름입니다. AWS ParallelCluster UI를 설치할 때 이 이름을 입력했습니다 (예: pcluster-ui)
- 입력한 스택 이름 앞에 접두사가 붙은 AWS ParallelCluster 스택입니다. (예: pcluster-ui-ParallelClusterApi-ABCD1234EFGH)

### 2단계: 로그 그룹 찾기

다음 예와 같이 AWS ParallelCluster UI 스택의 로그 그룹을 나열합니다.

```
$ aws cloudformation describe-stack-resources \
  --region aws-region-id \
  --stack-name pcluster-ui \
```



```
--query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' &&
(LogicalResourceId == 'ApiGatewayAccessLog' || LogicalResourceId ==
'ParallelClusterUILambdaLogGroup')].PhysicalResourceId" \
--output text
```

다음 예와 같이 AWS ParallelCluster API 스택의 로그 그룹을 나열합니다.

```
$ aws cloudformation describe-stack-resources \
  --region aws-region-id \
  --stack-name pcluster-ui-ParallelCluster-Api-ABCD1234EFGH \
  --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' && LogicalResourceId
== 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \
  --output text
```

다음 단계에서 사용할 수 있도록 로그 그룹 목록을 적어 둡니다.

### 3단계: 로그 내보내기

다음 단계를 사용하여 로그를 수집하고 내보낼 수 있습니다.

1. AWS Management Console에 로그인한 다음 AWS ParallelCluster UI가 켜져 있는 AWS 계정에서 [Amazon CloudWatch](#) 콘솔로 이동합니다.
2. 탐색 창에서 Logs(로그), Logs Insights를 선택합니다.
3. 이전 단계에서 나열된 로그 그룹을 모두 선택합니다.
4. 시간 범위(예: 12시간)를 선택합니다.
5. 다음 쿼리를 실행합니다.

```
$ fields @timestamp, @message
| sort @timestamp desc
| limit 10000
```

6. 결과 내보내기, 테이블 다운로드(JSON)를 선택합니다.

## 로그 검색 및 보존

AWS ParallelCluster 컴퓨팅 인스턴스 HeadNode 및 스토리지에 대한 Amazon EC2 지표를 생성합니다. CloudWatch 콘솔 사용자 지정 대시보드에서 지표를 볼 수 있습니다. AWS ParallelCluster 또한 로그 그룹에 클러스터 CloudWatch 로그 스트림을 생성합니다. CloudWatch 콘솔 사용자 지정 대시보드 또는 로그 그룹에서 이러한 로그를 볼 수 있습니다. [모니터링](#) 클러스터 구성 섹션에서는 클러스터

CloudWatch 로그 및 대시보드를 수정하는 방법을 설명합니다. 자세한 내용은 [Amazon CloudWatch Logs와 통합](#) 및 [아마존 CloudWatch 대시보드](#) 섹션을 참조하세요.

로그는 문제를 해결하는 데 유용한 리소스입니다. 예를 들어, 장애가 발생한 클러스터를 삭제하려면 먼저 클러스터 로그의 아카이브를 만드는 것이 유용할 수 있습니다. 아카이브를 생성하려면 [아카이브 로그](#)에서 다음 단계를 따르세요.

주제

- [에서는 클러스터 로그를 사용할 수 없습니다. CloudWatch](#)
- [아카이브 로그](#)
- [보존된 로그](#)
- [종료된 노드 로그](#)

에서는 클러스터 로그를 사용할 수 없습니다. CloudWatch

에서 CloudWatch 클러스터 로그를 사용할 수 없는 경우 구성에 사용자 지정 로그를 추가할 때 AWS ParallelCluster CloudWatch 로그 구성을 덮어쓰지 않았는지 확인하세요.

구성에 사용자 지정 로그를 추가하려면 CloudWatch 구성을 가져와서 덮어쓰지 말고 구성에 추가해야 합니다. `fetch-config` 및 `에` 대한 자세한 내용은 `사용 append-config` 설명서의 [다중 CloudWatch 에이전트 구성 파일](#)을 참조하십시오. CloudWatch

AWS ParallelCluster CloudWatch 로그 구성을 복원하려면 AWS ParallelCluster 노드 내에서 다음 명령을 실행할 수 있습니다.

```
$ PLATFORM="$(ohai platform | jq -r ".[]")"
LOG_GROUP_NAME="$(cat /etc/chef/dna.json | jq -r ".cluster.log_group_name")"
SCHEDULER="$(cat /etc/chef/dna.json | jq -r ".cluster.scheduler")"
NODE_ROLE="$(cat /etc/chef/dna.json | jq -r ".cluster.node_type")"
CONFIG_DATA_PATH="/usr/local/etc/cloudwatch_agent_config.json"
/opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/python /usr/local/bin/write_cloudwatch_agent_json.py --platform $PLATFORM --config $CONFIG_DATA_PATH --log-group $LOG_GROUP_NAME --scheduler $SCHEDULER --node-role $NODE_ROLE
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json -s
```

## 아카이브 로그

로그는 Amazon S3 또는 로컬 파일(--output-file 파라미터에 따라 다름)에 보관할 수 있습니다.

**Note**

Amazon S3 버킷 정책에 권한을 추가하여 CloudWatch 액세스 권한을 부여합니다. 자세한 내용은 CloudWatch Logs 사용 설명서의 [Amazon S3 버킷에 대한 권한 설정](#)을 참조하십시오.

```
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
  --bucket bucketname --bucket-prefix logs
{
  "url": "https://bucketname.s3.eu-west-1.amazonaws.com/export-log/mycluster-
logs-202109071136.tar.gz?..."
}

# use the --output-file parameter to save the logs locally
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
  --bucket bucketname --bucket-prefix logs --output-file /tmp/archive.tar.gz
{
  "path": "/tmp/archive.tar.gz"
}
```

아카이브에는 구성 또는 명령 파라미터에 명시적으로 지정되지 않은 한 지난 14일 동안 헤드 노드와 컴퓨팅 노드에서 발생한 Amazon CloudWatch Logs 스트림과 AWS CloudFormation 스택 이벤트가 포함됩니다. `export-cluster-logs` 명령이 완료되는 데 걸리는 시간은 클러스터의 노드 수와 Logs에서 CloudWatch 사용 가능한 로그 스트림 수에 따라 달라집니다. 사용 가능한 로그 스트림에 대한 자세한 내용은 [Amazon CloudWatch Logs와 통합](#) 섹션을 참조하세요.

## 보존된 로그

버전 3.0.0부터 클러스터가 삭제될 때 기본적으로 CloudWatch 로그를 AWS ParallelCluster 보존합니다. 클러스터를 삭제하고 해당 로그를 보존하려면 클러스터 구성에서 [Monitoring/Logs/CloudWatch/DeletionPolicy](#)가 Delete로 설정되어 있지 않은지 확인하세요. 그렇지 않으면 이 필드의 값을 Retain으로 변경하고 `pcluster update-cluster` 명령을 실행하세요. 그런 다음 `pcluster delete-cluster --cluster-name <cluster_name>` 를 실행하여 클러스터를 삭제하지만 Amazon에 저장된 로그 그룹은 그대로 유지합니다 CloudWatch.

## 종료된 노드 로그

정적 컴퓨팅 노드가 예기치 않게 종료되고 이에 대한 CloudWatch 로그가 없는 경우 해당 컴퓨팅 노드에 대한 콘솔 출력이 로그에 헤드 노드에 AWS ParallelCluster 기록되었는지 확인하십시오. `/var/`

log/parallelcluster/compute\_console\_output 자세한 정보는 [디버깅을 위한 키 로그](#)을 참조하세요.

/var/log/parallelcluster/compute\_console\_output 로그를 사용할 수 없거나 노드에 대한 출력이 포함되어 있지 않은 경우 를 사용하여 장애가 발생한 AWS CLI 노드에서 콘솔 출력을 검색하십시오. 클러스터 헤드 노드에 로그인하고 /var/log/parallelcluster/slurm\_resume.log 파일에서 장애가 발생한 노드 instance-id를 가져옵니다.

instance-id로 다음 명령을 사용하여 콘솔 출력을 검색합니다.

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

동적 컴퓨팅 노드가 시작 후 자체 종료되고 해당 노드에 대한 CloudWatch 로그가 없는 경우 클러스터 조정 작업을 활성화하는 작업을 제출하십시오. 인스턴스에 장애가 발생할 때까지 기다린 다음 인스턴스 콘솔 로그를 검색하세요.

클러스터 헤드 노드에 로그인하고 /var/log/parallelcluster/slurm\_resume.log 파일에서 컴퓨팅 노드 instance-id를 가져옵니다.

다음 명령을 사용하여 인스턴스 콘솔 로그를 검색합니다.

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

콘솔 출력 로그는 컴퓨팅 노드 로그를 사용할 수 없을 때 컴퓨팅 노드 장애의 근본 원인을 디버깅하는데 도움이 될 수 있습니다.

## AWS CloudFormation 사용자 지정 리소스

AWS ParallelCluster 버전 3.6.0부터 스택에서 AWS ParallelCluster CloudFormation 사용자 지정 리소스를 사용할 수 있습니다. AWS CloudFormation 사용자 지정 리소스는 AWS ParallelCluster 호스팅된 스택입니다. 이렇게 하면 클러스터를 구성하고 관리하는 CloudFormation 데 사용할 수 있습니다. 예를 들어 네트워크, 공유 스토리지, 보안 그룹 인프라와 같은 클러스터 외부 리소스를 CloudFormation 스택에 구성할 수 있습니다. 또한 코드 파이프라인으로서의 CloudFormation 인프라를 사용하여 클러스터를 관리할 수 있습니다.

다음을 수행하여 CloudFormation 템플릿에 AWS ParallelCluster 사용자 지정 리소스를 추가합니다.

1. 에서 소유하고 호스팅하는 사용자 지정 리소스 공급자 스택을 추가합니다 AWS ParallelCluster.

## 2. CloudFormation 템플릿의 제공자 스택을 사용자 지정 리소스로 참조하세요.

사용자 지정 리소스 제공자 스택은 CloudFormation 요청을 처리하고 요청에 응답합니다. 예를 들어 CloudFormation 스택을 배포할 때 클러스터도 구성하고 생성합니다. 클러스터를 업데이트하려면 CloudFormation 스택을 업데이트해야 합니다. 스택을 삭제하면 클러스터도 삭제됩니다. CloudFormation 사용자 지정 리소스에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 사용자 [지정 리소스](#)를 참조하십시오.

### Warning

CloudFormation 사용자 지정 리소스 드리프트를 감지하지 못합니다. 클러스터 구성을 CloudFormation 업데이트하고 클러스터를 삭제하는 데만 사용합니다.

[pcluster](#) CLI 또는 [AWS ParallelCluster UI](#)를 사용하여 클러스터의 상태를 모니터링하거나 컴퓨팅 플릿을 업데이트할 수 있지만, 클러스터 구성을 업데이트하거나 클러스터를 삭제하는 데 사용해서는 안 됩니다.

### Note

실수로 제거되지 않도록 스택에 [종료 보호](#)를 추가하는 것이 좋습니다.

## 에서 호스팅하는 제공자 스택 AWS ParallelCluster

사용자 지정 리소스 제공자 스택은 다음 CloudFormation 템플릿 스니펫에 표시된 대로 형식이 지정되어 있습니다.

```
PclusterClusterProvider:
  Type: AWS::CloudFormation::Stack
  Properties:
    Parameters:
      CustomLambdaRole: # (Optional) RoleARN to override default
      AdditionalIamPolicies: # (Optional) comma-separated list of IAM policies to add
    TemplateURL: !Sub
      - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.${AWS::URLSuffix}/
parallelcluster/${Version}/templates/custom_resource/cluster.yaml
      - { Version: 3.7.0 }
```

**속성:****파라미터:****CustomLambdaRole (선택 사항):**

클러스터를 생성하고 관리하는 AWS Lambda 의 실행 권한이 있는 사용자 지정 역할입니다. 기본적으로 역할은 [AWS ParallelCluster 설명서](#)에 기본적으로 정의된 것과 동일한 정책을 사용합니다.

**AdditionalIamPolicies (선택 사항):**

Lambda가 사용하는 역할에 추가할 추가 IAM 정책 Amazon 리소스 이름(ARN)을 쉼표로 구분한 목록입니다. 이는 CustomLambdaRole이 지정되지 않은 경우에만 사용되며 비워 둘 수 있습니다.

헤드 노드, 컴퓨팅 노드 또는 Amazon S3 버킷 액세스에 대한 추가 정책이 필요한 경우 CustomLambdaRole 또는 AdditionalIamPolicy 속성에 정책을 추가하세요.

기본 정책에 대한 자세한 내용은 [AWS Identity and Access Management 권한 입력 AWS ParallelCluster](#) 섹션을 참조하세요.

**TemplateURL(필수):**

AWS ParallelCluster 사용자 지정 리소스 파일 URL.

**출력:****ServiceToken:**

사용자 지정 리소스 ServiceToken 속성으로 사용할 수 있는 값입니다. 사용자 지정 리소스는 요청을 AWS CloudFormation 보내는 위치를 ServiceToken 지정합니다. AWS CloudFormation 템플릿에 포함하는 클러스터 리소스의 필수 입력입니다.

**LogGroupArn:**

기본 리소스가 CloudWatch LogGroup 기록하는 ARN입니다.

**LambdaLayerArn:**

작업 실행에 사용되는 Lambda 계층의 ARN입니다. AWS ParallelCluster

## 클러스터 리소스

CloudFormation 클러스터 리소스는 다음 템플릿 스니펫에 표시된 대로 형식이 지정됩니다.

### CloudFormation

```
PclusterCluster:
  Type: Custom::PclusterCluster
  Properties:
    ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
    ClusterName: !Sub 'c-${AWS::StackName}' # Must be different from StackName
    ClusterConfiguration:
      # Your Cluster Configuration
```

### 속성:

#### ServiceToken:

AWS ParallelCluster 제공자 스택 출력. ServiceToken

#### ClusterName:

생성 및 관리할 새 클러스터의 이름입니다. 이름은 CloudFormation 스택 이름과 일치하지 않아야 합니다. 클러스터를 생성한 후에는 이름을 변경할 수 없습니다.

#### ClusterConfiguration:

[클러스터 구성 파일](#)에 설명된 클러스터 구성 YAML 파일입니다. 하지만 [내장 CloudFormation 함수와](#) 같은 일반적인 구문을 사용할 수 있습니다.

#### DeletionPolicy:

루트 스택이 삭제될 때 클러스터를 삭제할지 여부를 정의합니다. 기본값은 Delete입니다.

#### 보관:

사용자 지정 리소스가 삭제되더라도 클러스터를 유지합니다.

#### Note

보존된 클러스터가 계속 작동하려면 스토리지 및 네트워킹과 같은 클러스터 종속 리소스에 보존할 삭제 정책이 설정되어 있어야 합니다.

**Delete:**

사용자 지정 리소스가 삭제된 경우 클러스터를 삭제합니다.

**Fn::GetAtt** 반환 값:

Fn::GetAtt 내장 함수는 이 유형의 지정된 속성에 대한 값을 반환합니다. Fn::GetAtt intrinsic 함수 사용에 대한 자세한 내용은 [Fn::](#) 을 참조하십시오. GetAtt

ClusterProperties:

[pcluster describe-cluster](#) 작업의 값입니다.

validationMessages:

마지막 생성 또는 업데이트 작업 중에 발생한 모든 검증 메시지를 포함하는 문자열입니다.

logGroupName:

Lambda 클러스터 작업을 로깅하는 데 사용되는 로그 그룹의 이름. 로그 이벤트는 90일 동안 보존되며 로그 그룹은 클러스터 삭제 후에도 보존됩니다.

## 예: Fn::GetAtt:

```
# Provide the public IP address of the head node as an output of a stack
Outputs:
  HeadNodeIp:
    Description: The public IP address of the head node
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]
```

## 예: AWS ParallelCluster 사용자 지정 리소스가 포함된 단순하고 완전한 CloudFormation 템플릿:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
  HeadNodeSubnet:
    Description: Subnet where the HeadNode will run
    Type: AWS::EC2::Subnet::Id

  ComputeSubnet:
    Description: Subnet where the Compute Nodes will run
    Type: AWS::EC2::Subnet::Id
```



```
KeyName:
  Description: KeyPair to login to the head node
  Type: AWS::EC2::KeyPair::KeyName

Resources:
  PclusterClusterProvider:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
        - { Version: 3.7.0 }

  PclusterCluster:
    Type: Custom::PclusterCluster
    Properties:
      ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
      ClusterName: !Sub 'c-${AWS::StackName}'
      ClusterConfiguration:
        Image:
          Os: alinux2
        HeadNode:
          InstanceType: t2.medium
          Networking:
            SubnetId: !Ref HeadNodeSubnet
          Ssh:
            KeyName: !Ref KeyName
        Scheduling:
          Scheduler: slurm
          SlurmQueues:
            - Name: queue0
              ComputeResources:
                - Name: queue0-cr0
                  InstanceType: t2.micro
              Networking:
                SubnetIds:
                  - !Ref ComputeSubnet

Outputs:
  HeadNodeIp:
    Description: The Public IP address of the HeadNode
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]
  ValidationMessages:
    Description: Any warnings from cluster create or update operations.
```

```
Value: !GetAtt PclusterCluster.validationMessages
```

CloudFormation AWS ParallelCluster 사용자 지정 리소스를 사용하는 방법에 대한 자세한 내용은 [을 참조하십시오](#) [를 사용하여 클러스터 생성 AWS CloudFormation](#).

## 클러스터 작업

클러스터 사용자 지정 리소스가 CloudFormation 스택에 추가되면 다음 클러스터 작업을 수행할 CloudFormation 수 있습니다.

- CloudFormation AWS ParallelCluster 사용자 지정 리소스가 포함된 스택을 배포할 때 별도의 새 스택에 클러스터를 생성합니다.
- 스택에 정의된 클러스터 구성을 업데이트하면 구성 업데이트 정책에 따라 클러스터가 업데이트됩니다. CloudFormation AWS ParallelCluster 사용자 지정 리소스 공급자는 클러스터를 업데이트하기 전에 컴퓨팅 플릿을 중지하지 않습니다. 클러스터 업데이트에는 [QueueUpdateStrategy](#) 설정을 사용하는 것이 좋습니다. 이렇게 하면 AWS ParallelCluster 사용자 지정 리소스를 사용할 때 업데이트 후 명시적인 `pcluster update-compute-fleet` 호출을 하지 않아도 됩니다.
- 스택을 삭제하면 클러스터도 삭제됩니다.

## 사용자 지정 리소스가 AWS ParallelCluster 포함된 문제 해결 스택

AWS ParallelCluster 사용자 지정 리소스를 사용하여 별도의 새 스택에서 클러스터를 CloudFormation 배포합니다. 다음 단계를 수행하여 클러스터 생성을 모니터링할 수 있습니다.

1. 로 CloudFormation 이동하여 탐색 창에서 스택을 선택합니다. AWS Management Console
2. 클러스터 이름에 정의한 이름을 가진 스택을 선택합니다.
3. 스택 상태가 ROLLBACK\_COMPLETE인 경우 클러스터를 생성하는 동안 오류가 발생했습니다.
4. 스택 세부 정보를 선택하고 이벤트 탭을 선택합니다.
5. 논리 ID에서 이벤트에서 클러스터 이름에 정의한 이름을 검색하세요. 문제의 원인을 알려주는 `Status reason`가 있습니다.
6. 스택 드롭다운 메뉴를 선택한 다음 삭제를 선택하여 삭제된 스택 목록을 볼 수도 있습니다. 클러스터 이름이 있는 스택을 선택하고 이벤트에서 자세한 내용을 확인하세요.
7. 클러스터를 관리하는 사용자 지정 리소스 공급자의 결과를 보려면 설명이 "AWS ParallelCluster 클러스터 사용자 지정 리소스"인 스택을 선택하십시오. 리소스 탭을 선택하고 논리 ID `PclusterCfnFunctionLogGroup`이 있는 리소스를 찾은 다음 해당 링크를 따라 이동합니다. Lambda 디버그 출력을 보여주는 로그 스트림을 확인합니다.

8. 클러스터 문제를 해결하려면 [AWS ParallelCluster 문제 해결](#) 섹션을 참조하세요.

## Elastic Fabric Adapter

Elastic Fabric Adapter(EFA)는 동일한 서브넷에 있는 다른 인스턴스와의 대기 시간이 짧은 네트워크 통신을 위한 OS 바이패스 기능을 갖춘 네트워크 디바이스입니다. EFA는 Libfabric을 사용하여 노출되며 Messaging Passing Interface(MPI)를 사용하는 애플리케이션에서 사용할 수 있습니다.

AWS ParallelCluster 및 Slurm 스케줄러와 함께 EFA를 사용하려면 [SlurmQueuesComputeResourcesEfaEnabled](#)를 설정합니다. true

EFA를 지원하는 Amazon EC2 인스턴스 목록을 보려면 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [지원되는 인스턴스 유형](#)을 참조하십시오.

배치 그룹에서 EFA 지원 인스턴스를 실행하는 것이 좋습니다. 이렇게 하면 인스턴스가 단일 가용 영역의 지연율이 낮은 그룹에서 시작됩니다. AWS ParallelCluster를 사용하여 배치 그룹을 구성하는 방법에 대한 자세한 내용은 [SlurmQueues/Networking/PlacementGroup](#)를 참조하세요.

자세한 내용은 Amazon EC2 사용 설명서의 [Elastic Fabric Adapter](#), [Elastic Fabric Adapter를 사용한 HPC 워크로드 확장 및 AWS ParallelCluster](#) 오픈 소스 블로그를 AWS 참조하십시오.

### Note

여러 가용 영역에 걸친 Elastic Fabric Adapter(EFA)는 지원되지 않습니다. [자세한 내용은 스케줄링/SlurmQueues네트워킹 /을 참조하십시오. SubnetIds](#)

### Note

기본적으로 Ubuntu 배포는 ptrace(프로세스 추적) 보호를 활성화합니다. Libfabric이 제대로 작동하도록 ptrace 보호가 비활성화됩니다. 자세한 내용은 Amazon EC2 [사용 설명서의 ptrace 보호 비활성화](#)를 참조하십시오.

## 인텔 활성화 MPI

MPI인텔은 AWS ParallelCluster AMIs 에서 사용할 수 있습니다.

**Note**

인텔을 사용하려면 [인텔 MPI 간체 소프트웨어 사용권](#) 약관을 인정하고 이에 동의해야 합니다.

기본적으로 MPI Open은 경로에 위치합니다. 열기 MPI MPI 대신 인텔을 활성화하려면 먼저 인텔 MPI 모듈을 로드해야 합니다. 그런 다음 `module load intelmpi`를 사용하여 최신 버전을 설치해야 합니다. 모듈의 정확한 이름은 업데이트할 때마다 변경됩니다. 사용 가능한 모듈을 확인하려면 `module avail`을 실행합니다. 출력값은 다음과 같습니다.

```
$ module avail
-----/usr/share/Modules/modulefiles
-----
dot                modules
libfabric-aws/1.16.0~amzn3.0  null
module-git         openmpi/4.1.4
module-info        use.own

-----/opt/intel/mpi/2021.6.0/modulefiles
-----
intelmpi
```

모듈을 로드하려면 `module load modulename`을 실행합니다. `mpirun`을 실행하는 데 사용된 스크립트에 이를 추가할 수 있습니다.

```
$ module load intelmpi
```

로드된 모듈을 확인하려면 `module list`를 실행합니다.

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

MPI인텔이 활성화되었는지 확인하려면 `mpirun --version`를 실행하십시오.

```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2021.6 Build 20220227 (id: 28877f3f32)
Copyright 2003-2022, Intel Corporation.
```

인텔 MPI 모듈이 로드되면 인텔 MPI 도구를 사용하도록 여러 경로가 변경됩니다. 인텔 MPI 도구로 컴파일한 코드를 실행하려면 먼저 인텔 MPI 모듈을 로드하십시오.

#### Note

MPI인텔은 AWS Graviton 기반 인스턴스와 호환되지 않습니다.

#### Note

AWS ParallelCluster 버전 2.5.0 이전에는 중국 (베이징) 및 중국 (닝샤) AWS ParallelCluster AMIs 지역에서 인텔을 사용할 수 MPI 없었습니다.

## AWS ParallelCluster API

AWS ParallelCluster API란 무엇인가요?

AWS ParallelCluster API는 일단 배포하면 API를 통해 AWS ParallelCluster 기능에 프로그래밍 방식으로 액세스할 수 AWS 계정있는 서버리스 애플리케이션입니다.

AWS ParallelCluster API는 기능을 제공하는 [Amazon API Gateway](#) 엔드포인트와 호출된 AWS ParallelCluster 기능을 처리하는 [AWS Lambda](#) 함수를 포함하는 독립형 [AWS CloudFormation](#) 템플릿으로 배포됩니다.

다음 이미지는 API 인프라의 상위 수준 아키텍처 다이어그램을 보여줍니다. AWS ParallelCluster

## AWS ParallelCluster API 설명서

API를 설명하는 OpenAPI 사양 파일은 다음에서 다운로드할 AWS ParallelCluster 수 있습니다.

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

[OpenAPI 사양 파일에서 시작하여 Swagger UI 또는 Redoc과 같은 사용 가능한 여러 도구 중 하나를 사용하여 AWS ParallelCluster API에 대한 문서를 생성할 수 있습니다.](#)

API 배포 방법 AWS ParallelCluster

AWS ParallelCluster API를 배포하려면 의 관리자여야 AWS 계정합니다.

API 배포에 사용되는 템플릿은 다음 URL에서 사용할 수 있습니다.

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

API를 배포해야 하는 위치와 AWS ParallelCluster 버전 (예: 3.7.0) <VERSION> 은 AWS 리전 어디입니까? <REGION>

AWS Lambda Lambda 계층 인터페이스를 와 함께 사용하여 API가 호출한 기능을 처리합니다. [AWS ParallelCluster Python 라이브러리 API](#)

### Warning

Amazon API Gateway 서비스에 대한 액세스 권한을 가진 모든 사용자는 API 리소스를 관리할 AWS Lambda AWS ParallelCluster 수 있는 권한을 자동으로 상속합니다. AWS 계정

## 를 사용하여 배포하십시오. AWS CLI

CLI와 함께 사용할 AWS 자격 증명을 구성하십시오 (아직 구성하지 않은 경우).

```
$ aws configure
```

다음 명령을 실행하여 API를 배포합니다.

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This can be any name
$ VERSION=3.7.0
$ aws cloudformation create-stack \
  --region ${REGION} \
  --stack-name ${API_STACK_NAME} \
  --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-create-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

### 배포 사용자 지정

템플릿에 표시된 AWS CloudFormation 매개변수를 사용하여 API 배포를 사용자 지정할 수 있습니다. CLI를 통해 배포할 때 파라미터 값을 구성하려면 다음 옵션을 사용할 수 있습니다: `--parameters ParameterKey=KeyName,ParameterValue=Value`.

다음 파라미터는 선택 사항입니다.

- 지역 - Region 매개변수를 사용하여 API가 리소스를 전부 AWS 리전 (기본값) 제어할 수 있는지 아니면 한 번에 제어할 수 있는지 지정합니다 AWS 리전. 액세스를 제한하려면 이 값을 배포 대상 API 로 설정하십시오. AWS 리전
- ParallelClusterFunctionRole- 이는 AWS Lambda 기능 구현 기능에 할당되는 IAM 역할을 재정의합니다. AWS ParallelCluster 파라미터는 IAM 역할의 ARN을 수락합니다. 이러한 역할을 IAM 보안 AWS Lambda 주체로 사용하도록 구성해야 합니다.
- CustomDomainName, CustomDomainCertificate, CustomDomainHostedZoneId - 이 파라미터를 사용하여 Amazon API Gateway 엔드포인트의 사용자 지정 도메인을 설정합니다. CustomDomainName은 사용할 도메인의 CustomDomainCertificate 이름이고, 은 이 도메인 이름에 대한 AWS 관리형 인증서의 CustomDomainHostedZoneId ARN이며, 레코드를 생성하려는 [Amazon Route 53](#) 호스팅 영역의 ID입니다.

#### Warning

API에 최소 버전의 전송 계층 보안(TLS)을 적용하도록 사용자 지정 도메인 설정을 구성할 수 있습니다. 자세한 내용은 [API Gateway의 사용자 지정 도메인에 대한 최소 TLS 버전 선택](#)을 참조하세요.

- EnableIamAdminAccess- 기본적으로 AWS Lambda 함수 처리 AWS ParallelCluster API 작업은 권한 있는 IAM 액세스를 차단하는 IAM 역할로 구성됩니다 (). EnableIamAdminAccess=false 이므로 인해 API는 IAM 역할 또는 정책 생성이 필요한 작업을 처리할 수 없게 됩니다. 따라서 IAM 역할이 리소스 구성의 일부로 입력으로 제공되는 경우에만 클러스터 또는 사용자 지정 이미지를 성공적으로 생성할 수 있습니다.

EnableIamAdminAccess가 AWS ParallelCluster API로 true 설정되면 클러스터를 배포하거나 사용자 지정 AMI를 생성하는 데 필요한 IAM 역할 생성을 관리할 권한이 부여됩니다.

#### Warning

이 값을 true로 설정하면 AWS Lambda 함수 처리 작업에 대한 IAM 관리자 권한이 부여됩니다. AWS ParallelCluster

이 모드를 활성화할 때 잠금 해제할 수 있는 기능에 대한 자세한 내용은 [AWS ParallelCluster IAM 리소스 관리를 위한 사용자 예제 정책](#)를 참조하세요.

- **PermissionsBoundaryPolicy**- 이 선택적 파라미터는 PC API 인프라에서 생성되는 모든 IAM 역할의 권한 경계로 설정되고 관리 IAM 권한의 조건으로 설정되는 기존 IAM 정책 ARN을 수락하므로 PC API에서 이 정책을 사용하는 역할만 생성할 수 있습니다.

이 모드에서 부과되는 제한에 대한 자세한 내용은 [PermissionsBoundary 모드](#)를 참조하세요.

- **CreateApiUserRole**- 기본적으로 API 배포에는 AWS ParallelCluster API 호출 권한이 부여된 유일한 역할로 설정된 IAM 역할 생성이 포함됩니다. Amazon API Gateway 엔드포인트는 생성된 사용자에 게만 호출 권한을 부여하는 리소스 기반 정책으로 구성됩니다. 이를 변경하려면 API 액세스를 설정한 `CreateApiUserRole=false` 다음 선택된 IAM 사용자에게 부여하십시오. 자세한 내용은 API Gateway 개발자 안내서의 [API 간접 호출을 위한 액세스 제어](#)를 참조하세요.

#### Warning

Amazon API Gateway 리소스 정책에 의해 API 엔드포인트에 `CreateApiUserRole=true` 대한 액세스가 제한되지 않는 경우, 무제한 `execute-api:Invoke` 권한을 가진 모든 IAM 역할이 기능에 액세스할 수 있습니다. AWS ParallelCluster 자세한 내용은 API Gateway 개발자 안내서의 [API Gateway 리소스 정책을 사용하는 액세스 제어](#)를 참조하세요.

#### Warning

`ParallelClusterApiUserRole`에는 모든 API 작업을 호출할 권한이 있습니다. AWS ParallelCluster API 리소스의 하위 집합에 대한 액세스를 제한하려면 API Gateway 개발자 안내서의 [IAM 정책을 사용하여 API Gateway API 메서드를 호출할 수 있는 사용자 제어](#)를 참조하세요.

- **IAM RoleAndPolicyPrefix** - 이 선택적 파라미터는 PC API 인프라의 일부로 생성된 IAM 역할과 정책 모두의 접두사로 사용되는 최대 10자의 문자열을 허용합니다.

## API 업데이트

최신 버전으로 업그레이드 AWS ParallelCluster



옵션 1: 위와 같이 해당 AWS CloudFormation 스택을 삭제하고 새 API를 배포하여 기존 API를 제거합니다.

옵션 2: 다음 명령을 실행하여 기존 API를 업데이트합니다.

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.7.0
$ aws cloudformation update-stack \
  --region ${REGION} \
  --stack-name ${API_STACK_NAME} \
  --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-update-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

## API 호출 AWS ParallelCluster

AWS ParallelCluster Amazon API Gateway 엔드포인트는 [AWS IAM 권한 부여 유형으로](#) 구성되며, 모든 요청에 유효한 IAM 자격 증명을 사용하여 SigV4로 서명해야 합니다 ([API 참조: http](#) 요청 생성).

기본 설정으로 배포하면 API로 생성한 기본 IAM 사용자에게만 API 간접 호출 권한이 부여됩니다.

기본 IAM 사용자의 ARN을 검색하려면 다음을 실행합니다.

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
  --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiUserRole'].OutputValue" --
output text
```

[기본 IAM 사용자의 임시 자격 증명을 가져오려면 STS 명령을 실행합니다. AssumeRole](#)

다음 명령을 실행하여 AWS ParallelCluster API 엔드포인트를 검색할 수 있습니다.

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
  --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiInvokeUrl'].OutputValue" --
output text
```

AWS ParallelCluster API는 다음에서 확인할 수 있는 OpenAPI 사양을 준수하는 모든 HTTP 클라이언트에서 호출할 수 있습니다.

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

요청은 [여기](#)에 설명된 대로 SigV4로 서명해야 합니다.

현재로서는 공식적인 API 클라이언트 구현을 제공하지 않습니다. 그러나 [OpenAPI 생성기](#)를 사용하면 OpenAPI 모델에서 API 클라이언트를 쉽게 생성할 수 있습니다. 클라이언트가 생성되면 SigV4 서명이 기본적으로 제공되지 않는 경우 SigV4 서명을 추가해야 합니다.

Python API 클라이언트에 대한 참조 구현은 [AWS ParallelCluster 리포지토리](#)에서 찾을 수 있습니다. Python API 클라이언트를 사용하는 방법에 대해 자세히 알아보려면 [AWS ParallelCluster API 사용하기](#) 자습서를 참조하세요.

[Amazon Cognito 또는 Lambda 권한 부여자와 같은 고급 액세스 제어 메커니즘을 구현하거나 또는 API 키를 사용하여 API를 추가로 보호하려면 Amazon API Gateway 설명서를 참조하십시오. AWS WAF](#)

#### Warning

AWS ParallelCluster API 호출 권한이 있는 IAM 사용자는 에서 관리하는 모든 리소스를 간접적으로 제어할 수 있습니다. AWS ParallelCluster 계정 여기에는 사용자 IAM 정책의 제한으로 인해 사용자가 직접 제어할 수 없는 AWS 리소스 생성이 포함됩니다. 예를 들어 AWS ParallelCluster 클러스터 생성에는 구성에 따라 Amazon EC2 인스턴스, Amazon Route 53, Amazon Elastic File System 파일 시스템, Amazon FSx 파일 시스템, IAM 역할 및 사용자가 직접 제어할 수 없는 다른 사용자의 리소스 배포가 포함될 수 있습니다.

#### Warning

구성에 AdditionalIamPolicies가 지정된 상태로 클러스터를 생성할 때는 추가 정책이 다음 패턴 중 하나와 일치해야 합니다.

- !Sub arn:\${AWS::Partition}:iam:\${AWS::AccountId}:policy/parallelcluster\*
- !Sub arn:\${AWS::Partition}:iam:\${AWS::AccountId}:policy/parallelcluster/\*
- !Sub arn:\${AWS::Partition}:iam:aws:policy/CloudWatchAgentServerPolicy
- !Sub arn:\${AWS::Partition}:iam:aws:policy/AmazonSSMManagedInstanceCore

```

- !Sub arn:${AWS::Partition}:iam::aws:policy/AWSBatchFullAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonS3ReadOnlyAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/AWSBatchServiceRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/EC2InstanceProfileForImageBuilder
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole

```

다른 추가 정책이 필요한 경우 다음 중 하나를 수행할 수 있습니다.

- DefaultParallelClusterIamAdminPolicy 편집:

```

https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml

```

ArnLike/iam:PolicyARN 섹션에 정책을 추가합니다.

- 구성 파일에서 AdditionalIamPolicies에 대한 정책을 지정하지 않고 클러스터 내에서 생성된 AWS ParallelCluster 인스턴스 역할에 정책을 수동으로 추가합니다.

## API 로그 및 지표에 액세스

API 로그는 30일 동안 보존된 CloudWatch 상태로 Amazon에 게시됩니다. API 배포와 관련된 LogGroup 이름을 검색하려면 다음 명령을 실행합니다.

```

$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --
stack-name ${API_STACK_NAME} --query "Stacks[0].Outputs[?
OutputKey=='ParallelClusterLambdaLogGroup'].OutputValue" --output text

```

Lambda 지표, 로그 및 [AWS X-Ray](#) 추적 로그는 Lambda 콘솔을 통해서도 액세스할 수 있습니다. API 배포와 관련된 Lambda 함수의 ARN을 검색하려면 다음 명령을 실행합니다.

```

$ REGION=<region>

```

```
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
  --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterLambdaArn'].OutputValue" --
output text
```

## AWS ParallelCluster 테라폼용

AWS ParallelCluster [3.8.0부터 Terraform을 사용하여 클러스터와 사용자 지정 이미지를 배포할 수 있습니다.](#) 이 기능을 사용하려면 Terraform 레지스트리의 [Terraform Provider](#)를 참조하십시오. AWS ParallelCluster

### Note

공급자를 사용하려면 계정에 [ParallelCluster API](#)가 배포되어 있어야 합니다.

다음 차트를 사용하여 제공자와 AWS ParallelCluster 버전 간의 호환성을 확인하세요.

| 제공자 버전 | AWS ParallelCluster 버전 |
|--------|------------------------|
| 1.0.0  | 3.8.0+                 |

제공자 사용 [방법의 예](#)를 참조하십시오.

더 원활한 경험을 위해 Terraform Registry의 공식 [Terraform 모듈](#)을 사용하세요. AWS ParallelCluster 이 모듈을 사용하면 다음을 배포할 수 있습니다.

1. ParallelCluster API
2. ParallelCluster YAML 구성 파일 및 HCL로 정의된 클러스터
3. 클러스터에 필요한 네트워킹 인프라 ParallelCluster

모듈 사용 방법의 [예](#)를 참조하십시오.

## NICE DCV를 통해 헤드 노드에 연결합니다.

NICE DCV는 사용자가 원격 고성능 서버에 호스팅된 그래픽 집약형 3D 애플리케이션에 안전하게 연결할 수 있는 원격 시각화 기술입니다. 자세한 내용은 [NICE DCV](#)를 참조하세요.

NICE DCV 소프트웨어는 헤드 노드에 자동으로 설치되며 [HeadNode](#) 구성의 [Dcv](#) 섹션을 사용하여 활성화할 수 있습니다.

```
HeadNode:
  Dcv:
    Enabled: true
```

이렇게 하면 헤드 노드를 [DCV 서버 스토리지](#) 폴더로 AWS ParallelCluster 설정합니다/  
home/<DEFAULT\_AMI\_USER>. NICE DCV 구성 파라미터에 대한 자세한 내용은 [HeadNode/Dcv](#) 섹션을 참조하세요. NICE DCV 세션에 연결하려면 [pcluster dcv-connect](#) 명령을 사용합니다.

## NICE DCV HTTPS 인증서

NICE DCV는 NICE DCV 클라이언트와 NICE DCV 서버 간의 트래픽을 보호하기 위해 자체 서명된 인증서를 자동으로 생성합니다.

기본 자체 서명 NICE DCV 인증서를 다른 인증서로 바꾸려면 먼저 헤드 노드에 연결합니다. 그런 다음 [pcluster dcv-connect](#) 명령을 실행하기 전에 인증서와 키를 모두 /etc/dcv 폴더에 복사합니다.

자세한 내용은 NICE DCV 관리자 안내서의 [TLS 인증서 변경](#)을 참조하세요.

## NICE DCV 라이선스

NICE DCV 서버는 Amazon EC2 인스턴스에서 실행될 때 라이선스 서버가 필요하지 않습니다. 그러나 NICE DCV 서버는 정기적으로 Amazon S3 버킷에 연결하여 유효한 라이선스를 사용할 수 있는지 여부를 확인해야 합니다.

AWS ParallelCluster 헤드 노드 IAM 정책에 필요한 권한을 자동으로 추가합니다. 사용자 지정 IAM 인스턴스 정책을 사용할 때는 [NICE DCV 관리자 안내서의 Amazon EC2 기반 NICE DCV에](#) 설명된 권한을 사용하십시오.

문제 해결 팁은 [NICE DCV 문제 해결](#) 섹션을 참조하세요.

## pcluster update-cluster 사용하기

AWS ParallelCluster 3.x에서는 현재 클러스터를 생성하는 데 사용된 설정과 구성 파일의 설정을 [pcluster update-cluster](#) 분석하여 문제를 확인합니다. 문제가 발견되면 해당 문제가 보고되고 문제 해결을 위해 취해야 할 단계가 표시됩니다. 예를 들어 [InstanceType](#) 컴퓨팅이 변경된 경우 업데이트를 진행하려면 먼저 컴퓨팅 플릿을 중지해야 합니다. 이 문제는 발견되면 보고됩니다. 차단 문제가 발견되지 않으면 업데이트 프로세스가 시작되고 변경 사항이 보고됩니다.

[pcluster update-cluster](#) `--dryrun` option를 사용하여 실행 전에 변경 사항을 확인할 수 있습니다. 자세한 정보는 [pcluster update-cluster 예제](#)를 참조하세요.

문제 해결에 대한 도움말은 [AWS ParallelCluster 문제 해결](#) 섹션을 참조하세요.

## 업데이트 정책: 정의

**업데이트 정책:** 이 설정은 업데이트 중에 변경할 수 있습니다.

이 설정을 변경한 후 [pcluster update-cluster](#)를 사용하여 클러스터를 업데이트할 수 있습니다.

**업데이트 정책:** 이 설정을 변경하면 업데이트가 허용되지 않습니다.

이 설정을 변경한 후에는 클러스터를 업데이트할 수 없습니다. 원래 클러스터의 설정을 되돌리고 업데이트된 설정으로 새 클러스터를 생성해야 합니다. 나중에 원래 클러스터를 삭제할 수 있습니다. [pcluster create-cluster](#)를 사용하여 새 클러스터를 생성합니다. 원래 클러스터를 삭제하려면 [pcluster delete-cluster](#)를 사용합니다.

**업데이트 정책:** 이 설정은 업데이트 중에 분석되지 않습니다.

이 설정들은 변경할 수 있으며 [pcluster update-cluster](#)를 사용하여 클러스터가 업데이트됩니다.

**업데이트 정책:** 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

컴퓨팅 플릿이 존재하는 동안에는 이러한 설정을 변경할 수 없습니다. 변경 내용을 되돌리거나 컴퓨팅 플릿을 중지([pcluster update-compute-fleet](#) 사용)해야 합니다. 컴퓨팅 플릿이 중지된 후 클러스터([pcluster update-cluster](#))를 업데이트하여 변경 사항을 활성화할 수 있습니다. 예를 들어 [SlurmQueues/ComputeResources/ - Name/MinCount > 0](#)인 Slurm 스케줄러를 사용하는 경우 컴퓨팅 플릿이 시작됩니다.

**업데이트 정책:** 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿과 로그인 노드를 중지해야 합니다.

컴퓨팅 플릿이 존재하거나 로그인 노드가 사용 중인 경우에는 이러한 설정을 변경할 수 없습니다. 변경 내용을 되돌리거나 컴퓨팅 플릿과 로그인 노드를 중지해야 합니다 (컴퓨팅 플릿 사용을 중지할 수 [pcluster update-compute-fleet](#) 있음). 컴퓨팅 플릿과 로그인 노드가 중지된 후 클러스터 ([pcluster update-cluster](#)) 를 업데이트하여 변경 사항을 활성화할 수 있습니다.

**업데이트 정책:** 업데이트 중에는 이 설정을 줄일 수 없습니다.

이러한 설정은 변경할 수 있지만 줄일 수는 없습니다. 이러한 설정을 줄여야 하는 경우 원래 클러스터의 설정을 되돌리고 업데이트된 설정으로 새 클러스터를 생성해야 합니다. 나중에 원래 클러스터

를 삭제할 수 있습니다. [pcluster create-cluster](#)를 사용하여 새 클러스터를 생성합니다. 원래 클러스터를 삭제하려면 [pcluster delete-cluster](#)를 사용합니다.

**업데이트 정책:** 이 설정을 변경하면 업데이트가 허용되지 않습니다. 강제로 업데이트하면 새 값이 무시되고 이전 값이 사용됩니다.

이 설정을 변경한 후에는 클러스터를 업데이트할 수 없습니다. 원래 클러스터의 설정을 되돌리고 업데이트된 설정으로 새 클러스터를 생성해야 합니다. 나중에 원래 클러스터를 삭제할 수 있습니다. [pcluster create-cluster](#)를 사용하여 새 클러스터를 생성합니다. 원래 클러스터를 삭제하려면 [pcluster delete-cluster](#)를 사용합니다.

**업데이트 정책:** 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)를 설정해야 합니다.

이러한 설정은 변경할 수 있습니다. 컴퓨팅 플릿을 중지([pcluster update-compute-fleet](#) 사용)하거나 [QueueUpdateStrategy](#)가 설정되어야 합니다. 컴퓨팅 플릿이 중지되거나 [QueueUpdateStrategy](#)가 설정된 이후에는 클러스터([pcluster update-cluster](#))를 업데이트하여 변경 사항을 활성화할 수 있습니다.

#### Note

이 업데이트 정책은 AWS ParallelCluster 버전 3.2.0부터 지원됩니다.

**업데이트 정책:** 이 목록 값 설정의 경우 업데이트 중에 새 값을 추가할 수 있으며, 또는 기존 값을 제거할 때 컴퓨팅 플릿을 중지해야 합니다.

업데이트 중에 이러한 설정의 새 값을 추가할 수 있습니다. 목록에 새 값을 추가한 후 ([pcluster update-cluster](#))를 사용하여 클러스터를 업데이트할 수 있습니다.

목록에서 기존 값을 제거하려면 컴퓨팅 플릿을 중지([pcluster update-compute-fleet](#) 사용)해야 합니다.

예를 들어 Slurm 스케줄러를 사용하고 새 인스턴스 유형을 [InstanceTypeInstances/](#)에 추가하는 경우 컴퓨팅 플릿을 중지하지 않고도 클러스터를 업데이트할 수 있습니다. [InstanceTypeInstances/](#)에서 기존 인스턴스 유형을 제거하려면 먼저 컴퓨팅 플릿을 중지해야 합니다 ([pcluster](#) 사용). [update-compute-fleet](#)

#### Note

이 업데이트 정책은 버전 3.2.0부터 지원됩니다. AWS ParallelCluster

업데이트 정책: 대기열 크기를 줄이려면 컴퓨팅 플릿을 중지하거나 TAREMITE로 [QueueUpdateStrategy](#) 설정해야 업데이트를 위해 이 설정을 변경할 수 있습니다.

이러한 설정은 변경할 수 있지만 변경으로 인해 대기열 크기가 줄어드는 경우에는 `pcluster update-compute-fleet` 를 사용하여 컴퓨팅 플릿을 중지하거나 TERMINATE로 [QueueUpdateStrategy](#) 설정해야 합니다. 컴퓨팅 플릿이 중지되거나 [QueueUpdateStrategy](#) TERMINATE로 설정된 후에는 클러스터를 업데이트 (`pcluster update-cluster`) 하여 변경 사항을 활성화할 수 있습니다.

클러스터 용량 크기 조정 시 TERMINATE 설정은 노드 목록 뒤에 있는 노드만 종료하고 동일한 파티션의 다른 모든 노드는 그대로 유지합니다.

예를 들어, 클러스터 초기 용량이 `MinCount = 5` 및 `MaxCount = 10` 인 경우 노드는 다음과 같습니다. `st-[1-5]; dy-[1-5]` 클러스터 크기를 `MinCount = 3` 및 `MaxCount = 5` 로 조정하면 새 클러스터 용량이 `st-[1-3]; dy-[1-2]` 노드별로 구성되므로 업데이트 중에는 변경되지 않습니다. 업데이트 `st-[4-5]; dy-[3-5]` 중에는 노드만 종료됩니다.

다음 변경 사항이 지원되며 컴퓨팅 플릿을 중지하거나 TERMINATE로 [QueueUpdateStrategy](#) 설정할 필요가 없습니다.

- 새 [SlurmQueue](#) 항목이 추가되었습니다.
- 새 [ComputeResource](#) 항목이 추가되었습니다.
- [MaxCount](#) 증가했습니다.
- [MinCount](#) 적어도 같은 양만큼 [MaxCount](#) 증가하고 증가합니다.

참고: 이 업데이트 정책은 AWS ParallelCluster 버전 3.9.0부터 지원됩니다.

업데이트 정책: 이 목록 값 설정의 경우 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)가 새 값을 추가하도록 설정해야 합니다. 기존 값을 제거할 때는 컴퓨팅 플릿을 중지해야 합니다.

업데이트 중에 이러한 설정의 새 값을 추가할 수 있습니다. 컴퓨팅 플릿을 중지(`pcluster update-compute-fleet` 사용)하거나 [QueueUpdateStrategy](#)가 설정되어야 합니다. 컴퓨팅 플릿이 중지되거나 [QueueUpdateStrategy](#)가 설정된 이후에는 클러스터(`pcluster update-cluster`)를 업데이트하여 변경 사항을 활성화할 수 있습니다.

목록에서 기존 값을 제거하려면 컴퓨팅 플릿을 중지(`pcluster update-compute-fleet` 사용)해야 합니다.

#### Note

이 업데이트 정책은 AWS ParallelCluster 버전 3.3.0부터 지원됩니다.



업데이트 정책: 관리형 배치 그룹 삭제를 위해서는 모든 컴퓨팅 노드를 중지해야 합니다. 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)를 설정해야 합니다.

관리형 배치 그룹을 제거하려면 컴퓨팅 플릿을 중지([pcluster update-compute-fleet](#) 사용)해야 합니다. 컴퓨팅 플릿을 중지하기 전에 클러스터 업데이트를 실행하여 관리형 배치 그룹을 제거하면 잘못된 구성 메시지가 반환되고 업데이트가 진행되지 않습니다. 컴퓨팅 플릿을 중지하면 실행 중인 인스턴스가 없음을 보장할 수 있습니다.

## pcluster update-cluster 예제

이러한 설정을 변경할 수 있지만 변경으로 인해 대기열 크기가 줄어드는 경우 컴퓨팅 플릿을 중지 ([pcluster](#) 사용 [update-compute-fleet](#)) 하거나 TERMINATE로 [QueueUpdateStrategy](#) 설정해야 합니다. 컴퓨팅 플릿이 중지되거나 [QueueUpdateStrategy](#) TERMINATE로 설정된 후에는 클러스터를 업데이트 ([pcluster update-cluster](#)) 하여 변경 사항을 활성화할 수 있습니다.

- 이 예제는 일부 허용된 변경 사항이 포함된 업데이트를 보여 주며 업데이트가 바로 시작됩니다.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
  "cluster": {
    "clusterName": cluster_name,
    "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
    "cloudformationStackArn": stack_arn,
    "region": "us-east-1",
    "version": "3.7.0",
    "clusterStatus": "UPDATE_IN_PROGRESS"
  },
  "changeSet": [
    {
      "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
      "requestedValue": [
        "sg-0cd61884c4ad11234"
      ],
      "currentValue": [
        "sg-0cd61884c4ad16341"
      ]
    }
  ]
}
```

- 이 예제는 일부 허용된 변경 사항이 포함된 `dryrun` 업데이트를 보여줍니다. `Dryrun`은 업데이트를 시작하지 않고 변경 세트를 보고하는 데 유용합니다.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1 --dryrun true
{
  "message": "Request would have succeeded, but DryRun flag is set.",
  "changeSet": [
    {
      "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
      "requestedValue": [
        "sg-0cd61884c4ad11234"
      ],
      "currentValue": [
        "sg-0cd61884c4ad16341"
      ]
    }
  ]
}
```

- 이 예제는 업데이트를 차단하는 일부 변경 사항이 포함된 업데이트를 보여줍니다.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
  "message": "Update failure",
  "updateValidationErrors": [
    {
      "parameter": "HeadNode.Ssh.KeyName",
      "requestedValue": "mykey_2",
      "message": "Update actions are not currently supported for the 'KeyName'
parameter. Restore 'KeyName' value to 'jenkinsjun'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
      "currentValue": "mykey_1"
    },
    {
      "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
      "requestedValue": "c4.xlarge",
      "message": "All compute nodes must be stopped. Stop the compute fleet with the
pcluster update-compute-fleet command",
      "currentValue": "t2.micro"
    }
  ],
}
```

```

    {
      "parameter": "SharedStorage[ebs1].MountDir",
      "requestedValue": "/my/very/very/long/shared_dir",
      "message": "Update actions are not currently supported for the 'MountDir'
parameter. Restore 'MountDir' value to '/shared'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
      "currentValue": "/shared"
    }
  ],
  "changeSet": [
    {
      "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
      "requestedValue": [
        "sg-0cd61884c4ad11234"
      ],
      "currentValue": [
        "sg-0cd61884c4ad16341"
      ]
    },
    {
      "parameter": "HeadNode.Ssh.KeyName",
      "requestedValue": "mykey_2",
      "currentValue": "mykey_1"
    },
    {
      "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
      "requestedValue": "c4.xlarge",
      "currentValue": "t2.micro"
    },
    {
      "parameter": "SharedStorage[ebs1].MountDir",
      "requestedValue": "/my/very/very/long/shared_dir",
      "currentValue": "/shared"
    }
  ]
}

```

## AWS ParallelCluster AMI 사용자 지정

사용자 지정 AMI를 구축해야 하는 AWS ParallelCluster 시나리오가 있습니다. 이 섹션에서는 사용자 지정 AWS ParallelCluster AMI를 구축할 때 고려해야 할 사항을 다룹니다.

다음 방법 중 하나를 사용하여 사용자 지정 AWS ParallelCluster AMI를 구축할 수 있습니다.

1. [빌드 이미지 구성 파일](#)을 생성한 다음 pcluster CLI를 사용하여 EC2 Image Builder로 이미지를 빌드합니다. 이 프로세스는 자동화되고 반복 가능하며 모니터링을 지원합니다. 자세한 내용은 [pcluster](#) 이미지 명령을 참조하세요.
2. AWS ParallelCluster AMI에서 인스턴스를 생성한 다음 로그인하여 수동으로 수정합니다. 마지막으로 Amazon EC2를 사용하여 수정된 인스턴스로부터 새 AMI를 생성합니다. 이 프로세스에는 시간이 걸립니다. 하지만 자동화나 반복이 불가능하며 pcluster CLI 이미지 모니터링 명령 사용을 지원하지 않습니다.

이 방법들에 대한 자세한 내용은 [사용자 지정 AWS ParallelCluster AMI 구축](#) 섹션을 참조하세요.

## AWS ParallelCluster AMI 사용자 지정 고려 사항

사용자 지정 이미지를 생성하는 방법에 관계없이 예비 검증 테스트를 수행하고 생성 중인 이미지의 상태를 모니터링하기 위한 조항을 포함하는 것이 좋습니다.

pcluster를 사용하여 사용자 지정 AMI를 구축하려면 [EC2 Image Builder](#)에서 사용자 지정 이미지를 구축하는 데 사용하는 [Build](#) 및 [Image](#) 섹션이 포함된 [빌드 이미지 구성 파일](#)을 생성합니다. Build 섹션에서는 Image Builder에서 이미지를 빌드하는 데 필요한 사항을 지정합니다. 여기에는 [ParentImage](#)(기본 이미지), 및 [Components](#)가 포함됩니다. [Image Builder 구성 요소](#)는 이미지를 생성하기 전에 인스턴스를 사용자 지정하거나 생성된 이미지로 시작된 인스턴스를 테스트하는 데 필요한 일련의 단계를 정의합니다. AWS ParallelCluster 구성 요소 예제는 [사용자 지정 AMI](#)를 참조하십시오. Image 섹션은 이미지 속성을 지정합니다.

사용자 지정 이미지를 [build-image](#) 생성하기 위해 pcluster에서 호출되면 Image Builder는 AWS ParallelCluster 콕북과 함께 빌드 이미지 구성을 사용하여 사용자 이미지를 AWS ParallelCluster 부트스트랩합니다. [ParentImage](#) Image Builder는 구성 요소를 다운로드하고, 빌드 및 검증 단계를 실행하고, AMI를 생성하고, AMI에서 인스턴스를 시작하고, 테스트를 실행합니다. 프로세스가 완료되면 Image Builder는 새 이미지나 중지 메시지를 생성합니다.

## 사용자 지정 구성 요소 검증 테스트 수행

Image Builder 구성 요소를 구성에 포함하기 전에 다음 방법 중 하나를 사용하여 구성 요소를 테스트하고 검증하세요. Image Builder 프로세스에는 최대 1시간이 걸릴 수 있으므로 구성 요소를 미리 테스트하는 것이 좋습니다. 이렇게 하면 시간을 크게 절약할 수 있습니다.

## 스크립트 케이스

빌드 이미지 프로세스 외부에서 실행 중인 인스턴스에서 스크립트를 테스트하고 스크립트가 종료 코드 0으로 종료되는지 확인합니다.

### Amazon 리소스 이름(ARN) 케이스

빌드 이미지 프로세스 외부에서 실행 중인 인스턴스에서 구성 요소 문서를 테스트합니다. 요구 사항 목록은 Image Builder 사용 설명서의 [구성 요소 관리자](#)를 참조하세요.

검증에 성공하면 구성 요소를 빌드 이미지 구성에 추가합니다.

사용자 지정 구성 요소가 작동하는지 확인한 후 해당 구성 요소를 [빌드 이미지 구성 파일](#)에 추가합니다.

디버깅을 지원하는 **pcluster** 명령을 사용하여 Image Builder 프로세스를 모니터링합니다.

### [describe-image](#)

이 명령을 사용하여 빌드 이미지 상태를 모니터링할 수 있습니다.

### [list-image-log-streams](#)

이 명령을 사용하면 [get-image-log-events](#)로 로그 이벤트를 검색하는 데 사용할 수 있는 로그 스트림의 ID를 가져올 수 있습니다.

### [get-image-log-events](#)

이 명령을 사용하면 빌드 이미지 프로세스 이벤트의 로그 스트림을 가져올 수 있습니다.

예를 들어 다음 명령을 사용하여 이미지 빌드 이벤트를 중단할 수 있습니다.

```
$ watch -n 1 'pcluster get-image-log-events -i <image-id> \
  --log-stream-name/1 <pcluster-version> \
  --query "events[*].message" | tail -n 50'
```

### [get-image-stack-events](#)

이 명령을 사용하면 Image Builder가 생성하는 스택에 대한 이미지 스택 이벤트를 검색할 수 있습니다.

### [export-image-logs](#)

이 명령을 사용하면 이미지 로그를 저장할 수 있습니다.

AWS ParallelCluster 로그와 CloudWatch Amazon에 대한 자세한 내용은 [Amazon CloudWatch Logs 빌드 이미지 로그](#) 및 [을 참조하십시오.아마존 CloudWatch 대시보드](#).

## 기타 고려 사항

### 새 AWS ParallelCluster 릴리스 및 사용자 지정 AMI

사용자 지정 AMI를 빌드하고 사용한다면, 새로운 각 AWS ParallelCluster 릴리스에서 사용자 지정 AMI를 생성하는 데 사용된 단계를 반복해야 합니다.

### 사용자 지정 부트스트랩 작업

[사용자 지정 부트스트랩 작업](#) 섹션을 검토하여 원하는 수정 내용을 스크립트로 작성하고 향후 AWS ParallelCluster 릴리스에서 지원할 수 있는지 확인하십시오.

### 사용자 지정 AMI 사용

[Image/CustomAmi](#) 및 [Scheduling/SlurmQueues/ - Name/Image/CustomAmi](#) 섹션의 클러스터 구성에서 사용자 지정 AMI를 지정할 수 있습니다.

사용자 지정 AMI 검증 경고 문제를 해결하려면 [사용자 지정 AMI 문제 해결](#)을 참조하세요.

## 온디맨드 용량 예약 (ODCR) 으로 인스턴스 시작

[온디맨드 용량 예약 \(ODCR\)](#) 을 사용하면 특정 가용 영역에서 클러스터 Amazon EC2 인스턴스의 용량을 예약할 수 있습니다. 따라서 [절감형 플랜](#) 또는 [리전 예약 인스턴스](#)에서 제공하는 결제 할인과는 별도로 용량 예약을 생성 및 관리할 수 있습니다.

ODCR을 구성할 open 수 있습니다. targeted 오픈 ODCR은 ODCR 특성과 일치하는 모든 인스턴스를 포함합니다. 이러한 속성은 인스턴스 유형, 플랫폼 및 가용 영역입니다. 클러스터 구성에서 대상 ODCR을 명시적으로 정의해야 합니다. ODCR이 open ODM인지 여부를 확인하려면 AWS CLI Amazon [describe-capacity-reservation](#)EC2 명령을 실행하십시오. targeted

[클러스터 배치 그룹 온디맨드 용량 예약\(CPG ODCR\)](#)이라고 하는 클러스터 배치 그룹에서 ODCR을 생성할 수도 있습니다.

여러 ODCR을 리소스 그룹으로 그룹화할 수 있습니다. 이는 클러스터 구성 파일에서 정의할 수 있습니다. 리소스 그룹에 대한 자세한 내용은 리소스 그룹 및 태그 사용 설명서의 [리소스 그룹이란 무엇입니까?](#)를 참조하세요.

## ODCR을 다음과 같이 사용할 수 있습니다. AWS ParallelCluster

AWS ParallelCluster 개방형 ODCR을 지원합니다. 개방형 ODCR을 사용하는 경우 AWS ParallelCluster에서 아무 것도 지정할 필요가 없습니다. 클러스터의 인스턴스는 자동으로 선택됩니다. 기존 배치 그룹을 지정하거나 새 배치 그룹을 AWS ParallelCluster 생성하도록 할 수 있습니다.

### 클러스터 구성의 ODCR

AWS ParallelCluster 버전 3.3.0부터 Amazon EC2 실행 인스턴스 재정의의 지정할 필요 없이 클러스터 구성 파일에서 ODCR을 정의할 수 있습니다.

먼저 링크된 각 문서에 설명된 방법을 사용하여 [용량 예약](#) 및 [리소스 그룹](#)을 생성합니다. 메서드를 사용하여 용량 예약 그룹을 생성해야 합니다. AWS CLI 를 사용하는 경우 태그 기반 또는 스택 기반 리소스 그룹만 생성할 수 있습니다. AWS Management Console태그 기반 및 스택 기반 리소스 그룹은 용량 예약이 있는 인스턴스를 시작할 AWS CLI 때 지원되지 않습니다. AWS ParallelCluster

용량 예약과 리소스 그룹을 생성한 후에는 다음 예제 클러스터 구성에 나온 것과 같이 [SlurmQueues/CapacityReservationTarget](#) 또는 [SlurmQueues/ComputeResources/CapacityReservationTarget](#)에 지정합니다. 빨간색으로 강조 표시된 #들을 유효한 값으로 바꾸세요.

```
Image:
  Os: os
HeadNode:
  InstanceType: head_node_instance
  Networking:
    SubnetId: public_subnet_id
  Ssh:
    KeyName: key_name
Scheduling:
  Scheduler: scheduler
SlurmQueues:
  - Name: queue1
    Networking:
      SubnetIds:
        - private_subnet_id
  ComputeResources:
    - Name: cr1
      Instances:
        - InstanceType: instance
      MaxCount: max_queue_size
```

```

MinCount: max_queue_size
Efa:
  Enabled: true
CapacityReservationTarget:
  CapacityReservationResourceGroupArn: capacity_reservation_arn

```

더 이상 사용되지 않음/ 권장되지 않음 - Amazon EC2 인스턴스 재정의의 사용하는 대상 ODCR

#### Warning

- AWS ParallelCluster 버전 3.3.0부터는 이 방법을 사용하지 않는 것이 좋습니다. 이 섹션은 이전 버전을 사용한 구현을 위한 참고 자료로 남아 있습니다.
- 이 방법은 Slurm을 사용한 다중 인스턴스 유형 할당과 호환되지 않습니다.

targetedODCR에 대한 지원은 3.1.1에 추가되었습니다. AWS ParallelCluster 이번 릴리스에는 EC2 RunInstances 파라미터를 재정의하고 AWS ParallelCluster에서 구성된 각 컴퓨팅 리소스에 사용할 예약에 대한 정보를 전달하는 메커니즘이 도입되었습니다. 이 메커니즘은 ODCR과 호환됩니다. targeted 하지만 targeted ODCR을 사용할 때는 run-instances 오버라이드 구성을 지정해야 합니다. Amazon EC2 AWS CLI 명령에 대상 ODCR을 명시적으로 정의해야 합니다. [run-instances](#) ODCR 여부를 open 확인하거나 AWS CLI Amazon EC2 targeted [describe-capacity-reservation](#) 명령을 실행하십시오.

여러 ODCR을 리소스 그룹으로 그룹화할 수 있습니다. 이는 실행 인스턴스 재정의에서 동시에 여러 ODCR을 대상으로 하는 데 사용할 수 있습니다.

targeted ODCR을 사용하는 경우 배치 그룹을 지정할 수 있습니다. 하지만 run-instances 재정의의 구성도 지정해야 합니다.

대신 targeted ODCR을 AWS 생성했거나 특정 예약 인스턴스 세트를 보유하고 있다고 가정해 보겠습니다. 그러면 배치 그룹을 지정할 수 없습니다. 에서 구성한 규칙이 배치 그룹 설정과 AWS 충돌할 수 있습니다. 따라서 애플리케이션에 배치 그룹이 필요한 경우 [CPG ODCR](#)을 사용하세요. 두 경우 모두 run-instances 재정의의 구성도 지정해야 합니다.

CPG ODCR을 사용하는 경우, run-instances 재정의의 구성을 지정하고 클러스터 구성에서 동일한 배치 그룹을 지정해야 합니다.



## 예약 인스턴스 사용: AWS ParallelCluster

예약 인스턴스는 용량 예약 (ODCR)과 [다릅니다](#). 예약 인스턴스에는 [2가지 유형](#)이 있습니다. 리전 예약 인스턴스에서는 용량을 예약하지 않습니다. 영역 예약 인스턴스에서는 지정된 가용 영역에서 용량을 예약합니다.

리전 예약 인스턴스를 사용하는 경우 용량 예약이 없으므로 용량 부족 오류가 발생할 수 있습니다. 영역 예약 인스턴스를 사용하는 경우 용량을 예약한 것이지만 용량을 지정하는 데 사용할 수 있는 `run-instances` API 파라미터가 없습니다.

예약 인스턴스는 모든 AWS ParallelCluster 버전에서 지원됩니다. 아무 것도 지정할 필요가 AWS ParallelCluster 없으며 인스턴스가 자동으로 선택됩니다.

영역 예약 인스턴스를 사용하는 경우 클러스터 구성에서 배치 그룹 사양을 생략하여 잠재적인 용량 부족 오류를 방지할 수 있습니다.

더 이상 사용되지 않음/권장하지 않음 - **targeted** 온디맨드 용량 예약 (ODCR) AWS ParallelCluster 3에서 **RunInstances** 사용자 지정 사용

### Warning

- AWS ParallelCluster 버전 3.3.0부터는 이 방법을 사용하지 않는 것이 좋습니다. 이 섹션은 이전 버전을 사용한 구현을 위한 참고 자료로 남아 있습니다.
- 이 방법은 Slurm을 사용한 다중 인스턴스 유형 할당과 호환되지 않습니다.

클러스터 대기열에 구성된 각 컴퓨팅 리소스의 Amazon RunInstances EC2 파라미터를 재정의할 수 있습니다. 이렇게 하려면 클러스터의 헤드 노드에 다음 코드 스니펫 콘텐츠로 `/opt/slurm/etc/pcluster/run_instances_overrides.json` 파일을 생성하세요.

- `${queue_name}`은 재정의할 대기열의 이름입니다.
- `${compute_resource_name}`은 재정의할 컴퓨팅 리소스입니다.
- `${overrides}`는 대기열과 인스턴스 유형의 특정 조합에 사용할 RunInstances 재정의 목록이 포함된 임의의 JSON 객체입니다. 재정의 구문은 [run\\_instances](#) boto3 호출에 설명된 것과 동일한 사양을 따라야 합니다.

```
{
  "${queue_name}": {
    "${compute_resource_name}": {
```

```

        ${overrides}
    },
    ...
},
...
}

```

예를 들어, 다음 JSON은 my-queue 및 my-compute-resource에서 구성된 p4d.24xlarge 인스턴스에 사용할 ODCR 그룹 group\_arn을 구성합니다.

```

{
  "my-queue": {
    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationResourceGroupArn": "group_arn"
        }
      }
    }
  }
}

```

이 JSON 파일이 생성되면 클러스터 확장을 담당하는 AWS ParallelCluster 데몬은 자동으로 인스턴스 시작을 위한 재정의의 구성을 사용합니다. 지정된 파라미터가 인스턴스 프로비저닝에 사용되고 있는지 확인하려면 다음 로그 파일을 살펴보세요.

- /var/log/parallelcluster/clustermgtd(고정 용량용)
- /var/log/parallelcluster/slurm\_resume.log(동적 용량용)

파라미터가 정확하면 다음을 포함하는 로그 항목을 찾을 수 있습니다.

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

더 이상 사용되지 않음/권장되지 않음 - 온디맨드 용량 예약 (ODCR) 을 사용하여 클러스터를 생성하십시오 **targeted**.

#### Warning

- AWS ParallelCluster 버전 3.3.0부터는 이 방법을 사용하지 않는 것이 좋습니다. 이 섹션은 이전 버전을 사용한 구현을 위한 참고 자료로 남아 있습니다.

- 이 방법은 [Slurm](#)을 사용하여 여러 인스턴스 유형 할당과 호환되지 않습니다.

## 1. 그룹 용량까지 리소스 그룹을 생성합니다.

```
$ aws resource-groups create-group --name EC2CRGroup \
  --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
  '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

### Note

리소스 그룹은 다른 계정과 공유하는 리소스를 지원하지 않습니다. 대상 ODCR을 다른 계정과 공유하는 경우에는 리소스 그룹을 만들 필요가 없습니다. 3단계의 리소스 그룹 대신 CapacityReservationId를 사용합니다.

```
#!/bin/bash
set -e

# Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
  "my-queue": {
    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationId": "cr-abcdef01234567890"
        }
      }
    }
  }
}
EOF
```

리소스 그룹에 용량 예약을 추가합니다. 새 ODCR을 생성할 때마다 그룹 예약에 추가합니다. 계정 ID, 용량 예약 ID, **PLACEHOLDER\_CAPACITY\_RESERVATION** AWS 리전 ID (예: us-east-1) **ACCOUNT\_IDREGION\_ID**로 바꾸십시오.

```
$ aws resource-groups group-resources --region REGION_ID --group EC2CRGroup \
  --resource-arns arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-
  reservation/PLACEHOLDER_CAPACITY_RESERVATION
```

로컬 컴퓨터에서 정책 문서를 생성합니다. 계정 **ACCOUNT\_ID**와 AWS 리전 ID (예: us-east-1) **REGION\_ID**로 바꾸십시오.

```
cat > policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RunInstancesInCapacityReservation",
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-reservation/*",
        "arn:aws:resource-groups:REGION_ID:ACCOUNT_ID:group/*"
      ]
    }
  ]
}
EOF
```

2. 생성한 json 파일을 AWS 계정 사용하여 IAM 정책을 생성합니다.

```
$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-
  document file://policy.json
```

3. 다음 설치 후 스크립트를 인스턴스에 로컬로 생성하고 이름을 postinstall.sh로 지정합니다.

사용자 **ACCOUNT\_ID** ID와 사용자 AWS 계정 AWS 리전 ID (예: us-east-1) **REGION\_ID** 로 교체하십시오.

```
#!/bin/bash
set -e

# Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
  "my-queue": {
```

```

    "my-compute-resource": {
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:REGION_ID:ACCOUNT_ID:group/EC2CRGroup"
        }
      }
    }
  }
}
EOF

```

Amazon S3 버킷에 이미지 파일을 업로드합니다. `S3_NAME_BUCKET#` 특정 S3 버킷 이름으로 바꿉니다.

```

$ aws s3 mb s3://S3_NAME_BUCKET
aws s3 cp postinstall.sh s3://S3_NAME_BUCKET/postinstall.sh

```

4. 자리 표시자를 자체 값으로 대체하여 로컬 클러스터 구성을 생성합니다.

```

Region: REGION_ID
Image:
  Os: alinux2
HeadNode:
  InstanceType: c5.2xlarge
  Ssh:
    KeyName: YOUR_SSH_KEY
  Iam:
    S3Access:
      - BucketName: S3_NAME_BUCKET
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::ACCOUNT_ID:policy/RunInstancesCapacityReservation
## This post-install script is executed after the node is configured.
## It is used to install scripts at boot time and specific configurations
## In the script below we are overriding the calls to RunInstance to force
## the provisioning of our my-queue partition to go through
## the On-Demand Capacity Reservation
CustomActions:
  OnNodeConfigured:
    Script: s3://S3_NAME_BUCKET/postinstall.sh
Networking:
  SubnetId: YOUR_PUBLIC_SUBNET_IN_TARGET_AZ

```

```

Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: my-queue
      ComputeResources:
        - MinCount: 0
          MaxCount: 100
          InstanceType: p4d.24xlarge
          Name: my-compute-resource
          Efa:
            Enabled: true
      Networking:
        ## PlacementGroup:
        ##   Enabled: true ## Keep PG disabled if using targeted ODCR
        SubnetIds:
          - YOUR_PRIVATE_SUBNET_IN_TARGET_AZ

```

## 5. 클러스터를 생성합니다.

다음 명령을 사용하여 클러스터를 생성합니다. 구성 파일 이름을 *cluster-config.yaml*으로 바꾸고, 클러스터 이름을 *cluster-dl*로 바꾸고 *REGION\_ID*를 사용자의 리전 ID(예: us-east-1)로 바꾸세요.

```

$ pcluster create-cluster --cluster-configuration cluster-config.yaml --cluster-name cluster-dl --region REGION_ID

```

클러스터가 생성되면 헤드 노드에서 설치 후 스크립트가 실행됩니다. 스크립트는 `run_instances_overrides.json` 파일을 생성하고 `RunInstances`에 대한 호출을 재정의하여 온디맨드 용량 예약을 통해 파티션 프로비저닝을 강제로 진행합니다.

클러스터 확장을 담당하는 AWS ParallelCluster 데몬은 새로 시작하는 인스턴스에 이 구성을 자동으로 사용합니다. 다음 로그 파일을 보면 지정된 파라미터가 인스턴스를 프로비저닝하는 데 사용되고 있는지 확인할 수 있습니다.

- `/var/log/parallelcluster/clustermgtd`(고정 용량용 - [MinCount](#) > 0)
- `/var/log/parallelcluster/slurm_resume.log`(동적 용량용)

파라미터가 정확하면 다음을 포함하는 로그 항목을 찾을 수 있습니다.

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

## RunInstances 재정의 업데이트

컴퓨팅 플릿을 중지하지 않고 언제든지 생성된 JSON 구성을 업데이트할 수 있습니다. 변경 사항이 적용된 후에는 모든 새 인스턴스가 업데이트된 구성으로 시작됩니다. 업데이트된 구성을 실행 중인 노드에 적용해야 하는 경우 인스턴스를 강제 종료하여 노드를 재활용하고 해당 노드가 교체될 때까지 기다리십시오 AWS ParallelCluster . Amazon EC2 콘솔에서 인스턴스를 AWS CLI 종료하거나 노드를 or 상태로 Slurm 설정하여 이 작업을 수행할 수 있습니다. DOWN DRAIN

다음 명령을 사용하여 Slurm 노드를 DOWN 또는 DRAIN으로 설정합니다.

```
$ scontrol update nodename=my-queue-dy-my-compute-resource-1 state=down
reason=your_reason
scontrol update nodename=my-queue-dy-my-compute-resource-1 state=drain
reason=your_reason
```

## 용량 블록 (CB) 으로 인스턴스 시작

AWS ParallelCluster 기계 [학습을 위한 온디맨드 용량 예약 \(ODCR\) 및 용량 블록 \(CB\)](#) 을 지원합니다. ODCR과 달리 CB는 향후 시작 시간을 가질 수 있으며 기한이 정해져 있습니다. ODCR 실행에 대한 자세한 내용은 [온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작](#) 을 참조하십시오.

## CB 사용: AWS ParallelCluster

CB를 사용하도록 새 클러스터 또는 기존 클러스터를 구성하려면 먼저 계정에 유효한 CB가 있어야 합니다. AWS 공식 설명서에 따라 AWS Management Console AWS Command Line Interface, 또는 SDK를 사용하여 사용 가능한 CB를 찾아 구매할 수 있습니다. 유효한 CB가 있으면 구성 파일에 CB Amazon 리소스 이름 (ARN) 및 관련 파라미터를 설정할 수 있습니다. AWS ParallelCluster 자세한 내용은 [용량 블록 \(CB\) 찾기 및 구매](#) 를 참조하십시오.

### 클러스터 구성의 CB

특정 대기열에 CB를 사용하려면 파라미터를 사용해야 합니다. CapacityReservationId 기존 CB ID로 구성하십시오. CB ARN은 CB를 생성할 AWS Management Console때 사용한 AWS CLI, 또는 SDK에서 가져올 수 있습니다.

CB를 CapacityType = CAPACITY\_BLOCK 사용할 대기열을 설정해야 합니다. 컴퓨팅 리소스 (InstanceTypeCB와 동일한 Amazon Elastic Compute Cloud 인스턴스 유형) 의 것으로 설정합니다.

컴퓨팅 리소스 수준에서 지정하는 CapacityReservationId 시기는 InstanceType 예약에서 자동으로 검색되므로 선택 사항입니다.

CB 예약의 CapacityType = CAPACITY\_BLOCK 일부인 모든 인스턴스가 정적 노드로 관리되므로 사용할 때는 0보다 크거나 MaxCount 같아야 합니다. MinCount

클러스터를 생성할 때 헤드 노드는 클러스터 생성 성공 신호를 보내기 전에 모든 정적 노드가 준비될 때까지 기다립니다. 하지만 사용 CapacityType = CAPACITY\_BLOCK 시 연결된 컴퓨팅 리소스의 일부인 노드는 이 검사 대상에서 제외됩니다. 구성된 모든 항목이 활성화되지 않은 경우에도 클러스터가 생성됩니다.

다음 구성 파일 스니펫은 AWS ParallelCluster 구성 파일에서 활성화하는 데 필요한 매개 변수를 보여줍니다.

```
SlurmQueues:
- Name: string
  CapacityType: CAPACITY_BLOCK
  InstanceType: String (EC2 Instance type of the CB)
  MinCount: integer (<= total capacity of the CB)
  MaxCount: integer (equal to MinCount)
  ComputeResources:
  - Name: string
    CapacityReservationTarget:
      CapacityReservationId: String (CB id)
```

## 용량 블록 (CB) AWS ParallelCluster 사용 방법

AWS ParallelCluster 관련된 정적 노드를 특이한 방식으로 관리합니다. AWS ParallelCluster CB가 아직 활성화되지 않은 경우에도 클러스터를 생성하고, CB가 활성화되면 인스턴스가 자동으로 시작됩니다.

컴퓨팅 리소스에 해당하는 Slurm 노드 중 아직 활성화되지 않은 노드는 CB 시작 시간에 도달할 때까지 유지 관리 상태로 유지됩니다. Slurm노드는 예약/유지 관리 상태를 유지하며 slurm 관리자 사용자와 연결됩니다. 즉, 작업을 수락할 수는 있지만 예약이 pending 제거될 때까지 작업은 그대로 유지됩니다.

AWS ParallelCluster Slurm예약을 자동으로 업데이트하고 관련 CB 노드를 유지 관리 상태로 전환합니다 (CB 상태에 따라 다름). CB가 활성화되면 Slurm 예약이 제거되고 노드가 시작되며 보류 중인 작업이나 새 작업 제출에 사용할 수 있게 됩니다.

CB 종료 시간에 도달하면 노드는 예약/유지 관리 상태로 다시 이동합니다. CB가 더 이상 활성화되지 않고 인스턴스가 종료된 경우 새 대기열/컴퓨팅 리소스에 작업을 다시 제출하거나 다시 등록하는 것은 사용자의 몫입니다.



## AMI 패치 적용 및 Amazon EC2 인스턴스 교체

동적으로 시작된 모든 클러스터 컴퓨팅 노드가 일관된 방식으로 작동하도록 하기 위해 클러스터 인스턴스 AWS ParallelCluster 자동 OS 업데이트를 비활성화합니다. 또한 각 버전 AWS ParallelCluster 및 관련 CLI에 대해 특정 AWS ParallelCluster AMI 세트가 구축됩니다. 이 특정 AMI 세트는 변경되지 않고 그대로 유지되며 빌드된 AWS ParallelCluster 버전에서만 지원됩니다. AWS ParallelCluster 출시된 버전의 AMI는 업데이트되지 않습니다.

그러나 새로운 보안 문제로 인해 고객은 이러한 AMI에 패치를 추가한 다음 패치된 AMI로 클러스터를 업데이트하기를 원할 수 있습니다. 이는 [AWS ParallelCluster 공동 책임 모델](#)과 일치합니다.

현재 사용 중인 CLI 버전에서 지원하는 특정 AWS ParallelCluster AWS ParallelCluster AMI 세트를 보려면 다음을 실행하십시오.

```
$ pcluster version
$ pcluster list-official-images
```

AWS ParallelCluster 헤드 노드는 정적 인스턴스이므로 수동으로 업데이트할 수 있습니다. 헤드 노드의 재시작 및 재부팅은 AWS ParallelCluster 버전 3.0.0부터 완전히 지원됩니다.

인스턴스에 임시 인스턴스 스토어가 있는 경우 수동 업데이트 전에 인스턴스 스토어 데이터를 저장해야 한다는 점을 기억해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [HeadNode/LocalStorage/EphemeralVolume](#) 클러스터 구성 및 [인스턴스 스토어 볼륨이 있는 인스턴스 유형](#)을 참조하세요.

컴퓨팅 노드는 휘발성 인스턴스입니다. 기본적으로 헤드 노드에서만 액세스할 수 있습니다. AWS ParallelCluster 버전 3.0.0부터 컴퓨팅 플릿을 중지한 후 [Scheduling//SlurmQueuesImage/CustomAmi](#) 파라미터를 수정하고 [pcluster update-cluster](#) 명령을 실행하여 컴퓨팅 인스턴스와 연결된 AMI를 업데이트할 수 있습니다. [pcluster update-compute-fleet](#)

```
$ pcluster update-compute-fleet-status --status STOP_REQUESTED
```

다음 방법 중 하나를 사용하여 컴퓨팅 노드에 대한 업데이트된 사용자 지정 AMI 생성을 자동화할 수 있습니다.

- 업데이트된 [Build/ParentImage](#)와 함께 [pcluster build-image](#) 명령을 사용하세요.
- [Build/UpdateOsPackages/Enabled](#):true로 빌드를 실행합니다.

## 헤드 노드 인스턴스 업데이트 또는 교체

경우에 따라 헤드 노드를 재시작하거나 재부팅해야 할 수 있습니다. 예를 들어, OS를 수동으로 업데이트하거나 헤드 노드 인스턴스를 다시 시작해야 하는 [AWS 예약된 인스턴스 사용 중지](#)가 있는 경우 이 설정이 필요합니다.

인스턴스에 휘발성 드라이브가 없는 경우 언제든지 이 인스턴스를 중지했다가 다시 시작할 수 있습니다. 사용 중지가 예정된 경우 중지된 인스턴스를 시작하면 새 하드웨어를 사용하도록 마이그레이션됩니다.

마찬가지로 인스턴스 스토어가 없는 인스턴스를 수동으로 중지하고 시작할 수 있습니다. 이 경우와 휘발성 볼륨이 없는 다른 인스턴스의 경우 [클러스터의 헤드 노드 중지 및 시작](#)로 가세요.

인스턴스에 휘발성 드라이브가 있고 중지된 경우 인스턴스 스토어의 데이터가 손실됩니다. 헤드 노드에 사용되는 인스턴스 유형에 인스턴스 스토어가 있는지 여부는 [인스턴스 스토어 볼륨](#)에 있는 표에서 확인할 수 있습니다.

### 휘발성 드라이브의 데이터를 저장합니다.

AWS ParallelCluster 버전 3.0.0부터 모든 인스턴스 유형에서 헤드 노드 재시작 및 재부팅이 완전히 지원됩니다. 하지만 인스턴스에 임시 드라이브가 있는 경우 데이터가 손실됩니다. 헤드 노드를 재시작하거나 재부팅하기 전에 다음 단계에 따라 데이터를 보존하세요.

보존해야 할 데이터가 있는지 확인하려면 [EphemeralVolume/MountDir](#) 폴더의 콘텐츠를 확인하세요(기본값 /scratch).

데이터를 루트 볼륨 또는 클러스터에 연결된 공유 스토리지 시스템(예: Amazon FSx, Amazon EFS 또는 Amazon EBS)으로 전송할 수 있습니다. 단, 원격 스토리지로 데이터를 전송할 때는 추가 비용이 발생할 수 있습니다.

데이터를 저장한 후 [클러스터의 헤드 노드 중지 및 시작](#)로 계속 진행하세요.

### 클러스터의 헤드 노드 중지 및 시작

1. 클러스터에 실행 중인 작업이 없는지 확인하세요.

Slurm 스케줄러를 사용하는 경우:

- `sbatch --no-requeue` 옵션이 지정되어 있지 않으면 실행 중인 작업이 다시 대기됩니다.
- `--no-requeue` 옵션이 지정되어 있으면 실행 중인 작업이 실패합니다.

## 2. 클러스터 컴퓨팅 플릿 중지 요청:

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOP_REQUESTED",
  ...
}
```

## 3. 컴퓨팅 플릿 상태가 STOPPED일 때까지 기다리세요.

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
  "status": "STOPPED",
  ...
}
```

## 4. OS 재부팅 또는 인스턴스 재시작을 통한 수동 업데이트의 경우 OR를 사용할 수 있습니다. AWS Management Console AWS CLI다음은 AWS CLI를 사용한 예입니다.

```
# Retrieve head node instance id
$ pcluster describe-cluster --cluster-name cluster-name --status STOP_REQUESTED
{
  "headNode": {
    "instanceId": "i-1234567890abcdef0",
    ...
  },
  ...
}
# stop and start the instance
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
  "StoppingInstances": [
    {
      "CurrentState": {
        "Name": "stopping"
        ...
      },
      "InstanceId": "i-1234567890abcdef0",
      "PreviousState": {
        "Name": "running"
        ...
      }
    }
  ]
}
```

```

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

```

5. 클러스터 컴퓨팅 플릿을 시작합니다.

```

$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
  "status": "START_REQUESTED",
  ...
}

```

## 운영 체제

AWS ParallelCluster 아마존 리눅스 2, 아마존 리눅스 2023, 센토스 7, 우분투 22.4, 우분투 2004, 레드햇 엔터프라이즈 리눅스 8 (RHEL8), 록키 8, 레드햇 엔터프라이즈 리눅스 9 (RHEL9), 록키 9를 지원합니다. AWS ParallelCluster 일부 운영 체제용으로 사전 구축된 AMI를 제공합니다. 제공된 AMI에 대한 자세한 내용은 [을 참조하십시오](#). AWS ParallelCluster [Image 섹션](#)

## 운영 체제 고려 사항

### 우분투 22.04

Ubuntu 2204에서는 ssh에 더 많은 보안 키가 필요하며 기본적으로 RSA 키를 지원하지 않습니다. ed25519 키를 생성하여 클러스터 생성에 사용하세요.

Ubuntu 2204는 해당 커널에 대한 Fsx 클라이언트가 없기 때문에 최신 커널로 업데이트할 수 없습니다.

## RHEL 8

RedHat 엔터프라이즈 리눅스 8.7 (rhel8) 은 버전 3.6.0부터 추가되었습니다. AWS ParallelCluster rhel8을 사용하도록 클러스터를 구성하는 경우 지원되는 다른 운영 체제를 사용하도록 클러스터를 구성할 때보다 모든 인스턴스 유형에 대한 온디맨드 비용이 더 높습니다.

요금에 대한 자세한 내용은 [온디맨드 요금](#) 및 [Amazon Elastic Compute Cloud의 Red Hat Enterprise Linux는 어떻게 제공되고 가격이 책정되나요?](#) 를 참조하십시오. .

## 록키 8

AWS ParallelCluster 3.8.0은 록키 리눅스 8을 지원하지만 사전 빌드된 록키 리눅스 8 AMI (x86 및 ARM 아키텍처용) 는 사용할 수 없습니다. AWS ParallelCluster 3.8.0은 속성을 사용하는 사용자 지정 AMI를 사용하여 Rocky Linux 8로 클러스터를 생성할 수 있도록 지원합니다. [CustomAmi](#) 사용자 지정 AMI 구축에 대한 자세한 내용은 을 참조하십시오. [AWS ParallelCluster AMI 사용자 지정](#)

[기본 Rocky Linux 8 AMI에서 사용자 지정 AMI를 구축하려면 마켓플레이스에서 사용할 수 있는 Rocky Linux 8 AMI를 구독하는 것을 고려할 수 있습니다.](#) AWS 마켓플레이스에서 AWS Rocky Linux 8 AMI의 가격 및 구독 비용을 검토하십시오. 또는 [공식 Rocky Linux 8 AMI를 기본 AMI로](#) 사용할 수도 있습니다.

## 센토스 7

[Gdrcopy](#)는 운영 체제 지원 매트릭스에서 센토스7을 제거했습니다. 즉, gdrcopy 2.3.1이 이 OS를 지원하는 최신 버전입니다. 최신 NVIDIA 오픈 소스 드라이버 버전 (OpenRM 즉, 535.129.03+) 은 이 버전의 gdrcopy와 호환되지 않으므로 Centos7용 NVIDIA 및 gdrcopy 버전을 고정해야 합니다. 3.8.0부터 당사의 공식 센토스7 AMI는 gdrcopy 2.3.1 및 엔비디아 드라이버 535.129.03과 함께 출시될 예정입니다.

## ParallelCluster

### 록키나인

AWS ParallelCluster 3.9.0은 록키 리눅스 9를 지원하지만 사전 빌드된 록키 리눅스 9 AMI (x86 및 ARM 아키텍처용) 는 사용할 수 없습니다. AWS ParallelCluster 3.9.0은 속성을 사용하는 사용자 지정 AMI를 사용하여 Rocky Linux 9에서 클러스터를 생성할 수 있도록 지원합니다. [CustomAmi](#) 사용자 지정 [AWS ParallelCluster AMI 구축에 대한 자세한 내용은 AMI 사용자 지정을](#) 참조하십시오. 기본 Rocky Linux 9 AMI에서 사용자 지정 AMI를 구축하려면 [공식 Rocky Linux 9 AMI를 기본 AMI로](#) 사용할 수도 있습니다. 기본 AMI에 최신 커널이 없는 경우 사용자 지정 Rocky Linux 9 AMI 빌드가 실패할 수 있습니다. AMI를 빌드하기 전에 커널을 업그레이드하려면

- [여기에서 rocky9 AMI ID를 사용하여 인스턴스를 시작하십시오.](https://rockylinux.org/cloud-images/)

- ssh로 인스턴스에 접속하고 다음 명령을 실행합니다. `sudo yum -y update`
- 인스턴스에서 다음과 같이 사용할 이미지를 생성합니다. ParentImage

# 에 대한 참조 AWS ParallelCluster

## 주제

- [AWS ParallelCluster CLI 명령](#)
- [구성 파일](#)
- [AWS ParallelCluster API 레퍼런스](#)
- [AWS ParallelCluster Python 라이브러리 API](#)

## AWS ParallelCluster CLI 명령

`pcluster`는 기본 AWS ParallelCluster CLI 명령입니다. `pcluster`를 사용하여 AWS 클라우드에서 HPC 클러스터를 시작 및 관리하고 사용자 지정 AMI 이미지를 생성하고 관리합니다.

`pcluster3-config-converter`는 AWS ParallelCluster 버전 2 형식의 클러스터 구성을 AWS ParallelCluster 버전 3 형식으로 변환하는 데 사용됩니다.

```
pcluster [-h] ( build-image | configure |
               create-cluster | dcv-connect |
               delete-cluster | delete-cluster-instances | delete-image |
               describe-cluster | describe-cluster-instances |
               describe-compute-fleet | describe-image |
               export-cluster-logs | export-image-logs |
               get-cluster-log-events | get-cluster-stack-events |
               get-image-log-events | get-image-stack-events |
               list-cluster-log-streams | list-clusters |
               list-images | list-image-log-streams | list-official-images |
               ssh | update-cluster |
               update-compute-fleet | version ) ...
pcluster3-config-converter [-h] [-t CLUSTER_TEMPLATE]
                           [-c CONFIG_FILE]
                           [--force-convert]
                           [-o OUTPUT_FILE]
```

## 주제

- [pcluster](#)
- [pcluster3-config-converter](#)

# pcluster

pcluster 기본 AWS ParallelCluster CLI 명령입니다. pcluster를 사용하여 AWS 클라우드에서 HPC 클러스터를 시작하고 관리할 수 있습니다.

pcluster은 명령 로그를 /home/user/.parallelcluster/ 내의 pcluster.log.# 파일에 기록합니다. 자세한 내용은 [pcluster CLI 로그](#) 항목을 참조하세요.

pcluster를 사용하려면 실행에 필요한 [권한](#)이 있는 IAM 역할이 있어야 합니다.

```
pcluster [-h]
```

## 인수

### pcluster *command*

가능한 선택: [build-image](#) [configure](#) [create-cluster](#) [dcv-connect](#) [delete-cluster](#) [delete-cluster-instances](#) [delete-image](#) [describe-cluster](#) [describe-cluster-instances](#) [describe-compute-fleet](#) [describe-image](#) [export-cluster-logs](#) [export-image-logs](#) [get-cluster-log-events](#) [get-cluster-stack-events](#) [get-image-log-events](#) [get-image-stack-events](#) [list-clusters](#) [list-cluster-log-streams](#) [list-images](#) [list-image-log-streams](#) [list-official-images](#) [ssh](#) [update-cluster](#) [update-compute-fleet](#) [version](#)

## 하위 명령:

### 주제

- [pcluster build-image](#)
- [pcluster configure](#)
- [pcluster create-cluster](#)
- [pcluster dcv-connect](#)
- [pcluster delete-cluster](#)
- [pcluster delete-cluster-instances](#)
- [pcluster delete-image](#)
- [pcluster describe-cluster](#)



- [pcluster describe-cluster-instances](#)
- [pcluster describe-compute-fleet](#)
- [pcluster describe-image](#)
- [pcluster export-cluster-logs](#)
- [pcluster export-image-logs](#)
- [pcluster get-cluster-log-events](#)
- [pcluster get-cluster-stack-events](#)
- [pcluster get-image-log-events](#)
- [pcluster get-image-stack-events](#)
- [pcluster list-clusters](#)
- [pcluster list-cluster-log-streams](#)
- [pcluster list-images](#)
- [pcluster list-image-log-streams](#)
- [pcluster list-official-images](#)
- [pcluster ssh](#)
- [pcluster update-cluster](#)
- [pcluster update-compute-fleet](#)
- [pcluster version](#)

## pcluster build-image

지정된 리전에서 커스텀 AWS ParallelCluster 이미지를 생성합니다.

```
pcluster build-image [-h]
                    --image-configuration IMAGE_CONFIGURATION
                    --image-id IMAGE_ID
                    [--debug]
                    [--dryrun DRYRUN]
                    [--query QUERY]
                    [--region REGION]
                    [--rollback-on-failure ROLLBACK_ON_FAILURE]
                    [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
                    [--validation-failure-level {INFO,WARNING,ERROR}]
```

이름 지정된 인수

### **-h, --help**

pcluster build-image에 대한 도움말 텍스트를 표시합니다.

### **--image-configuration, -c *IMAGE\_CONFIGURATION***

이미지 구성 파일을 YAML 문서로 지정합니다.

### **--image-id, -i *IMAGE\_ID***

빌드될 이미지의 ID를 지정합니다.

### **--debug**

디버깅 로깅을 활성화합니다.

### **--dryrun *DRYRUN***

true면 명령이 리소스를 만들지 않고 유효성 검사를 수행합니다. 이를 사용하여 이미지 구성을 검증할 수 있습니다. (false로 디폴트됩니다.)

### **--query *QUERY***

출력에서 수행할 JMESPath 쿼리입니다.

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

사용할 AWS 리전 를 지정합니다. 는 이미지 구성 파일의 [지역](#) 설정, AWS\_DEFAULT\_REGION 환경 변수, 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다. ~/.aws/config

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

true인 경우 실패 시 이미지 스택 롤백이 자동으로 시작됩니다. (기본값은 false입니다.)

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

억제할 구성 유효성 검사기를 하나 이상 식별하세요.

형식: (ALL|type:[A-Za-z0-9]+)

### **--validation-failure-level {INFO,WARNING,ERROR}**

생성 실패의 원인이 되는 최소 유효성 검사 수준을 지정합니다. (ERROR로 디폴트됩니다.)

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```
$ pcluster build-image --image-configuration image-config.yaml --image-id custom-  
alinux2-image  
{  
  "image": {  
    "imageId": "custom-alinux2-image",  
    "imageBuildStatus": "BUILD_IN_PROGRESS",  
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",  
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
custom-alinux2-image/1234abcd-56ef-78gh-90ij-abcd1234efgh",  
    "region": "us-east-1",  
    "version": "3.1.2"  
  }  
}
```

### ⚠ Warning

pcluster build-image는 기본 VPC를 사용합니다. AWS Control Tower 또는 AWS Landing Zone을 사용하여 기본 VPC를 삭제한 경우 이미지 구성 파일에 서브넷 ID를 지정해야 합니다. 자세한 내용은 [SubnetId](#)를 참조하세요.

## pcluster configure

AWS ParallelCluster 버전 3의 대화형 구성 마법사를 시작합니다. 자세한 정보는 [AWS ParallelCluster 명령줄 인터페이스를 사용하여 클러스터를 구성하고 생성합니다](#)을 참조하세요.

```
pcluster configure [-h]  
    --config CONFIG  
    [--debug]  
    [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster configure에 대한 도움말 텍스트를 표시합니다.

### --config *CONFIG*

생성된 구성 파일을 출력할 경로입니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--region, -r REGION**

사용할 AWS 리전 를 지정합니다. 이미지 구성 파일의 [리전](#) 설정, AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 파라미터를 사용하여 리전을 지정해야 합니다.

**pcluster create-cluster**

AWS ParallelCluster 클러스터를 생성합니다.

```
pcluster create-cluster [-h]
                        --cluster-configuration CLUSTER_CONFIGURATION
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--dryrun DRYRUN]
                        [--query QUERY]
                        [--region REGION]
                        [--rollback-on-failure ROLLBACK_ON_FAILURE]
                        [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]

                        [--validation-failure-level {INFO,WARNING,ERROR}]
```

이름 지정된 인수

**-h, --help**

pcluster create-cluster에 대한 도움말 텍스트를 표시합니다.

**--cluster-configuration, -c CLUSTER\_CONFIGURATION**

YAML 클러스터 구성 파일을 지정합니다.

**--cluster-name, -n CLUSTER\_NAME**

생성될 클러스터의 이름을 지정합니다.

이름은 영문자로 시작해야 합니다. 계획 이름은 최대 60자까지 가능합니다. Slurm 회계가 활성화된 경우 이름은 최대 40자까지 사용할 수 있습니다.

유효한 문자는 A-Z, a-z, 0-9 및 -(하이픈)입니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--dryrun *DRYRUN***

true면 명령이 리소스를 만들지 않고 유효성 검사를 수행합니다. 이를 사용하여 클러스터 구성을 검증할 수 있습니다. (false로 디폴트됩니다.)

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전 를 지정합니다. 는 클러스터 구성 파일의 [Region](#) 설정, AWS\_DEFAULT\_REGION 환경 변수, 파일 [default] 섹션의 region 설정 또는 --region 매개 변수를 사용하여 AWS 리전 지정해야 합니다. ~/.aws/config

**--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

true인 경우, 실패 시 클러스터 스택 롤백을 자동으로 시작합니다. (true로 디폴트됩니다.)

**--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

억제할 구성 유효성 검사기를 하나 이상 식별하세요.

형식: (ALL|유형:[A-Za-z0-9]+)

**--validation-failure-level {INFO,WARNING,ERROR}**

생성 실패의 원인이 되는 최소 유효성 검사 수준을 지정합니다. (ERROR로 디폴트됩니다.)

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster create-cluster -c cluster-config.yaml -n cluster-v3
{
  "cluster": {
    "clusterName": "cluster-v3",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
    "region": "us-east-1",
    "version": "3.1.4",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

```
}

```

## pcluster dcv-connect

NICE DCV를 사용하여 대화형 세션을 통해 헤드 노드에 연결할 수 있습니다.

```
pcluster dcv-connect [-h]
                    --cluster-name CLUSTER_NAME
                    [--debug]
                    [--key-path KEY_PATH]
                    [--region REGION]
                    [--show-url]
```

이름 지정된 인수

### -h, --help

pcluster dcv-connect에 대한 도움말 텍스트를 표시합니다.

### --cluster-name, -n *CLUSTER\_NAME*

클러스터의 이름을 지정합니다.

### --debug

디버깅 로깅을 활성화합니다.

### --key-path *KEY\_PATH*

연결에 사용할 SSH 키의 경로입니다.

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

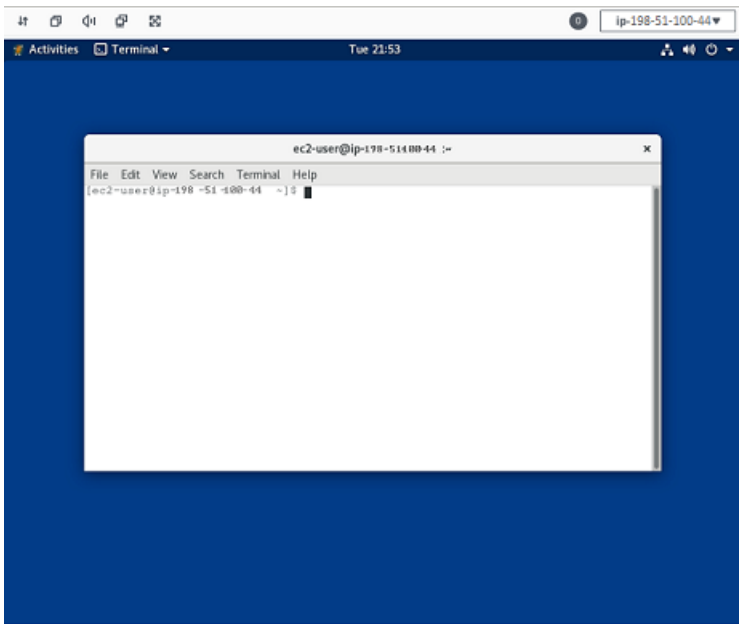
사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

### --show-url

DCV 연결에 사용할 URL을 인쇄하고 종료합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster dcv-connect -n cluster-3Dcv -r us-east-1 --key-path /home/user/.ssh/key.pem
```



## pcluster delete-cluster

클러스터 삭제를 시작합니다.

```
pcluster delete-cluster [-h]
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster delete-cluster에 대한 도움말 텍스트를 표시합니다.

### --cluster-name, -n *CLUSTER\_NAME*

클러스터의 이름을 지정합니다.

### --debug

디버깅 로깅을 활성화합니다.

### --query *QUERY*

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 리전은 `AWS_DEFAULT_REGION` 환경 변수, `~/.aws/config` 파일 [default] 섹션의 `region` 설정 또는 `--region` 파라미터를 사용하여 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster delete-cluster -n cluster-v3
{
  "cluster": {
    "clusterName": "cluster-v3",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
    "region": "us-east-1",
    "version": "3.1.4",
    "clusterStatus": "DELETE_IN_PROGRESS"
  }
}
```

**pcluster delete-cluster-instances**

모든 클러스터 컴퓨팅 노드의 강제 종료를 시작합니다. AWS Batch 클러스터에서는 작동하지 않습니다.

```
pcluster delete-cluster-instances [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--force FORCE]
    [--query QUERY]
    [--region REGION]
```

이름 지정된 인수

**-h, --help**

`pcluster delete-cluster-instances`에 대한 도움말 텍스트를 표시합니다.

**--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.



**--debug**

디버깅 로깅을 활성화합니다.

**--force *FORCE***

true면 유효성 검사 오류를 무시하여 강제로 삭제합니다. (false로 디폴트됩니다.)

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

```
$ pcluster delete-cluster-instances -n cluster-v3
```

**pcluster delete-image**

사용자 지정 AWS ParallelCluster 이미지 삭제를 시작합니다.

```
pcluster delete-image [-h]
    --image-id IMAGE_ID
    [--debug]
    [--force FORCE]
    [--query QUERY]
    [--region REGION]
```

이름 지정된 인수

**-h, --help**

pcluster delete-image에 대한 도움말 텍스트를 표시합니다.

**--image-id, -i *IMAGE\_ID***

삭제될 이미지의 ID를 지정합니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--force *FORCE***

true면 AMI를 사용하는 인스턴스가 있거나 AMI를 공유하는 경우 강제로 삭제합니다. (false로 디폴트됩니다.)

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전을 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster delete-image --image-id custom-alinux2-image
{
  "image": {
    "imageId": "custom-alinux2-image",
    "imageBuildStatus": "DELETE_IN_PROGRESS",
    "region": "us-east-1",
    "version": "3.1.4"
  }
}
```

**pcluster describe-cluster**

클러스터에 관한 자세한 정보를 가져옵니다.

```
pcluster describe-cluster [-h]
                        --cluster-name CLUSTER_NAME
                        [--debug]
                        [--query QUERY]
                        [--region REGION]
```

이름 지정된 인수

**-h, --help**

pcluster describe-cluster에 대한 도움말 텍스트를 표시합니다.

**--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 이는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전을 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

클러스터 세부 정보 설명:

```
$ pcluster describe-cluster -n cluster-v3
{
  "creationTime": "2022-07-12T17:19:16.101Z",
  "headNode": {
    "launchTime": "2022-07-12T17:22:21.000Z",
    "instanceId": "i-1234567890abcdef0",
    "publicIpAddress": "198.51.100.44",
    "instanceType": "t2.micro",
    "state": "running",
    "privateIpAddress": "192.0.2.0.196"
  },
  "loginNodes": {
    "status": "active",
    "address": "8af2145440569xyz.us-east-1.amazonaws.com",
    "scheme": "internet-facing|internal",
    "healthyNodes": 3,
    "unhealthyNodes": 0
  },
  "version": "3.1.4",
  "clusterConfiguration": {
    "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
  },
}
```

```

"tags": [
  {
    "value": "3.1.4",
    "key": "parallelcluster:version"
  }
],
"cloudFormationStackStatus": "CREATE_COMPLETE",
"clusterName": "cluster-v3",
"computeFleetStatus": "RUNNING",
"cloudFormationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
"lastUpdatedTime": "2022-07-12T17:19:16.101Z",
"region": "us-east-1",
"clusterStatus": "CREATE_COMPLETE"
}

```

describe-cluster를 사용하여 클러스터 구성을 검색합니다.

```

$ curl -o - - $(pcluster describe-cluster -n cluster-v3 --query clusterConfiguration.url
| xargs echo)
Region: us-east-1
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: adpc
  Iam:
    S3Access:
      - BucketName: cluster-v3-bucket
        KeyName: logs
        EnableWriteAccess: true
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue1
        ComputeResources:
          - Name: t2micro
            InstanceType: t2.micro
            MinCount: 0
            MaxCount: 10

```

```
Networking:
  SubnetIds:
    - subnet-021345abcdef6789
```

## pcluster describe-cluster-instances

클러스터의 인스턴스를 설명하세요.

```
pcluster describe-cluster-instances [-h]
  --cluster-name CLUSTER_NAME
  [--debug]
  [--next-token NEXT_TOKEN]
  [--node-type {HeadNode,ComputeNode,LoginNode}]
  [--query QUERY]
  [--queue-name QUEUE_NAME]
  [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster describe-cluster-instances에 대한 도움말 텍스트를 표시합니다.

### --cluster-name, -n *CLUSTER\_NAME*

클러스터의 이름을 지정합니다.

### --debug

디버깅 로깅을 활성화합니다.

### --next-token *NEXT\_TOKEN*

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

### --node-type {HeadNode,ComputeNode,LoginNode}

나열할 노드 유형을 지정합니다. 지원되는 값은 HeadNode, ComputeNode 및 LoginNode입니다. 이 파라미터를 지정하지 않으면 HeadNode, ComputeNode 및 LoginNode 인스턴스가 설명됩니다.

### --query *QUERY*

출력에서 수행할 JMESPath 쿼리를 지정합니다.

**--queue-name** *QUEUE\_NAME*

나열할 대기열의 이름을 지정합니다. 이 파라미터를 지정하지 않으면 모든 대기열의 인스턴스가 설명됩니다.

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

사용할 AWS 리전을 지정합니다. 이는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전을 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster describe-cluster-instances -n cluster-v3
{
  "instances": [
    {
      "launchTime": "2022-07-12T17:22:21.000Z",
      "instanceId": "i-1234567890abcdef0",
      "publicIpAddress": "198.51.100.44",
      "instanceType": "t2.micro",
      "state": "running",
      "nodeType": "HeadNode",
      "privateIpAddress": "192.0.2.0.196"
    },
    {
      "launchTime": "2022-07-12T17:37:42.000Z",
      "instanceId": "i-021345abcdef6789",
      "queueName": "queue1",
      "publicIpAddress": "198.51.100.44",
      "instanceType": "t2.micro",
      "state": "pending",
      "nodeType": "ComputeNode",
      "privateIpAddress": "192.0.2.0.196"
    }
  ]
}
```

**pcluster describe-compute-fleet**

컴퓨팅 플릿의 상태를 설명합니다.

```
pcluster describe-compute-fleet [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--query QUERY]
    [--region REGION]
```

이름 지정된 인수

### **-h, --help**

pcluster describe-compute-fleet에 대한 도움말 텍스트를 표시합니다.

### **--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

### **--debug**

디버깅 로깅을 활성화합니다.

### **--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전 를 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
  "status": "RUNNING",
  "lastStatusUpdatedTime": "2022-07-12T17:24:26.000Z"
}
```

## **pcluster describe-image**

이미지에 대한 세부 정보를 가져옵니다.

```
pcluster describe-image [-h]
```

```

--image-id IMAGE_ID
[--debug]
[--query QUERY]
[--region REGION]

```

이름 지정된 인수

### **-h, --help**

pcluster describe-image에 대한 도움말 텍스트를 표시합니다.

### **--image-id, -i *IMAGE\_ID***

이미지의 ID를 지정합니다.

### **--debug**

디버깅 로깅을 활성화합니다.

### **--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```

$ pcluster describe-image --image-id custom-alinux2-image
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678-v1-do-not-delete.../configs/image-
config.yaml"
  },
  "imageId": "custom-alinux2-image",
  "creationTime": "2022-04-05T20:23:07.000Z"
  "imageBuildStatus": "BUILD_COMPLETE",
  "region": "us-east-1",
  "ec2AmiInfo": {
    "amiName": "custom-alinux2-image 2022-04-05T19-55-22.518Z",
    "amiId": "ami-1234abcd5678efgh",

```



```

    "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.268-205.500.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.14.2-1.amzn2.x86_64, dcv-2021.3.11591-1.el7.x86_64, slurm-21-08-6-1",
    "state": "AVAILABLE",
    "tags": [
      {
        "value": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
        "key": "Ec2ImageBuilderArn"
      },
      {
        "value": "parallelcluster-1234abcd5678efgh-v1-do-not-delete",
        "key": "parallelcluster:s3_bucket"
      },
      {
        "value": "custom-alinux2-image",
        "key": "parallelcluster:image_name"
      },
      {
        "value": "available",
        "key": "parallelcluster:build_status"
      },
      {
        "value": "s3://parallelcluster-1234abcd5678efgh-v1-do-not-delete/
parallelcluster/3.1.2/images/custom-alinux2-image-1234abcd5678efgh/configs/image-
config.yaml",
        "key": "parallelcluster:build_config"
      },
      {
        "value": "Amazon EC2 Image Builder",
        "key": "CreatedBy"
      },
      {
        "value": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-alinux2-image",
        "key": "parallelcluster:build_log"
      },
      {
        "value": "4.14.268-205.500.amzn2.x86_64",
        "key": "parallelcluster:kernel_version"
      },
      {
        "value": "arn:aws:imagebuilder:us-east-1:444455556666:image/amazon-linux-2-
x86/2022.3.16/1",

```

```
    "key": "parallelcluster:parent_image"
  },
  {
    "value": "3.1.2",
    "key": "parallelcluster:version"
  },
  {
    "value": "0.5.14",
    "key": "parallelcluster:munge_version"
  },
  {
    "value": "21-08-6-1",
    "key": "parallelcluster:slurm_version"
  },
  {
    "value": "2021.3.11591-1.el7.x86_64",
    "key": "parallelcluster:dcv_version"
  },
  {
    "value": "alinux2-image",
    "key": "parallelcluster:image_id"
  },
  {
    "value": "3.2.3",
    "key": "parallelcluster:pmix_version"
  },
  {
    "value": "parallelcluster/3.7.0/images/alinux2-image-abcd1234efgh56781234",
    "key": "parallelcluster:s3_image_dir"
  },
  {
    "value": "1.14.2-1.amzn2.x86_64",
    "key": "parallelcluster:efa_version"
  },
  {
    "value": "alinux2",
    "key": "parallelcluster:os"
  },
  {
    "value": "aws-parallelcluster-cookbook-3.1.2",
    "key": "parallelcluster:bootstrap_file"
  },
  {
    "value": "1.8.23-10.amzn2.1.x86_64",
```

```

    "key": "parallelcluster:sudo_version"
  },
  {
    "value": "2.10.8-5.amzn2.x86_64",
    "key": "parallelcluster:lustre_version"
  }
],
"architecture": "x86_64"
},
"version": "3.1.2"
}

```

## pcluster export-cluster-logs

Amazon S3 버킷을 통과하여 클러스터의 로그를 로컬 tar.gz 아카이브로 내보냅니다.

```

pcluster export-cluster-logs [-h]
    --bucket BUCKET_NAME
    --cluster-name CLUSTER_NAME
    [--bucket-prefix BUCKET_PREFIX]
    [--debug]
    [--end-time END_TIME]
    [--filters FILTER [FILTER ...]]
    [--keep-s3-objects KEEP_S3_OBJECTS]
    [--output-file OUTPUT_FILE]
    [--region REGION]
    [--start-time START_TIME]

```

이름 지정된 인수

### -h, --help

pcluster export-cluster-logs에 대한 도움말 텍스트를 표시합니다.

### --bucket *BUCKET\_NAME*

클러스터 로그 데이터를 내보낼 Amazon S3 버킷의 이름을 지정합니다. 그것은 클러스터와 같은 리전에 있어야 합니다.

**Note**

CloudWatch 액세스를 허용하려면 Amazon S3 버킷 정책에 권한을 추가해야 합니다. 자세한 내용은 CloudWatch Logs 사용 설명서의 [Amazon S3 버킷에 대한 권한 설정을](#) 참조하십시오.

**--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

**--bucket-prefix *BUCKET\_PREFIX***

내보낸 로그 데이터가 저장되는 Amazon S3 버킷의 경로를 지정합니다.

기본적으로 버킷 접두사는 다음과 같습니다.

```
cluster-name-logs-202209061743.tar.gz
```

*202209061743*는 현재 시간을 %Y%m%d%H%M 형식으로 나타낸 것입니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--end-time *END\_TIME***

로그 이벤트를 수집할 시간 범위의 끝을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다. 시간 요소(예: 분 및 초)는 생략될 수 있습니다. 기본값은 현재 시간입니다.

**--filters *FILTER* [*FILTER* ...]**

로그의 필터를 지정합니다. 형식: Name=a,Values=1 Name=b,Values=2,3 지원되는 필터는 다음과 같습니다.

private-dns-name

인스턴스의 프라이빗 DNS 이름의 약식을 지정합니다(예:ip-10-0-0-101).

node-type

노드 유형을 지정합니다. 이 필터에 허용되는 유일한 값은 HeadNode입니다.

**--keep-s3-objects** *KEEP\_S3\_OBJECTS*

true면 내보낸 객체를 아마존 S3로 내보낸 것을 보관합니다. (false로 디폴트됩니다.)

**--output-file** *OUTPUT\_FILE*

로그 아카이브를 저장할 파일 경로를 지정합니다. 이 정보가 제공되면 로그가 로컬에 저장됩니다. 그렇지 않으면 출력에 반환된 URL과 함께 Amazon S3에 업로드됩니다. Amazon S3로 업로드하는 것이 기본값입니다.

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

사용할 AWS 리전 항목을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

**--start-time** *START\_TIME*

시간 범위의 시작을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 로그 이벤트가 포함됩니다. 지정하지 않을 경우 기본값은 클러스터가 생성된 시간입니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
{
  "url": "https://cluster-v3-bucket..."
}
```

**pcluster export-image-logs**

Amazon S3 버킷을 통과하여 이미지 빌더 스택의 로그를 로컬 tar.gz 아카이브로 내보냅니다.

```
pcluster export-image-logs [-h]
    --bucket BUCKET
    --image-id IMAGE_ID
    [--bucket-prefix BUCKET_PREFIX]
    [--debug]
    [--end-time END_TIME]
    [--keep-s3-objects KEEP_S3_OBJECTS]
    [--output-file OUTPUT_FILE]
    [--region REGION]
```

```
[--start-time START_TIME]
```

이름 지정된 인수

### **-h, --help**

pcluster export-image-logs에 대한 도움말 텍스트를 표시합니다.

### **--bucket *BUCKET\_NAME***

이미지 빌드 로그를 내보낼 Amazon S3 버킷 이름을 지정합니다. 그것은 이미지와 같은 리전에 있어야 합니다.

#### Note

CloudWatch 액세스를 허용하려면 Amazon S3 버킷 정책에 권한을 추가해야 합니다. 자세한 내용은 CloudWatch Logs 사용 설명서의 [Amazon S3 버킷에 대한 권한 설정을 참조하십시오](#).

### **--image-id, -i *IMAGE\_ID***

로그를 내보낼 이미지 ID입니다.

### **--bucket-prefix *BUCKET\_PREFIX***

내보낸 로그 데이터가 저장되는 Amazon S3 버킷의 경로를 지정합니다.

기본적으로 버킷 접두사는 다음과 같습니다.

```
ami-id-logs-202209061743.tar.gz
```

*202209061743*는 현재 시간을 %Y%m%d%H%M 형식으로 나타낸 것입니다.

### **--debug**

디버깅 로깅을 활성화합니다.

### **--end-time *END\_TIME***

로그 이벤트를 수집할 시간 범위의 끝을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다. 시간 요소(예: 분 및 초)는 생략될 수 있습니다. 기본값은 현재 시간입니다.

**--keep-s3-objects** *KEEP\_S3\_OBJECTS*

true면 내보낸 객체를 아마존 S3로 내보낸 것을 보관합니다. (false로 디폴트됩니다.)

**--output-file** *OUTPUT\_FILE*

로그 아카이브를 저장할 파일 경로를 지정합니다. 이 정보가 제공되면 로그가 로컬에 저장됩니다. 그렇지 않으면 출력에 반환된 URL과 함께 Amazon S3에 업로드됩니다. Amazon S3로 업로드하는 것이 기본값입니다.

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

사용할 AWS 리전 항목을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

**--start-time** *START\_TIME*

시간 범위의 시작을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 로그 이벤트가 포함됩니다. 지정하지 않을 경우 기본값은 클러스터가 생성된 시간입니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster export-image-logs --bucket image-v3-bucket --image-id ami-1234abcd5678efgh
{
  "url": "https://image-v3-bucket..."
}
```

**pcluster get-cluster-log-events**

로그 스트림과 관련된 이벤트를 검색합니다.

```
pcluster get-cluster-log-events [-h]
    --cluster-name CLUSTER_NAME
    --log-stream-name LOG_STREAM_NAME
    [--debug]
    [--end-time END_TIME]
    [--limit LIMIT]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

```
[--start-from-head START_FROM_HEAD]  
[--start-time START_TIME]
```

이름 지정된 인수

## **-h, --help**

pcluster get-cluster-log-events에 대한 도움말 텍스트를 표시합니다.

## **--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

## **--log-stream-name *LOG\_STREAM\_NAME***

로그 스트림의 이름을 지정합니다. list-cluster-log-streams 명령을 사용하여 이벤트 또는 이벤트와 관련된 로그 스트림을 검색할 수 있습니다.

## **--debug**

디버깅 로깅을 활성화합니다.

## **--end-time *END\_TIME***

시간 범위의 끝을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다.

## **--limit *LIMIT***

반환된 로그 이벤트의 최대 수를 지정합니다. 값이 지정되지 않는 경우 최대값은 1MB의 응답 크기에 들어갈 수 있는 로그 이벤트 수(최대 10,000개의 로그 이벤트)입니다.

## **--next-token *NEXT\_TOKEN***

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

## **--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.



**--start-from-head *START\_FROM\_HEAD***

값이 true인 경우 가장 이른 로그 이벤트가 먼저 반환됩니다. 값이 false인 경우 가장 최근의 로그 이벤트가 먼저 반환됩니다. (false로 디폴트됩니다.)

**--start-time *START\_TIME***

시간 범위의 시작을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트가 포함됩니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster get-cluster-log-events \
  -c cluster-v3 \
  -r us-east-1 \
  --log-stream-name ip-198-51-100-44.i-1234567890abcdef0.clustermgtd \
  --limit 3
{
  "nextToken": "f/36966906399261933213029082268132291405859205452101451780/s",
  "prevToken": "b/36966906399239632467830551644990755687586557090595471362/s",
  "events": [
    {
      "message": "2022-07-12 19:16:53,379 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Performing node maintenance actions",
      "timestamp": "2022-07-12T19:16:53.379Z"
    },
    {
      "message": "2022-07-12 19:16:53,380 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Following nodes are currently in replacement: (x0) []",
      "timestamp": "2022-07-12T19:16:53.380Z"
    },
    {
      "message": "2022-07-12 19:16:53,380 -
[slurm_plugin.clustermgtd:_terminate_orphaned_instances] - INFO - Checking for
orphaned instance",
      "timestamp": "2022-07-12T19:16:53.380Z"
    }
  ]
}
```

**pcluster get-cluster-stack-events**

지정된 클러스터의 스택과 관련된 이벤트를 검색합니다.

**Note**

버전 3.6.0부터 증첩된 스택을 AWS ParallelCluster 사용하여 대기열 및 컴퓨팅 리소스와 관련된 리소스를 생성합니다. GetClusterStackEvents API와 `pcluster get-cluster-stack-events` 명령은 클러스터 메인 스택 이벤트만 반환합니다. 콘솔에서 대기열 및 컴퓨팅 리소스와 관련된 이벤트를 비롯한 클러스터 스택 이벤트를 볼 수 있습니다. CloudFormation

```
pcluster get-cluster-stack-events [-h]
    --cluster-name CLUSTER_NAME
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

이름 지정된 인수

**-h, --help**

`pcluster get-cluster-stack-events`에 대한 도움말 텍스트를 표시합니다.

**--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--next-token *NEXT\_TOKEN***

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전 항목을 지정합니다. 는 `AWS_DEFAULT_REGION` 환경 변수, `~/.aws/config` 파일 [default] 섹션의 `region` 설정 또는 `--region` 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster get-cluster-stack-events \
  -n cluster-v3 \
  -r us-east-1 \
  --query "events[0]"
{
  "eventId": "1234abcd-56ef-78gh-90ij-abcd1234efgh",
  "physicalResourceId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
  "resourceStatus": "CREATE_COMPLETE",
  "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
  "stackName": "cluster-v3",
  "logicalResourceId": "cluster-v3",
  "resourceType": "AWS::CloudFormation::Stack",
  "timestamp": "2022-07-12T18:29:12.140Z"
}
```

## pcluster get-image-log-events

이미지 빌드와 관련된 이벤트를 검색합니다.

```
pcluster get-image-log-events [-h]
  --image-id IMAGE_ID
  --log-stream-name LOG_STREAM_NAME
  [--debug]
  [--end-time END_TIME]
  [--limit LIMIT]
  [--next-token NEXT_TOKEN]
  [--query QUERY]
  [--region REGION]
  [--start-from-head START_FROM_HEAD]
  [--start-time START_TIME]
```

이름 지정된 인수

### -h, --help

pcluster get-image-log-events에 대한 도움말 텍스트를 표시합니다.

### --image-id, -i **IMAGE\_ID**

이미지의 ID를 지정합니다.

**--log-stream-name** *LOG\_STREAM\_NAME*

로그 스트림의 이름을 지정합니다. `list-image-log-streams` 명령을 사용하여 이벤트 또는 이벤트와 관련된 로그 스트림을 검색할 수 있습니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--end-time** *END\_TIME*

시간 범위의 끝을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다.

**--limit** *LIMIT*

반환된 로그 이벤트의 최대 수를 지정합니다. 값이 지정되지 않는 경우 최대값은 1MB의 응답 크기에 들어갈 수 있는 로그 이벤트 수(최대 10,000개의 로그 이벤트)입니다.

**--next-token** *NEXT\_TOKEN*

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

**--query** *QUERY*

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 `AWS_DEFAULT_REGION` 환경 변수, `~/.aws/config` 파일 [default] 섹션의 `region` 설정 또는 `--region` 매개변수를 사용하여 AWS 리전 지정해야 합니다.

**--start-from-head** *START\_FROM\_HEAD*

값이 `true`인 경우 가장 이른 로그 이벤트가 먼저 반환됩니다. 값이 `false`인 경우 가장 최근의 로그 이벤트가 먼저 반환됩니다. (`false`로 디폴트됩니다.)

**--start-time** *START\_TIME*

시간 범위의 시작을 지정합니다. ISO 8601 형식(YYYY-MM-DDThh:mm:ssZ, 예시로는 2021-01-01T20:00:00Z)으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트가 포함됩니다.

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```
$ pcluster get-image-log-events --image-id custom-alinux2-image --region us-east-1 --
log-stream-name 3.1.2/1 --limit 3
{
  "nextToken": "f/36778317771100849897800729464621464113270312017760944178/s",
  "prevToken": "b/36778317766952911290874033560295820514557716777648586800/s",
  "events": [
    {
      "message": "ExecuteBash: FINISHED EXECUTION",
      "timestamp": "2022-04-05T22:13:26.633Z"
    },
    {
      "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-1234abcd-56ef-78gh-90ij-abcd1234efgh/3.1.2/1",
      "timestamp": "2022-04-05T22:13:26.741Z"
    },
    {
      "message": "TOE has completed execution successfully",
      "timestamp": "2022-04-05T22:13:26.819Z"
    }
  ]
}
```

## pcluster get-image-stack-events

지정된 이미지 빌드의 스택과 관련된 이벤트를 검색합니다.

```
pcluster get-image-stack-events [-h]
    --image-id IMAGE_ID
    [--debug]
    [--next-token NEXT_TOKEN]
    [--query QUERY]
    [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster get-image-stack-events에 대한 도움말 텍스트를 표시합니다.

### --image-id, -i *IMAGE\_ID*

이미지의 ID를 지정합니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--next-token *NEXT\_TOKEN***

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```
$ pcluster get-image-stack-events --image-id custom-alinux2-image --region us-east-1 --
query "events[0]"
{
  "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
  "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
  "resourceStatus": "CREATE_IN_PROGRESS",
  "resourceStatusReason": "Resource creation Initiated",
  "resourceProperties": "{\"InfrastructureConfigurationArn\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\", \\\"ImageRecipeArn
\\\": \\\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-custom-alinux2-image/3.1.2\\\", \\\"DistributionConfigurationArn
\\\": \\\"arn:aws:imagebuilder:us-east-1:123456789012:distribution-
configuration/parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\",
\\\"EnhancedImageMetadataEnabled\\\": \\\"false\\\", \\\"Tags\\\": {\\\"parallelcluster:image_name\\\":
\\\"custom-alinux2-image\\\", \\\"parallelcluster:image_id\\\": \\\"custom-alinux2-image\\\"}}\",
  "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/custom-alinux2-
image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
  "stackName": "custom-alinux2-image",
  "logicalResourceId": "ParallelClusterImage",
  "resourceType": "AWS::ImageBuilder::Image",
  "timestamp": "2022-04-05T21:39:24.725Z"
}
```

## pcluster list-clusters

기존 클러스터의 목록을 가져옵니다.

```
pcluster list-clusters [-h]
                        [--cluster-status {CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
                        DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
                        UPDATE_COMPLETE,UPDATE_FAILED}]
                        [{CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
                        DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
                        UPDATE_COMPLETE,UPDATE_FAILED} ...]]
                        [--debug]
                        [--next-token NEXT_TOKEN]
                        [--query QUERY]
                        [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster list-clusters에 대한 도움말 텍스트를 표시합니다.

```
--cluster-status {CREATE_IN_PROGRESS, CREATE_FAILED, CREATE_COMPLETE,
DELETE_IN_PROGRESS, DELETE_FAILED, UPDATE_IN_PROGRESS, UPDATE_COMPLETE,
UPDATE_FAILED} [{CREATE_IN_PROGRESS, CREATE_FAILED, CREATE_COMPLETE,
DELETE_IN_PROGRESS, DELETE_FAILED, UPDATE_IN_PROGRESS, UPDATE_COMPLETE,
UPDATE_FAILED} ...]
```

필터링할 클러스터 상태 목록을 지정합니다. (all로 디폴트됩니다.)

### --debug

디버깅 로깅을 활성화합니다.

### --next-token *NEXT\_TOKEN*

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

### --query *QUERY*

출력에서 수행할 JMESPath 쿼리를 지정합니다.

**--region, -r REGION**

사용할 AWS 리전을 지정합니다. 이는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전을 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster list-clusters
{
  "clusters": [
    {
      "clusterName": "cluster-v3",
      "cloudformationStackStatus": "CREATE_COMPLETE",
      "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
      "region": "us-east-1",
      "version": "3.1.4",
      "clusterStatus": "CREATE_COMPLETE"
    }
  ]
}
```

**pcluster list-cluster-log-streams**

클러스터와 관련된 로그 스트림의 목록을 검색합니다.

```
pcluster list-cluster-log-streams [-h]
    --cluster-name CLUSTER_NAME
    [--filters FILTERS [FILTERS ...]]
    [--next-token NEXT_TOKEN] [--debug]
    [--query QUERY]
    [--region REGION]
```

이름 지정된 인수

**-h, --help**

pcluster list-cluster-log-streams에 대한 도움말 텍스트를 표시합니다.



**--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--filters *FILTERS* [*FILTERS* ...]**

로그 스트림의 필터를 지정합니다. 형식: Name=a,Values=1 Name=b,Values=2,3 지원되는 필터는 다음과 같습니다.

private-dns-name

인스턴스의 프라이빗 DNS 이름의 약식을 지정합니다(예:ip-10-0-0-101).

node-type

노드 유형을 지정합니다. 이 필터에 허용되는 유일한 값은 HeadNode입니다.

**--next-token *NEXT\_TOKEN***

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster list-cluster-log-streams \
  -n cluster-v3 \
  -r us-east-1 \
  --query 'logStreams[*].logStreamName'
[
  "ip-172-31-58-205.i-1234567890abcdef0.cfn-init",
  "ip-172-31-58-205.i-1234567890abcdef0.chef-client",
  "ip-172-31-58-205.i-1234567890abcdef0.cloud-init",
```

```
"ip-172-31-58-205.i-1234567890abcdef0.clustermgtd",
"ip-172-31-58-205.i-1234567890abcdef0.slurmctld",
"ip-172-31-58-205.i-1234567890abcdef0.supervisord",
"ip-172-31-58-205.i-1234567890abcdef0.system-messages"
]
```

## pcluster list-images

기존 사용자 지정 이미지 목록을 검색합니다.

```
pcluster list-images [-h]
                    --image-status {AVAILABLE,PENDING,FAILED}
                    [--debug]
                    [--next-token NEXT_TOKEN]
                    [--query QUERY]
                    [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster list-images에 대한 도움말 텍스트를 표시합니다.

### --image-status {AVAILABLE,PENDING,FAILED}

제공된 상태를 기준으로 반환된 이미지를 필터링합니다.

### --debug

디버깅 로깅을 활성화합니다.

### --next-token *NEXT\_TOKEN*

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

### --query *QUERY*

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전 를 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```
$ pcluster list-images --image-status AVAILABLE
{
  "images": [
    {
      "imageId": "custom-alinux2-image",
      "imageBuildStatus": "BUILD_COMPLETE",
      "ec2AmiInfo": {
        "amiId": "ami-1234abcd5678efgh"
      },
      "region": "us-east-1",
      "version": "3.1.2"
    }
  ]
}
```

## pcluster list-image-log-streams

이미지와 관련된 로그 스트림 목록을 검색합니다.

```
pcluster list-image-log-streams [-h]
                                --image-id IMAGE_ID
                                [--next-token NEXT_TOKEN] [--debug]
                                [--query QUERY]
                                [--region REGION]
```

이름 지정된 인수

### -h, --help

pcluster list-image-log-streams에 대한 도움말 텍스트를 표시합니다.

### --image-id, -i **IMAGE\_ID**

이미지의 ID를 지정합니다.

### --debug

디버깅 로깅을 활성화합니다.

### --next-token **NEXT\_TOKEN**

페이지가 매겨진 요청에 사용할 토큰을 지정합니다.

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전을 지정합니다. 이는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전을 지정해야 합니다.

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```
$ pcluster list-image-log-streams --image-id custom-alinux2-image --region us-east-1 --
query 'LogStreams[*].LogStreamName'
[
  "3.0.0/1",
  "3.1.2/1"
]
```

**pcluster list-official-images**

공식 AWS ParallelCluster AMI를 설명해 주세요.

```
pcluster list-official-images [-h]
                        [--architecture ARCHITECTURE]
                        [--debug]
                        [--os OS]
                        [--query QUERY]
                        [--region REGION]
```

이름 지정된 인수

**-h, --help**

pcluster list-official-images에 대한 도움말 텍스트를 표시합니다.

**--architecture *ARCHITECTURE***

결과를 필터링하는 데 사용할 아키텍처를 지정합니다. 이 파라미터를 지정하지 않으면 모든 아키텍처가 반환됩니다.

**--debug**

디버깅 로깅을 활성화합니다.

**--os *OS***

결과를 필터링하는 데 사용할 운영 체제를 지정합니다. 이 파라미터를 지정하지 않으면 모든 운영 체제가 반환됩니다.

**--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전 항목을 지정합니다. 는 이미지 구성 파일의 [지역](#) 설정, AWS\_DEFAULT\_REGION 환경 변수, 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다. ~/.aws/config

AWS ParallelCluster 버전 3.1.2를 사용하는 예:

```
$ pcluster list-official-images
{
  "images": [
    {
      "amiId": "ami-015cfefb4e0d6306b2",
      "os": "ubuntu2004",
      "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-x86_64-202202261505
2022-02-26T15-08-34.759Z",
      "version": "3.1.2",
      "architecture": "x86_64"
    },
    {
      "amiId": "ami-036f23237ce49d25b",
      "os": "ubuntu2204",
      "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-x86_64-202202261505
2022-02-26T15-08-17.558Z",
      "version": "3.1.2",
      "architecture": "x86_64"
    },
    {
      "amiId": "ami-09e5327e694d89ef4",
      "os": "ubuntu2004",
      "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.736Z",
      "version": "3.1.2",
      "architecture": "arm64"
    },
  ],
}
```

```

{
  "amiId": "ami-0b9b0874c35f626ae",
  "os": "alinux2",
  "name": "aws-parallelcluster-3.1.2-amzn2-hvm-x86_64-202202261505
2022-02-26T15-08-31.311Z",
  "version": "3.1.2",
  "architecture": "x86_64"
},
{
  "amiId": "ami-0bf6d01f398f3737e",
  "os": "centos7",
  "name": "aws-parallelcluster-3.1.2-centos7-hvm-x86_64-202202261505
2022-02-26T15-08-25.001Z",
  "version": "3.1.2",
  "architecture": "x86_64"
},
{
  "amiId": "ami-0d0de4f95f56374bc",
  "os": "alinux2",
  "name": "aws-parallelcluster-3.1.2-amzn2-hvm-arm64-202202261505
2022-02-26T15-08-46.088Z",
  "version": "3.1.2",
  "architecture": "arm64"
},
{
  "amiId": "ami-0ebf7bc54b8740dc6",
  "os": "ubuntu2204",
  "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.293Z",
  "version": "3.1.2",
  "architecture": "arm64"
}
]
}

```

## pcluster ssh

클러스터 사용자 이름과 IP 주소가 미리 채워진 상태로 ssh 명령을 실행합니다. 임의의 인수가 ssh 명령줄 끝에 추가됩니다.

```

pcluster ssh [-h]
              --cluster-name CLUSTER_NAME
              [--debug]

```

```
[--dryrun DRYRUN]  
[--region REGION]
```

이름 지정된 인수

### **-h, --help**

pcluster ssh에 대한 도움말 텍스트를 표시합니다.

### **--cluster-name, -n *CLUSTER\_NAME***

연결할 클러스터의 이름을 지정합니다.

### **--debug**

디버깅 로깅을 활성화합니다.

### **--dryrun *DRYRUN***

true면 실행할 명령줄을 인쇄하고 종료합니다. (false로 디폴트됩니다.)

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

사용할 AWS 리전 를 지정합니다. 는 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

예제

```
$ pcluster ssh --cluster-name mycluster -i ~/.ssh/id_rsa
```

클러스터의 IP 주소와 사용자 이름이 미리 채워진 상태로 ssh 명령을 실행합니다.

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

## **pcluster update-cluster**

지정된 구성 파일의 설정과 일치하도록 기존 클러스터를 업데이트합니다.

```
pcluster update-cluster [-h]  
    --cluster-configuration CLUSTER_CONFIGURATION  
    --cluster-name CLUSTER_NAME  
    [--debug]  
    [--dryrun DRYRUN]
```

```

[--force-update FORCE_UPDATE]
[--query QUERY]
[--region REGION]
[--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
[--validation-failure-level {INFO,WARNING,ERROR}]

```

이름 지정된 인수

### **-h, --help**

pcluster update-cluster에 대한 도움말 텍스트를 표시합니다.

### **--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

YAML 클러스터 구성 파일을 지정합니다.

### **--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

### **--debug**

디버깅 로깅을 활성화합니다.

### **--dryrun *DRYRUN***

true면 클러스터를 업데이트하거나 리소스를 생성하지 않고 검증을 수행합니다. 이미지 구성을 검증하고 요구 사항을 업데이트하는 데 사용할 수 있습니다. (false로 디폴트됩니다.)

### **--force-update *FORCE\_UPDATE***

true면 업데이트 유효성 검사 오류를 무시하여 업데이트를 강제로 실행합니다. (false로 디폴트됩니다.)

### **--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 AWS 리전 를 지정합니다. 는 클러스터 구성 파일의 [Region](#) 설정, AWS\_DEFAULT\_REGION 환경 변수, 파일 [default] 섹션의 region 설정 또는 --region 매개 변수를 사용하여 AWS 리전 지정해야 합니다. ~/.aws/config

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS* ...]**

억제할 구성 유효성 검사기를 하나 이상 식별하세요.

형식: (ALL|type:[A-Za-z0-9]+)



**--validation-failure-level** *{INFO,WARNING,ERROR}*

업데이트에 대해 보고되는 유효성 검사 실패 수준을 지정합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster update-cluster -c cluster-config.yaml -n cluster-v3 -r us-east-1
{
  "cluster": {
    "clusterName": "cluster-v3",
    "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
    "region": "us-east-1",
    "version": "3.1.4",
    "clusterStatus": "UPDATE_IN_PROGRESS"
  },
  "changeSet": [
    {
      "parameter": "HeadNode.Iam.S3Access",
      "requestedValue": {
        "BucketName": "pc-beta-test",
        "KeyName": "output",
        "EnableWriteAccess": false
      }
    },
    {
      "parameter": "HeadNode.Iam.S3Access",
      "currentValue": {
        "BucketName": "pcluster-east-test-bucket",
        "KeyName": "logs",
        "EnableWriteAccess": true
      }
    }
  ]
}
```

**pcluster update-compute-fleet**

클러스터 컴퓨팅 플릿의 상태를 업데이트합니다.

```
pcluster update-compute-fleet [-h]
```

```

--cluster-name CLUSTER_NAME
--status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

[--debug]
[--query QUERY]
[--region REGION]

```

이름 지정된 인수

## **-h, --help**

pcluster update-compute-fleet에 대한 도움말 텍스트를 표시합니다.

## **--cluster-name, -n *CLUSTER\_NAME***

클러스터의 이름을 지정합니다.

## **--status {START\_REQUESTED,STOP\_REQUESTED,ENABLED,DISABLED}**

클러스터 컴퓨팅 플릿에 적용되는 상태를 지정합니다. 상태 START\_REQUESTED 및 는 스케줄러에 STOP\_REQUESTED ENABLED 해당하고 상태는 Slurm 스케줄러에 DISABLED 해당합니다. AWS Batch

## **--debug**

디버깅 로깅을 활성화합니다.

## **--query *QUERY***

출력에서 수행할 JMESPath 쿼리를 지정합니다.

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

사용할 항목을 지정합니다. AWS 리전은 AWS\_DEFAULT\_REGION 환경 변수, ~/.aws/config 파일 [default] 섹션의 region 설정 또는 --region 매개변수를 사용하여 AWS 리전 지정해야 합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```

$ pcluster update-compute-fleet -n cluster-v3 --status STOP_REQUESTED
{
  "status": "STOP_REQUESTED",
  "lastStatusUpdatedTime": "2022-07-12T20:19:47.653Z"
}

```

```
}
```

## pcluster version

의 AWS ParallelCluster 버전을 표시합니다.

```
pcluster version [-h] [--debug]
```

이름 지정된 인수

### -h, --help

pcluster version에 대한 도움말 텍스트를 표시합니다.

### --debug

디버깅 로깅을 활성화합니다.

AWS ParallelCluster 버전 3.1.4를 사용하는 예:

```
$ pcluster version
{
  "version": "3.1.4"
}
```

## pcluster3-config-converter

AWS ParallelCluster 버전 2 구성 파일을 읽고 AWS ParallelCluster 버전 3 구성 파일을 씁니다.

```
pcluster3-config-converter [-h]
                          [-t CLUSTER_TEMPLATE]
                          [-c CONFIG_FILE]
                          [--force-convert]
                          [-o OUTPUT_FILE]
```

이름 지정된 인수

### -h, --help

pcluster3-config-converter에 대한 도움말 텍스트를 표시합니다.

**-t *CLUSTER\_TEMPLATE*, --cluster-template *CLUSTER\_TEMPLATE***

변환할 구성 파일의 [\[cluster\] 섹션](#)을 지정합니다. 지정하지 않으면 스크립트는 [\[global\] 섹션](#)에서 [cluster-template](#) 파라미터를 찾거나 [cluster default]를 검색합니다.

**-c *CONFIG\_FILE*, --config-file *CONFIG\_FILE***

읽을 AWS ParallelCluster 버전 2 구성 파일을 지정합니다.

**--force-convert**

하나 이상의 설정이 지원되지 않고 권장되지 않는 경우에도 변환할 수 있습니다.

**-o *OUTPUT\_FILE*, --output-file *OUTPUT\_FILE***

기록할 AWS ParallelCluster 버전 3 구성 파일을 지정합니다. 이 파라미터를 지정하지 않으면 구성이 stdout에 기록됩니다.

#### Note

이 pcluster3-config-converter 명령은 AWS ParallelCluster 버전 3.0.1에 추가되었습니다.

## 구성 파일

AWS ParallelCluster 구성 매개변수에 YAML 1.1 파일을 사용합니다.

주제

- [클러스터 구성 파일](#)
- [빌드 이미지 구성 파일](#)

## 클러스터 구성 파일

AWS ParallelCluster 버전 3에서는 별도의 구성 파일을 사용하여 클러스터 인프라 정의와 사용자 지정 AMI 정의를 제어합니다. 모든 구성 파일은 YAML 1.1 파일을 사용합니다. 각 구성 파일에 대한 자세한 정보는 아래에 링크되어 있습니다. 일부 구성 예제는 [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example\\_configs](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs)를 참조하세요.

이러한 객체는 AWS ParallelCluster 버전 3 클러스터 구성에 사용됩니다.

## 주제

- [클러스터 구성 파일 속성](#)
- [Imds 섹션](#)
- [Image 섹션](#)
- [HeadNode 섹션](#)
- [Scheduling 섹션](#)
- [SharedStorage 섹션](#)
- [Iam 섹션](#)
- [LoginNodes 섹션](#)
- [Monitoring 섹션](#)
- [Tags 섹션](#)
- [AdditionalPackages 섹션](#)
- [DirectoryService 섹션](#)
- [DeploymentSettings 섹션](#)

## 클러스터 구성 파일 속성

### Region(선택 사항, String)

클러스터의 AWS 리전 를 지정합니다. 예: us-east-2

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### CustomS3Bucket(선택 사항, String)

클러스터에서 사용하는 리소스 (예: 클러스터 구성 파일) AWS 계정 를 저장하기 위해 에서 생성 되는 Amazon S3 버킷의 이름을 지정합니다. AWS ParallelCluster 클러스터를 AWS 리전 생성 하는 각 버킷에 Amazon S3 버킷 1개를 유지 관리합니다. 기본적으로 이러한 Amazon S3 버킷은 parallelcluster-*hash*-v1-D0-NOT-DELETE으로 명명됩니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다. 강제로 업데이트하면 새 값이 무시되고 이전 값이 사용됩니다.

### AdditionalResources(선택 사항, String)

클러스터와 함께 실행할 추가 AWS CloudFormation 템플릿을 정의합니다. 이 추가 템플릿은 클러스터 외부에 있지만 클러스터 수명 주기의 일부인 리소스를 생성하는 데 사용됩니다.

해당 값은 모든 파라미터가 제공된 퍼블릭 템플릿의 HTTP URL이어야 합니다.

기본값이 없습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## Imds 섹션

(선택 사항) 글로벌 인스턴스 메타데이터 서비스(IMDS) 구성을 지정합니다.

```
Imds:
  ImdsSupport: string
```

## Imds 속성

ImdsSupport(선택 사항, String)

클러스터 노드에서 지원되는 IMDS 버전을 지정합니다. 지원되는 값은 v1.0 및 v2.0입니다. 기본 값은 v2.0입니다.

ImdsSupport이 v1.0로 설정되어 있으면 IMDSv1 및 IMDSv2가 모두 지원됩니다.

ImdsSupport이 v2.0로 설정되어 있으면 IMDSv2만 지원됩니다.

자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [IMDSv2 사용](#)을 참조하십시오.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### Note

AWS ParallelCluster 3.7.0부터 기본값은 v2.0입니다. ImdsSupport v2.0 사용자 지정 작업 호출에서 IMDSv1을 IMDSv2로 교체하고 ImdsSupport을 v2.0로 설정하는 것이 좋습니다.

[ImdsImdsSupport](#)/에 대한 지원은 AWS ParallelCluster 버전 3.3.0에 추가되었습니다.

## Image 섹션

(필수) 클러스터의 운영 체제를 정의합니다.

**Image:**

**Os:** *string*

**CustomAmi:** *string*

**Image 속성**

**Os(필수, String)**

클러스터에 사용할 운영 체제를 지정합니다. 지원되는 값은 `alinux2,,alinux2023,centos7,ubuntu2204,ubuntu2004, rhel8 rocky8rhel9, rocky9` 입니다.

**Note**

RedHat 엔터프라이즈 리눅스 8.7 (rhel8) 은 AWS ParallelCluster 버전 3.6.0부터 추가됩니다.

rhel를 사용하도록 클러스터를 구성하면 지원되는 다른 운영 체제를 사용하도록 클러스터를 구성할 때보다 모든 인스턴스 유형에 대한 온디맨드 비용이 더 높습니다. 요금에 대한 자세한 내용은 [온디맨드 요금 및 Amazon EC2용 Red Hat Enterprise Linux는 어떻게 제공되고 가격이 책정되나요?](#)를 참조하세요.

RedHat 엔터프라이즈 리눅스 9 (rhel9) 는 버전 3.9.0부터 추가됩니다. AWS ParallelCluster

다음 표에 AWS 리전 언급된 특정 사항 외에는 지원되지 않습니다. centos7 다른 모든 AWS 상업 지역은 다음 운영 체제를 모두 지원합니다.

| 파티션(AWS 리전)                           | alinux2 | centos7 | ubuntu2204 및 ubuntu2004 | rhel8 | rhel9 | alinux2023 |
|---------------------------------------|---------|---------|-------------------------|-------|-------|------------|
| 상업용 (모두 구체적으로 AWS 리전 언급되지 않음)         | True    | True    | True                    | True  | True  | True       |
| AWS GovCloud (미국 동부) (us-gov-east-1 ) | True    | False   | True                    | True  | True  | True       |

| 파티션(AWS 리전)                                    | alinux2 | centos7 | ubuntu22<br>4 및<br>ubuntu20<br>4 | rhel8 | rhel9 | alinux202<br>3 |
|------------------------------------------------|---------|---------|----------------------------------|-------|-------|----------------|
| AWS GovCloud (미<br>국 서부) ( us-gov-<br>west-1 ) | True    | False   | True                             | True  | True  | True           |
| 중국(베이징)(cn-<br>north-1 )                       | True    | False   | True                             | True  | True  | True           |
| 중국(닝샤)(cn-<br>northwest-1 )                    | True    | False   | True                             | True  | True  | True           |

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**Note**

AWS ParallelCluster 3.8.0은 록키 리눅스 8을 지원하지만 사전 빌드된 록키 리눅스 8 AMI (x86 및 ARM 아키텍처용) 는 사용할 수 없습니다. AWS ParallelCluster 3.8.0은 사용자 지정 AMI를 사용하여 Rocky Linux 8에서 클러스터를 생성할 수 있도록 지원합니다. 자세한 내용은 [이 링크](#)를 참조하십시오. [운영 체제 고려 사항](#) AWS ParallelCluster 3.9.0은 록키 리눅스 9를 지원하지만 사전 빌드된 록키 리눅스 9 AMI (x86 및 ARM 아키텍처용) 는 사용할 수 없습니다. AWS ParallelCluster 3.9.0은 사용자 지정 AMI를 사용하여 Rocky Linux 9에서 클러스터를 생성할 수 있도록 지원합니다. [자세한 내용은 운영 체제 고려 사항을 참조하십시오.](#)

**CustomAmi(선택 사항, String)**

기본 AMI 대신 헤드 및 컴퓨팅 노드에 사용할 사용자 지정 AMI의 ID를 지정합니다. 자세한 내용은 [AWS ParallelCluster AMI 사용자 지정](#) 항목을 참조하십시오.

사용자 지정 AMI를 시작하기 위해 추가 권한이 필요한 경우 이러한 권한을 사용자 및 헤드 노드 정책 모두에 추가해야 합니다.

예를 들어 사용자 지정 AMI에 암호화된 스냅샷이 연결된 경우 사용자 및 헤드 노드 정책 모두에 다음과 같은 추가 정책이 필요합니다.



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

RedHat 엔터프라이즈 Linux 사용자 지정 AMI를 구축하려면 RHUI (AWS) 리포지토리에서 제공하는 패키지 (, 및) 를 설치하도록 OS를 구성해야 합니다. `rhel-<version>-baseos-rhui-rpms` `rhel-<version>-appstream-rhui-rpms` `codeready-builder-for-rhel-<version>-rhui-rpms` 또한 사용자 지정 AMI의 리포지토리에는 실행 중인 커널 버전과 동일한 버전의 `kernel-devel` 패키지가 포함되어야 합니다.

알려진 제한 사항:

- RHEL 8.2 이상 버전에서만 FSx for Lustre를 지원합니다.
- RHEL 8.7 커널 버전 4.18.0-425.3.1.el8은 FSx for Lustre를 지원하지 않습니다.
- RHEL 8.4 이상 버전만 EFA를 지원합니다.
- AL23은 NICE DCV를 실행하는 데 필요한 그래픽 데스크톱 환경을 포함하지 않으므로 NICE DCV를 지원하지 않습니다. 자세한 내용은 공식 [NICE DCV](#) 설명서를 참조하십시오.

사용자 지정 AMI 검증 경고 문제를 해결하려면 [사용자 지정 AMI 문제 해결](#)을 참조하세요.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

## HeadNode 섹션

(필수) 헤드 노드의 구성을 지정합니다.

[HeadNode](#):

```
InstanceType: string
Networking:
  SubnetId: string
  ElasticIp: string/boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
  Proxy:
    HttpProxyAddress: string
DisableSimultaneousMultithreading: boolean
Ssh:
  KeyName: string
  AllowedIps: string
LocalStorage:
  RootVolume:
    Size: integer
    Encrypted: boolean
    VolumeType: string
    Iops: integer
    Throughput: integer
    DeleteOnTermination: boolean
  EphemeralVolume:
    MountDir: string
SharedStorageType: string
Dcv:
  Enabled: boolean
  Port: integer
  AllowedIps: string
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
        Script: string
        Args:
          - string
    OnNodeConfigured:
      Sequence:
        - Script: string
          Args:
            - string
          Script: string
```

```

  Args:
    - string
  OnNodeUpdated:
    Sequence:
      - Script: string
        Args:
          - string
        Script: string
        Args:
          - string
  Iam:
    InstanceRole: string
    InstanceProfile: string
    S3Access:
      - BucketName: string
        EnableWriteAccess: boolean
        KeyName: string
    AdditionalIamPolicies:
      - Policy: string
  Imds:
    Secured: boolean
  Image:
    CustomAmi: string

```

## HeadNode 속성

InstanceType(필수, String)

헤드 노드의 인스턴스 유형을 지정합니다.

헤드 노드에 사용되는 Amazon EC2 인스턴스 유형을 지정합니다. 인스턴스 유형의 아키텍처는 AWS Batch [InstanceType](#) 또는 Slurm [InstanceType](#) 설정에 사용된 아키텍처와 동일해야 합니다.

### Note

AWS ParallelCluster HeadNode 설정에 대해 다음 인스턴스 유형을 지원하지 않습니다.

- hpc6id

p4d 인스턴스 유형이나 여러 네트워크 인터페이스 또는 네트워크 인터페이스 카드가 있는 다른 인스턴스 유형을 정의하는 경우 퍼블릭 액세스를 제공하도록 [ElasticIp](#)을 true로 설정해야 합니다.

다. AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작된 인스턴스에만 할당할 수 있습니다. 이 경우 [NAT 게이트웨이](#)를 사용하여 클러스터 컴퓨팅 노드에 대한 퍼블릭 액세스를 제공하는 것이 좋습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

DisableSimultaneousMultithreading(선택 사항, Boolean)

true이면 헤드 노드에서의 하이퍼스레딩을 비활성화합니다. 기본값은 false입니다.

모든 인스턴스 유형이 하이퍼 스레딩을 비활성화할 수 있는 것은 아닙니다. 하이퍼스레딩 비활성화를 지원하는 인스턴스 유형 목록은 Amazon EC2 사용 설명서에서 [인스턴스 유형별 각 CPU 코어의 CPU 코어 및 스레드를 참조하십시오.](#)

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

SharedStorageType(선택 사항, String)

내부 공유 데이터에 사용되는 스토리지 유형을 지정합니다. 내부 공유 데이터에는 클러스터를 관리하는 데 AWS ParallelCluster 사용되는 데이터와 공유 파일 시스템 볼륨을 마운트하기 위한 기본 공유 데이터 ([SharedStorage 섹션](#) as a Mount 디렉토리에 지정되지 않은 /home 경우)가 포함됩니다. 내부 공유 데이터에 대한 자세한 내용은 을 참조하십시오. [AWS ParallelCluster 내부 디렉터리](#)

기본 스토리지 Ebs 유형인 경우 헤드 노드는 루트 볼륨의 일부를 NFS를 사용하는 컴퓨팅 노드 및 로그인 노드의 공유 디렉터리로 내보냅니다.

만약Efs, Parallelcluster는 공유 내부 데이터에 사용할 EFS 파일 시스템을 생성할 것이며, /home

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

#### Note

클러스터가 확장되면 헤드 노드가 NFS 익스포트를 사용하여 루트 볼륨의 데이터를 컴퓨팅 노드와 공유하므로 EBS 스토리지 유형에서 성능 병목 현상이 발생할 수 있습니다. EFS를 사용하면 클러스터가 확장될 때 NFS 내보내기를 피하고 클러스터와 관련된 성능 병목 현상을 피할 수 있습니다. 작은 파일 및 설치 프로세스에 대한 읽기/쓰기 잠재력을 극대화하려면 EBS를 선택하는 것이 좋습니다. 확장하려면 EFS를 선택하십시오.

## Networking

(필수) 헤드 노드의 네트워킹 구성을 정의합니다.

**Networking:****SubnetId:** *string***ElasticIp:** *string/boolean***SecurityGroups:**- *string***AdditionalSecurityGroups:**- *string***Proxy:****HttpProxyAddress:** *string*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**Networking 속성****SubnetId(필수, String)**

헤드 노드를 프로비저닝할 기존 서브넷의 ID를 지정합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**ElasticIp(선택 사항, String)**

헤드 노드에 탄력적 IP 주소를 생성하거나 할당합니다. 지원되는 값은 true, false 또는 기존 탄력적 IP 주소의 ID입니다. 기본값은 false입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**SecurityGroups(선택 사항, [String])**

헤드 노드에 사용할 Amazon VPC 보안 그룹 ID 목록입니다. 이는 이 속성이 포함되지 않은 경우 AWS ParallelCluster 생성되는 보안 그룹을 대체합니다.

보안 그룹이 [SharedStorage](#) 시스템에 맞게 구성되어 있는지 확인하십시오.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**AdditionalSecurityGroups(선택 사항, [String])**

헤드 노드에 사용할 추가 Amazon VPC 보안 그룹 ID 목록입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**Proxy(선택 사항)**

헤드 노드의 프록시 설정을 지정합니다.

**Proxy:****HttpProxyAddress:** *string*

HttpProxyAddress(선택 사항, String)

HTTP 또는 HTTPS 프록시 서버, 일반적으로 `https://x.x.x.x:8080`을 정의합니다.

기본값이 없습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**Ssh**

(선택 사항) 헤드 노드에 대한 SSH 액세스를 위한 구성을 정의합니다.

**Ssh:****KeyName:** *string***AllowedIps:** *string*

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**Ssh 속성**

KeyName(선택 사항, String)

헤드 노드로의 SSH 액세스가 가능하도록 기존 Amazon EC2 키 쌍의 이름을 지정합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

AllowedIps(선택 사항, String)

헤드 노드에 대한 SSH 연결을 위한 CIDR 형식의 IP 범위 또는 접두사 목록 ID를 지정합니다. 기본 값은 `0.0.0.0/0`입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**LocalStorage**

(선택 사항) 헤드 노드의 로컬 스토리지 구성을 정의합니다.

**LocalStorage:****RootVolume:**

```

Size: integer
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
DeleteOnTermination: boolean
EphemeralVolume:
MountDir: string

```

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## LocalStorage 속성

### RootVolume (필수)

헤드 노드의 루트 볼륨 스토리지를 지정합니다.

```

RootVolume:
Size: integer
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
DeleteOnTermination: boolean

```

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Size(선택 사항, Integer)

헤드 노드 루트 볼륨 크기를 기비바이트(GiB) 단위로 지정합니다. 기본 크기는 AMI에서 가져옵니다. 다른 크기를 사용하려면 AMI에서 growroot를 지원해야 합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### Encrypted(선택 사항, Boolean)

루트 볼륨이 암호화되는지 여부를 지정합니다. 기본값은 true입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### VolumeType(선택 사항, String)

Amazon EBS 볼륨 유형을 지정합니다. 지원되는 값은 gp2, gp3, io1, io2, sc1, st1, standard입니다. 기본값은 gp3입니다.

자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 [Amazon EBS 볼륨 유형](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### Iops(선택 사항, Integer)

io1, io2, gp3 유형 볼륨의 IOPS 수를 정의합니다.

기본 값, 지원되는 값, volume\_iops:volume\_size 비율은 VolumeType 및 Size에 따라 달라집니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

VolumeType = io1

기본 Iops = 100

지원되는 값 Iops = 100~64000†

최대 Iops와 Size의 비율 = 50IOPS/GiB. 5000 IOPS는 최소 100GiB의 Size가 필요합니다.

VolumeType = io2

기본 Iops = 100

지원되는 값 Iops = 100~64000(io2 Block Express 볼륨의 경우 256000)†

최대 Iops와 Size 비율 = 500IOPS/GiB. 5000 IOPS에는 최소 10GiB의 Size가 필요합니다.

VolumeType = gp3

기본 Iops = 3000

지원되는 값 Iops = 3000~16000

최대 Iops와 Size 비율 = 500IOPS/GiB. 5000 IOPS에는 최소 10GiB Size가 필요합니다.

† 최대 IOPS는 32,000 IOPS 이상으로 프로비저닝된 [Nitro 시스템 기반 인스턴스](#)에서만 보장됩니다. 다른 인스턴스는 최대 32,000 IOPS를 보장합니다. 이전 io1 볼륨은 [볼륨을 수정](#)하지 않는 한 전체 성능에 도달할 수 없습니다. io2 블록 익스프레스 볼륨은 R5b 인스턴스 유형에서 최



대 256000의 Iops 값을 지원합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [io2 블록 인 스프레스 볼륨](#)을 참조하십시오.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

Throughput(선택 사항, Integer)

gp3 볼륨 유형의 처리량을 MiB/s 단위로 정의합니다. 이 설정은 VolumeType가 gp3일 때만 유효합니다. 기본값은 125입니다. 지원되는 값: 125-1000MiB/s

Throughput:Iops의 비율은 0.25를 초과할 수 없습니다. 1000MiB/s의 최대 처리량을 위해서는 Iops 설정이 최소 4000이어야 합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

DeleteOnTermination(선택 사항, Boolean)

헤드 노드가 종료될 때 루트 볼륨을 삭제할지 여부를 지정합니다. 기본값은 true입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

EphemeralVolume(선택 사항)

모든 인스턴스 스토어 볼륨의 세부 정보를 지정합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 스토어 볼륨](#)을 참조하세요.

EphemeralVolume:  
MountDir: *string*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

MountDir(선택 사항, String)

인스턴스 스토어 볼륨의 탑재 디렉터리를 지정합니다. 기본값은 /scratch입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Dcv

(선택 사항) 헤드 노드에서 실행되는 NICE DCV 서버의 구성 설정을 정의합니다.

자세한 정보는 [NICE DCV를 통해 헤드 노드에 연결합니다.](#)을 참조하세요.

Dcv:

`Enabled`: *boolean*  
`Port`: *integer*  
`AllowedIps`: *string*

### Important

기본적으로 NICE DCV 포트 설정은 모든 IPv4 AWS ParallelCluster 주소에 개방되어 있습니다. 그러나 사용자는 NICE DCV 세션에 대한 URL이 있는 경우에만 NICE DCV 포트에 연결하고 `pcluster dcv-connect`에서 URL이 반환된 후 30초 이내에 NICE DCV 세션에 연결할 수 있습니다. `AllowedIps` 설정을 사용해 CIDR 형식 IP 범위의 NICE DCV 포트로의 액세스를 추가 제한하고, `Port` 설정을 사용하여 비표준 포트를 설정합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Dcv 속성

Enabled(필수, Boolean)

헤드 노드에서의 NICE DCV 활성화 여부를 지정합니다. 기본값은 `false`입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### Note

NICE DCV는 NICE DCV 클라이언트와 헤드 노드에서 실행 중인 NICE DCV 서버 간의 트래픽을 보호하는 데 사용되는 자체 서명된 인증서를 자동으로 생성합니다. 자체 인증서를 구성하려면 [NICE DCV HTTPS 인증서](#) 섹션을 참조하세요.

Port(선택 사항, Integer)

NICE DCV의 포트를 지정합니다. 기본값은 8443입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

AllowedIps (선택 사항, 권장됨, String)

NICE DCV에 연결할 CIDR 형식의 IP 범위를 지정합니다. 이 설정은 보안 그룹을 AWS ParallelCluster 생성할 때만 사용됩니다. 기본값은 어떤 인터넷 주소에서도 액세스할 수 있는 `0.0.0.0/0`입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## CustomActions

(선택 사항) 헤드 노드상에서 실행할 사용자 지정 스크립트를 지정합니다.

```

CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeConfigured:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeUpdated:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string

```

### CustomActions 속성

#### OnNodeStart(선택 사항)

노드 배포 부트스트랩 작업이 시작되기 전에 헤드 노드에서 실행할 단일 스크립트 또는 스크립트 시퀀스를 지정합니다. 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

#### Sequence(선택 사항)

실행할 스크립트 목록. AWS ParallelCluster 구성 파일에 나열된 순서대로 첫 번째 스크립트부터 시작하여 스크립트를 실행합니다.

Script(필수, String)

사용할 파일을 지정합니다. 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

Args(선택 사항, [String])

스크립트에 전달할 인수 목록

Script(필수, String)

단일 스크립트에 사용할 파일을 지정합니다. 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

Args(선택 사항, [String])

단일 스크립트에 전달할 인수 목록

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

OnNodeConfigured(선택 사항)

노드 부트스트랩 작업이 완료된 후 헤드 노드에서 실행할 단일 스크립트 또는 스크립트 시퀀스를 지정합니다. 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

Sequence(선택 사항)

실행할 스크립트 목록을 지정합니다.

Script(필수, String)

사용할 파일을 지정합니다. 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

Args(선택 사항, [String])

스크립트에 전달할 인수 목록

Script(필수, String)

단일 스크립트에 사용할 파일을 지정합니다. 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

Args(선택 사항, [String])

단일 스크립트에 전달할 인수 목록

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

## OnNodeUpdated(선택 사항)

노드 업데이트 작업이 완료된 후 헤드 노드에서 실행할 단일 스크립트 또는 스크립트 시퀀스를 지정합니다. 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

### Sequence(선택 사항)

실행할 스크립트 목록을 지정합니다.

Script(필수, String)

사용할 파일을 지정합니다. 파일 경로는 `https://` 또는 `s3://`로 시작되어야 합니다.

Args(선택 사항, [String])

스크립트에 전달할 인수 목록

### Script(필수, String)

단일 스크립트에 사용할 파일을 지정합니다. 파일 경로는 `https://` 또는 `s3://`로 시작되어야 합니다.

Args(선택 사항, [String])

단일 스크립트에 전달할 인수 목록

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### Note

OnNodeUpdated AWS ParallelCluster 3.4.0부터 추가됩니다.

Sequence AWS ParallelCluster 버전 3.6.0부터 추가되었습니다. 지정하는 Sequence 경우 사용자 지정 작업에 사용할 여러 스크립트를 나열할 수 있습니다. AWS ParallelCluster 스크립트는 포함하지 않고 단일 스크립트로 사용자 정의 액션을 구성할 수 있도록 계속 지원합니다Sequence.

AWS ParallelCluster 단일 스크립트와 Sequence 동일한 사용자 지정 작업을 모두 포함하는 것은 지원하지 않습니다.

## Iam

(선택 사항) 헤드 노드에서 클러스터의 기본 인스턴스 역할 또는 인스턴스 프로파일을 재정의하는 데 사용할 인스턴스 역할 또는 인스턴스 프로파일을 지정합니다.

```

Iam:
  InstanceRole: string
  InstanceProfile: string
  S3Access:
    - BucketName: string
      EnableWriteAccess: boolean
      KeyName: string
  AdditionalIamPolicies:
    - Policy: string

```

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## Iam 속성

### InstanceProfile(선택 사항, String)

기본 헤드 노드 인스턴스 프로파일을 재정의할 인스턴스 프로파일을 지정합니다. InstanceProfile 및 InstanceRole를 둘 다 지정할 수 없습니다. 형식은 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`입니다.

이를 지정하면 S3Access 및 AdditionalIamPolicies 설정을 지정할 수 없습니다.

기능을 추가하려면 새 권한이 필요한 AWS ParallelCluster 경우가 많으므로 S3Access 및 AdditionalIamPolicies 설정 중 하나 또는 둘 다를 지정하는 것이 좋습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### InstanceRole(선택 사항, String)

기본 헤드 노드 인스턴스 역할을 재정의할 인스턴스 역할을 지정합니다.

InstanceProfile 및 InstanceRole를 둘 다 지정할 수 없습니다. 형식은 `arn:Partition:iam::Account:role/RoleName`입니다.

이를 지정하면 S3Access 및 AdditionalIamPolicies 설정을 지정할 수 없습니다.

기능을 추가하려면 새 권한이 필요한 AWS ParallelCluster 경우가 많으므로 S3Access 및 AdditionalIamPolicies 설정 중 하나 또는 둘 다를 지정하는 것이 좋습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## S3Access

### S3Access(선택 사항)

버킷을 지정합니다. 이는 버킷에 지정된 액세스 권한을 부여하는 정책을 생성하는 데 사용됩니다.

이를 지정하면 InstanceProfile 및 InstanceRole 설정을 지정할 수 없습니다.

기능을 추가하려면 새 권한이 필요한 AWS ParallelCluster 경우가 많으므로 S3Access 및 AdditionalIamPolicies 설정 중 하나 또는 둘 다를 지정하는 것이 좋습니다.

#### S3Access:

- `BucketName`: *string*
- `EnableWriteAccess`: *boolean*
- `KeyName`: *string*

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### BucketName(필수, String)

버킷의 이름입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### KeyName(선택 사항, String)

버킷의 키입니다. 기본값은 "\*"입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### EnableWriteAccess(선택 사항, Boolean)

버킷에 대해 쓰기 액세스가 활성화되어 있는지 여부를 나타냅니다. 기본값은 false입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## AdditionalIamPolicies

### AdditionalIamPolicies(선택 사항)

Amazon EC2에 대한 IAM 정책의 Amazon 리소스 이름(ARN) 목록을 쉼표로 구분하여 지정합니다. 이 목록은 에 필요한 권한 외에도 헤드 노드에 사용되는 루트 역할에 첨부되어 AWS ParallelCluster 있습니다.

IAM 정책 이름과 해당 ARN은 서로 다릅니다. 이름은 사용할 수 없습니다.

이를 지정하면 InstanceProfile 및 InstanceRole 설정을 지정할 수 없습니다.

필요한 권한에 AdditionalIamPolicies 추가되고 필요한 모든 권한이 InstanceRole 포함되어야 AdditionalIamPolicies 하므로 사용하는 것이 좋습니다. AWS ParallelCluster 기능이 추가됨에 따라 필요한 권한은 종종 릴리스마다 변경됩니다.

기본값이 없습니다.

AdditionalIamPolicies:

- Policy: *string*

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

Policy(선택 사항, [String])

IAM 정책 목록

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## Imds

(선택 사항) 인스턴스 메타데이터 서비스(IMDS)를 위한 속성을 지정합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 메타데이터 서비스 버전 2의 [작동 방식을](#) 참조하십시오.

Imds:

Secured: *boolean*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Imds 속성

Secured(선택 사항, Boolean)

true이면 헤드 노드의 IMDS(및 인스턴스 프로파일 보안 인증)에 대한 액세스를 슈퍼유저의 하위 집합으로 제한합니다.

false이면 헤드 노드의 모든 사용자가 헤드 노드의 IMDS에 액세스할 수 있습니다.

다음 사용자는 헤드 노드의 IMDS에 액세스할 수 있습니다.

- 루트 사용자
- 클러스터 관리 사용자(기본값 pc-cluster-admin)



- 운영 체제별 기본 사용자 (ec2-user 아마존 리눅스 2 RedHat, ubuntu 우분투 18.04, centos CentOS 7)

기본값은 true입니다.

default 사용자는 클러스터가 리소스와 상호 작용하는 데 필요한 권한을 갖도록 할 책임이 있습니다. AWS default 사용자 IMDS 액세스를 비활성화하면 컴퓨팅 노드를 관리할 AWS ParallelCluster 수 없고 작동이 중지됩니다. default 사용자 IMDS 액세스를 비활성화하지 마세요.

헤드 노드의 IMDS에 대한 액세스 권한이 부여된 사용자는 [헤드 노드의 인스턴스 프로파일에](#) 포함된 권한을 사용할 수 있습니다. 예를 들어 이러한 권한을 사용하여 Amazon EC2 인스턴스를 시작하거나 클러스터가 인증에 사용하도록 구성된 AD 도메인의 암호를 읽을 수 있습니다.

IMDS 액세스를 제한하기 위해 체인을 AWS ParallelCluster 관리합니다. iptables

sudo 액세스 권한이 있는 클러스터 사용자는 다음 명령을 실행하여 default 사용자를 포함한 다른 개별 사용자의 헤드 노드의 IMDS 액세스를 선택적으로 활성화 또는 비활성화할 수 있습니다.

```
$ sudo /opt/parallelcluster/scripts/imds/imds-access.sh --allow <USERNAME>
```

이 명령의 --deny 옵션으로 사용자 IMDS 액세스를 비활성화할 수 있습니다.

default 사용자 IMDS 액세스를 모르는 사이에 비활성화한 경우 --allow 옵션을 사용하여 권한을 복원할 수 있습니다.

#### Note

iptables 또는 ip6tables 규칙을 사용자 지정하면 헤드 노드에서 IMDS 액세스를 제한하는 데 사용되는 메커니즘이 방해될 수 있습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Image

(선택 사항) 헤드 노드의 사용자 지정 이미지를 정의합니다.

```
Image:
  CustomAmi: string
```

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Image 속성

### CustomAmi(선택 사항, String)

기본 AMI 대신 헤드 노드에 사용할 사용자 지정 AMI의 ID를 지정합니다. 자세한 내용은 [AWS ParallelCluster AMI 사용자 지정](#) 항목을 참조하세요.

사용자 지정 AMI를 시작하기 위해 추가 권한이 필요한 경우 이러한 권한을 사용자 및 헤드 노드 정책 모두에 추가해야 합니다.

예를 들어 사용자 지정 AMI에 암호화된 스냅샷이 연결된 경우 사용자 및 헤드 노드 정책 모두에 다음과 같은 추가 정책이 필요합니다.

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

사용자 지정 AMI 검증 경고 문제를 해결하려면 [사용자 지정 AMI 문제 해결](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Scheduling 섹션

(필수) 클러스터에서 사용되는 작업 스케줄러와 작업 스케줄러가 관리하는 컴퓨팅 인스턴스를 정의합니다. Slurm or AWS Batch 스케줄러를 사용할 수 있습니다. 각각은 서로 다른 설정 및 속성 세트를 지원합니다.

## 주제

- [Scheduling 속성](#)
- [AwsBatchQueues](#)
- [SlurmQueues](#)
- [SlurmSettings](#)

Scheduling:Scheduler: slurmScalingStrategy: *string*SlurmSettings:MungeKeySecretArn: *string*ScaledownIdleTime: *integer*QueueUpdateStrategy: *string*EnableMemoryBasedScheduling: *boolean*CustomSlurmSettings: *[dict]*CustomSlurmSettingsIncludeFile: *string*Database:Uri: *string*UserName: *string*PasswordSecretArn: *string*DatabaseName: *string*ExternalSlurmdbd: *boolean*Host: *string*Port: *integer*Dns:DisableManagedDns: *boolean*HostedZoneId: *string*UseEc2Hostnames: *boolean*SlurmQueues:- Name: *string*ComputeSettings:LocalStorage:RootVolume:Size: *integer*Encrypted: *boolean*VolumeType: *string*Iops: *integer*Throughput: *integer*EphemeralVolume:MountDir: *string*CapacityReservationTarget:

```
CapacityReservationId: string  
CapacityReservationResourceGroupArn: string  
CapacityType: string  
AllocationStrategy: string  
JobExclusiveAllocation: boolean  
CustomSlurmSettings: dict  
Tags:  
- Key: string  
  Value: string  
HealthChecks:  
  Gpu:  
    Enabled: boolean  
Networking:  
  SubnetIds:  
    - string  
  AssignPublicIp: boolean  
  SecurityGroups:  
    - string  
  AdditionalSecurityGroups:  
    - string  
  PlacementGroup:  
    Enabled: boolean  
    Id: string  
    Name: string  
  Proxy:  
    HttpProxyAddress: string  
ComputeResources:  
- Name: string  
  InstanceType: string  
  Instances:  
    - InstanceType: string  
  MinCount: integer  
  MaxCount: integer  
  DynamicNodePriority: integer  
  StaticNodePriority: integer  
  SpotPrice: float  
  DisableSimultaneousMultithreading: boolean  
  SchedulableMemory: integer  
  HealthChecks:  
    Gpu:  
      Enabled: boolean  
    Efa:  
      Enabled: boolean  
      GdrSupport: boolean
```

```

CapacityReservationTarget:
  CapacityReservationId: string
  CapacityReservationResourceGroupArn: string
Networking:
  PlacementGroup:
    Enabled: boolean
    Name: string
  CustomSlurmSettings: dict
Tags:
  - Key: string
    Value: string
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeConfigured:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
Iam:
  InstanceProfile: string
  InstanceRole: string
  S3Access:
    - BucketName: string
      EnableWriteAccess: boolean
      KeyName: string
  AdditionalIamPolicies:
    - Policy: string
Image:
  CustomAmi: string

```

Scheduling:

```

Scheduler: awsbatch
AwsBatchQueues:

```

```

- Name: string
CapacityType: string
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
ComputeResources: # this maps to a Batch compute environment (initially we
support only 1)
  - Name: string
    InstanceTypes:
      - string
    MinvCpus: integer
    DesiredvCpus: integer
    MaxvCpus: integer
    SpotBidPercentage: float

```

## Scheduling 속성

### Scheduler(필수, String)

사용되는 스케줄러 유형을 지정합니다. 지원되는 값은 slurm 및 awsbatch입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

#### Note

awsbatch은 alinux2 운영 체제 및 x86\_64 플랫폼만 지원합니다.

### ScalingStrategy(선택 사항, String)

동적 Slurm 노드의 확장 방법을 선택할 수 있습니다. 지원되는 값은 greedy-all-or-nothing 이며 all-or-nothing, best-effort 기본값은 all-or-nothing입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**Note**

스케일링 전략은 Slurm이 재개할 노드에만 적용되며, 최종적으로 이미 실행 중인 노드에는 적용되지 않습니다.

- `all-or-nothing`이 전략은 확장 프로세스가 끝날 때 유휴 인스턴스가 발생하지 않도록 하기 위한 `a`를 엄격하게 따릅니다. `all-or-nothing-approach` `all-or-nothing` 기본적으로 운영되므로 완전히 확장하거나 전혀 확장하지 않을 수 있습니다. 작업에 500개 이상의 노드가 필요하거나 여러 컴퓨팅 리소스에 걸쳐 있는 경우 일시적으로 시작된 인스턴스로 인해 추가 비용이 발생할 수 있다는 점에 유의하십시오. 이 전략은 가능한 세 가지 확장 전략 중에서 처리량이 가장 낮습니다. 조정 시간은 Slurm 재개 프로그램 실행당 제출된 작업 수에 따라 달라집니다. 또한 실행당 기본 `RunInstances` 자원 계정 제한 (기본 인스턴스 1000개) 을 크게 초과하여 확장할 수 없습니다. 자세한 내용은 [Amazon EC2 API](#) 제한 설명서에서 확인할 수 있습니다.
- `greedy-all-or-nothing` 이 전략과 마찬가지로 이 `all-or-nothing` 전략은 확장 후 유휴 인스턴스를 피하는 것을 목표로 합니다. 이 전략에서는 `all-or-nothing` 접근 방식보다 높은 처리량을 달성하기 위해 조정 프로세스 중에 일시적으로 오버스케일링을 수행할 수 있지만, `RunInstances` 리소스 계정 한도에 따라 1,000개의 인스턴스라는 동일한 규모 조정 한도가 적용됩니다.
- `best-effort` 이 전략은 확장 프로세스가 끝날 때 일부 인스턴스가 유휴 상태일 수 있더라도 높은 처리량을 우선시합니다. 작업에서 요청한 만큼의 노드를 할당하려고 시도하지만 전체 요청을 충족하지 못할 수도 있습니다. 다른 전략과 달리 최선의 접근 방식은 표준 `RunInstances` 한도보다 많은 인스턴스를 누적할 수 있지만, 이는 여러 확장 프로세스 실행 과정에서 리소스가 유휴 상태로 유지되는 대가를 치르게 됩니다.

각 전략은 다양한 규모 조정 요구 사항을 충족하도록 설계되었으므로 특정 요구 사항 및 제약 조건에 맞는 전략을 선택할 수 있습니다.

## AwsBatchQueues

(선택 사항) AWS Batch 대기열 설정. 대기열은 하나만 지원됩니다. [Scheduler](#)이 `awsbatch`로 설정된 경우 이 섹션은 필수입니다. `awsbatch` 스케줄러에 대한 자세한 내용은 [네트워킹 설정 및 AWS Batch \(awsbatch\)](#)을 참조하세요.

### AwsBatchQueues:

- Name: *string*
- CapacityType: *string*
- Networking:

```

SubnetIds:
  - string
AssignPublicIp: boolean
SecurityGroups:
  - string
AdditionalSecurityGroups:
  - string
ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
  - Name: string
    InstanceTypes:
      - string
    MinvCpus: integer
    DesiredvCpus: integer
    MaxvCpus: integer
    SpotBidPercentage: float

```

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## AwsBatchQueues 속성

### Name(필수, **String**)

AWS Batch 대기열의 이름.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### CapacityType(선택 사항, **String**)

AWS Batch 대기열이 사용하는 컴퓨팅 리소스의 유형. 지원되는 값은 ONDEMAND SPOT 또는 CAPACITY\_BLOCK 입니다. 기본값은 ONDEMAND입니다.

#### Note

CapacityType을 SPOT로 설정하는 경우 계정에 AWSServiceRoleForEC2Spot 서비스 연결 역할이 포함되어야 합니다. 다음 AWS CLI 명령을 사용하여 이 역할을 생성할 수 있습니다.

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

자세한 내용은 Linux [인스턴스용 Amazon Amazon EC2 사용 설명서의 스팟 인스턴스 요청에 대한 서비스 연결 역할을 참조하십시오.](#)



업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

## Networking

(필수) 대기열의 네트워킹 구성을 정의합니다. AWS Batch

```
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
```

### Networking 속성

#### SubnetIds(필수, [String])

AWS Batch 큐를 프로비저닝할 기존 서브넷의 ID를 지정합니다. 현재 하나의 서브넷만 지원됩니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

#### AssignPublicIp(선택 사항, String)

큐의 노드에 퍼블릭 IP 주소를 만들거나 할당합니다. AWS Batch 지원되는 값은 true 및 false입니다. 기본값은 지정한 서브넷에 따라 달라집니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

#### SecurityGroups(선택 사항, [String])

AWS Batch 대기열이 사용하는 보안 그룹 목록. 보안 그룹을 지정하지 않는 경우 새 보안 그룹을 AWS ParallelCluster 생성합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### AdditionalSecurityGroups(선택 사항, [String])

AWS Batch 대기열이 사용하는 보안 그룹 목록.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## ComputeResources

(필수) AWS Batch 대기열의 ComputeResources 구성을 정의합니다.

```

ComputeResources: # this maps to a Batch compute environment (initially we support
only 1)
- Name: string
  InstanceTypes:
  - string
  MinvCpus: integer
  DesiredvCpus: integer
  MaxvCpus: integer
  SpotBidPercentage: float

```

### ComputeResources 속성

#### Name(필수, String)

AWS Batch 대기열 컴퓨팅 환경의 이름.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

#### InstanceTypes(필수, [String])

인스턴스 유형으로 구성된 AWS Batch 컴퓨팅 환경 배열. 모든 인스턴스 유형은 x86\_64 아키텍처를 사용해야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

#### MinvCpus(선택 사항, Integer)

AWS Batch 컴퓨팅 환경에서 사용할 수 있는 최소 vCPU 수.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### DesiredVcpus(선택 사항, Integer)

컴퓨팅 환경에서 원하는 vCPU 수. AWS Batch 이 값을 작업 대기열의 수요 사이에서 MinvCpus 또는 필요에 MaxvCpus 따라 조정합니다.

업데이트 정책: 이 설정은 업데이트 중에 분석되지 않습니다.

#### MaxvCpus(선택 사항, Integer)

컴퓨팅 환경의 최대 vCPU 수입니다. AWS Batch 이 값은 DesiredVcpus보다 낮은 값으로 설정할 수 없습니다.

업데이트 정책: 업데이트 중에는 이 설정을 줄일 수 없습니다.

## SpotBidPercentage(선택 사항, Float)

Amazon EC2 스팟 인스턴스 가격이 인스턴스 시작 전에 도달할 수 있는 인스턴스 유형의 온디맨드 가격의 최대 비율입니다. 기본값은 100(100%)입니다. 지원되는 범위는 1~100입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## SlurmQueues

(선택 사항) Slurm 대기열 설정 [Scheduler](#)이 slurm로 설정된 경우 이 섹션은 필수입니다.

### SlurmQueues:

- Name: *string*
- ComputeSettings:
  - LocalStorage:
    - RootVolume:
      - Size: *integer*
      - Encrypted: *boolean*
      - VolumeType: *string*
      - Iops: *integer*
      - Throughput: *integer*
    - EphemeralVolume:
      - MountDir: *string*
  - CapacityReservationTarget:
    - CapacityReservationId: *string*
    - CapacityReservationResourceGroupArn: *string*
  - CapacityType: *string*
  - AllocationStrategy: *string*
  - JobExclusiveAllocation: *boolean*
  - CustomSlurmSettings: *dict*
  - Tags:
    - Key: *string*
    - Value: *string*
  - HealthChecks:
    - Gpu:
      - Enabled: *boolean*
  - Networking:
    - SubnetIds:
      - *string*
    - AssignPublicIp: *boolean*
    - SecurityGroups:
      - *string*

```
AdditionalSecurityGroups:  
  - string  
PlacementGroup:  
  Enabled: boolean  
  Id: string  
  Name: string  
Proxy:  
  HttpProxyAddress: string  
ComputeResources:  
  - Name: string  
    InstanceType: string  
    Instances:  
      - InstanceType: string  
    MinCount: integer  
    MaxCount: integer  
    DynamicNodePriority: integer  
    StaticNodePriority: integer  
    SpotPrice: float  
    DisableSimultaneousMultithreading: boolean  
    SchedulableMemory: integer  
    HealthChecks:  
      Gpu:  
        Enabled: boolean  
      Efa:  
        Enabled: boolean  
        GdrSupport: boolean  
    CapacityReservationTarget:  
      CapacityReservationId: string  
      CapacityReservationResourceGroupArn: string  
    Networking:  
      PlacementGroup:  
        Enabled: boolean  
        Name: string  
    CustomSlurmSettings: dict  
    Tags:  
      - Key: string  
        Value: string  
CustomActions:  
  OnNodeStart:  
    Sequence:  
      - Script: string  
        Args:  
          - string  
    Script: string
```

```

  Args:
    - string
  OnNodeConfigured:
    Sequence:
      - Script: string
        Args:
          - string
        Script: string
        Args:
          - string
    Iam:
      InstanceProfile: string
      InstanceRole: string
      S3Access:
        - BucketName: string
          EnableWriteAccess: boolean
          KeyName: string
      AdditionalIamPolicies:
        - Policy: string
    Image:
      CustomAmi: string

```

업데이트 정책: 이 목록 값 설정의 경우 업데이트 중에 새 값을 추가할 수 있으며, 또는 기존 값을 제거할 때 컴퓨팅 플릿을 중지해야 합니다.

## SlurmQueues 속성

### Name(필수, String)

Slurm 대기열의 이름입니다.

#### Note

클러스터 크기는 업데이트 중에 변경될 수 있습니다. 자세한 내용은 [클러스터 용량 크기 및 업데이트](#)를 참조하십시오.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## CapacityReservationTarget

### Note

CapacityReservationTarget AWS ParallelCluster 버전 3.3.0에 추가되었습니다.

### CapacityReservationTarget:

CapacityReservationId: *string*

CapacityReservationResourceGroupArn: *string*

대기열의 컴퓨팅 리소스에 대한 온디맨드 용량 예약을 지정합니다.

### CapacityReservationId(선택 사항, **String**)

대기열의 컴퓨팅 리소스를 대상으로 하는 기존 용량 예약의 ID입니다. ID는 [ML용 ODCR 또는 용량 블록](#)을 참조할 수 있습니다.

예약은 인스턴스가 사용하는 것과 동일한 플랫폼을 사용해야 합니다. 예를 들어, 인스턴스가 rhe18에서 실행되는 경우 용량 예약은 Red Hat Enterprise Linux 플랫폼에서 실행되어야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [지원되는 플랫폼](#)을 참조하세요.

### Note

클러스터 구성에 [Instances](#)를 포함하는 경우 이 대기열 레벨 CapacityReservationId 설정을 구성에서 제외해야 합니다.

### CapacityReservationResourceGroupArn(선택 사항, **String**)

대기열의 컴퓨팅 리소스에 대해 서비스에 연결된 용량 예약 그룹 역할을 하는 리소스 그룹의 Amazon 리소스 이름(ARN)입니다. AWS ParallelCluster 은 다음 조건에 따라 리소스 그룹에서 가장 적절한 용량 예약을 식별하고 사용합니다.

- [SlurmQueuesSlurmQueuesComputeResources/Networking](#) 또는 //에서 활성화된 경우 [Networking](#), 인스턴스 유형을 대상으로 하는 리소스 그룹을 선택하고 PlacementGroup 컴퓨팅 PlacementGroup 리소스가 있는 경우 컴퓨팅 리소스로 사용할 리소스 그룹을 AWS ParallelCluster 선택합니다.

PlacementGroup는 [ComputeResources](#)에 정의된 인스턴스 유형 중 하나를 대상으로 해야 합니다.

- [SlurmQueuesSlurmQueuesComputeResources/Networking](#) 또는 //에서 PlacementGroup 활성화되지 않은 경우 [Networking](#), 컴퓨팅 리소스가 있는 경우 컴퓨팅 리소스의 인스턴스 유형만 대상으로 하는 리소스 그룹을 AWS ParallelCluster 선택합니다.

리소스 그룹에는 대기열의 모든 컴퓨팅 리소스 및 가용 영역에 걸쳐 가용 영역에 예약된 각 인스턴스 유형에 대해 하나 이상의 ODCR이 있어야 합니다. 자세한 내용은 [온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작](#) 항목을 참조하세요.

다중 서브넷 구성 요구 사항에 대한 자세한 내용은 [Networking/SubnetIds](#)를 참조하세요.

#### Note

AWS ParallelCluster 버전 3.4.0에는 다중 가용 영역이 추가되었습니다.

[업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.](#)

### CapacityType(선택 사항, String)

Slurm 대기열에 사용되는 컴퓨팅 리소스의 유형입니다. 지원되는 값은 ONDEMAND, SPOT 또는 CAPACITY\_BLOCK 입니다. 기본값은 ONDEMAND입니다.

#### Note

CapacityType을 SPOT로 설정하는 경우 계정에 AWSServiceRoleForEC2Spot 서비스 연결 역할이 있어야 합니다. 다음 AWS CLI 명령을 사용하여 이 역할을 생성할 수 있습니다.

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

자세한 내용은 Linux [인스턴스용 Amazon Amazon EC2 사용 설명서의 스팟 인스턴스 요청에 대한 서비스 연결 역할](#)을 참조하십시오.

[업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.](#)

## AllocationStrategy(선택 사항, String)

[Instances](#)에 정의된 모든 컴퓨팅 리소스에 대한 할당 전략을 지정합니다.

유효한 값: lowest-price | capacity-optimized | price-capacity-optimized

기본값: lowest-price

### lowest-price

- 를 사용하는 CapacityType = ONDEMAND 경우 Amazon EC2 Fleet은 가격을 기준으로 주문을 결정하고 가장 저렴한 인스턴스를 먼저 시작합니다.
- 를 사용하는 CapacityType = SPOT 경우 Amazon EC2 플릿은 가용 용량이 있는 최저 가격의 스팟 인스턴스 풀에서 인스턴스를 시작합니다. 필요한 용량을 채우기 전에 풀의 용량이 부족한 경우 Amazon EC2 Fleet은 인스턴스를 시작하여 요청을 충족합니다. 특히 Amazon EC2 플릿은 가용 용량이 있는 최저 가격의 스팟 인스턴스 풀에서 인스턴스를 시작합니다. Amazon EC2 플릿은 여러 다른 풀에서 스팟 인스턴스를 시작할 수 있습니다.
- 설정하면 CapacityType = CAPACITY\_BLOCK 할당 전략이 없으므로 AllocationStrategy 파라미터를 구성할 수 없습니다.

### capacity-optimized

- CapacityType = ONDEMAND을 설정하면 capacity-optimized는 사용할 수 없습니다.
- 설정하면 CapacityType = SPOT Amazon EC2 플릿은 시작할 인스턴스 수에 맞게 최적의 용량을 갖춘 스팟 인스턴스 풀에서 인스턴스를 시작합니다.

### price-capacity-optimized

- CapacityType = ONDEMAND을 설정하면 capacity-optimized는 사용할 수 없습니다.
- 설정하면 CapacityType = SPOT Amazon EC2 플릿은 시작 중인 인스턴스 수 중 가장 높은 용량 가용성을 갖춘 풀을 식별합니다. 즉, 가까운 시일 내에 중단될 가능성이 가장 낮다고 판단되는 풀에서 스팟 인스턴스를 요청합니다. 그러면 Amazon EC2 플릿이 이러한 풀 중 가장 저렴한 가격의 스팟 인스턴스를 요청합니다.

[업데이트 정책](#): 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)를 설정해야 합니다.

#### Note

AllocationStrategy은 AWS ParallelCluster 버전 3.3.0부터 지원됩니다.



## JobExclusiveAllocation(선택 사항, String)

true로 설정하면 Slurm 파티션 OverSubscribe 플래그가 EXCLUSIVE로 설정됩니다. OverSubscribe=EXCLUSIVE인 경우 파티션의 작업은 할당된 모든 노드에 독점적으로 액세스할 수 있습니다. 자세한 내용은 Slurm 설명서의 [EXCLUSIVE](#)를 참조하세요.

유효한 값: true | false

기본값: false

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Note

JobExclusiveAllocation은 AWS ParallelCluster 버전 3.7.0부터 지원됩니다.

## CustomSlurmSettings(선택 사항, Dict)

사용자 지정 Slurm 파티션(대기열) 구성 설정을 정의합니다.

대기열(파티션)에 적용되는 사용자 지정 Slurm 구성 파라미터 키-값 쌍의 사전을 지정합니다.

각 개별 키-값 쌍(예: Param1: Value1)은 Slurm 파티션 구성 라인 끝에 Param1=Value1 형식으로 별도로 추가됩니다.

CustomSlurmSettings 거부 목록에 없는 Slurm 구성 파라미터만 지정할 수 있습니다. 거부 목록에 있는 Slurm 구성 파라미터에 관한 자세한 내용은 [CustomSlurmSettings을 위한 거부 목록에 등록된 Slurm 구성 파라미터](#)를 참조하세요.

AWS ParallelCluster 파라미터가 거부 목록에 있는지 여부만 확인합니다. AWS ParallelCluster 사용자 지정 Slurm 구성 매개변수 구문이나 의미 체계를 검증하지 않습니다. 사용자 지정 Slurm 구성 파라미터의 유효성을 검사하는 것은 사용자의 책임입니다. 잘못된 사용자 지정 Slurm 구성 파라미터로 인해 Slurm 대몬(daemon) 장애가 발생하여 클러스터 생성 및 업데이트 실패로 이어질 수 있습니다.

를 사용하여 사용자 지정 Slurm 구성 매개변수를 지정하는 방법에 대한 자세한 내용은 [AWS ParallelCluster참조하십시오. Slurm 구성 사용자 지정](#)

Slurm 구성 파라미터에 대한 자세한 내용은 Slurm의 [slurm.conf](#)를 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**Note**

CustomSlurmSettings은 AWS ParallelCluster 버전 3.6.0부터 지원됩니다.

**Tags**(선택 사항, [문자열])

태그 키-값 쌍의 목록입니다. [ComputeResource](#) 태그는 [Tags 섹션](#) 또는 SlurmQueues/Tags에 지정된 중복 태그보다 우선합니다.

**Key**(선택 사항, **String**)

태그 키

**Value**(선택 사항, **String**)

태그 값

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

**HealthChecks**(선택 사항)

대기열에 있는 모든 컴퓨팅 리소스에 대해 컴퓨팅 노드 상태 확인을 지정하세요.

**Gpu**(선택 사항)

대기열에 있는 모든 컴퓨팅 리소스에 대한 GPU 상태 확인을 지정하세요.

**Note**

AWS ParallelCluster alinux2ARM 운영 체제를 사용하는 Gpu 노드에서는 HealthChecks /를 지원하지 않습니다. 이러한 플랫폼은 [NVIDIA 데이터 센터 GPU 관리자\(DCGM\)](#)를 지원하지 않습니다.

**Enabled**(선택 사항, **Boolean**)

컴퓨팅 노드에서 GPU 상태 검사를 AWS ParallelCluster 수행하는지 여부. 기본값은 false입니다.

**Gpu** 상태 확인 동작

- Gpu/Enabled가 true로 설정된 경우 AWS ParallelCluster 는 대기열의 컴퓨팅 리소스에 대해 GPU 상태 확인을 수행합니다.

- Gpu 상태 확인은 컴퓨팅 리소스에서 GPU 상태 확인을 수행하여 성능이 저하된 GPU가 있는 노드에서 작업을 제출하지 못하도록 합니다.
- 컴퓨팅 노드가 Gpu 상태 확인에 실패하면 컴퓨팅 노드 상태가 DRAIN로 변경됩니다. 이 노드에서는 새 작업이 시작되지 않습니다. 기존 작업이 완료될 때까지 실행됩니다. 실행 중인 모든 작업이 완료된 후 동적 노드인 경우 컴퓨팅 노드가 종료되고 정적 노드인 경우 대체됩니다.
- Gpu 상태 확인 기간은 선택한 인스턴스 유형, 인스턴스의 GPU 수, Gpu 상태 확인 대상 수(작업 GPU 대상 수와 동일)에 따라 다릅니다. GPU가 8개인 인스턴스의 경우 일반적인 지속 시간은 3분 미만입니다.
- 지원되지 않는 인스턴스에서 Gpu가 상태 확인을 실행하면 인스턴스가 종료되고 작업이 컴퓨팅 노드에서 실행됩니다. 예를 들어 인스턴스에 GPU가 없거나 인스턴스에 GPU가 있지만 NVIDIA GPU가 아닌 경우 상태 확인이 종료되고 작업이 컴퓨팅 노드에서 실행됩니다. NVIDIA GPU만 지원됩니다.
- Gpu 상태 확인은 `dcgmi` 도구를 사용하여 노드에서 상태 확인을 수행하고 다음 단계를 수행합니다.

노드에서 Gpu 상태 확인이 시작되는 경우:

1. `nvidia-dcgm` 및 `nvidia-fabricmanager` 서비스가 실행 중인지 여부를 감지합니다.
2. 이러한 서비스가 실행되고 있지 않으면 Gpu 상태 확인이 시작됩니다.
3. 지속성 모드가 활성화되었는지 여부를 감지합니다.
4. 지속성 모드가 활성화되지 않은 경우 Gpu 상태 확인을 통해 활성화됩니다.

상태 확인이 끝나면 상태 확인은 Gpu 이러한 서비스와 리소스를 초기 상태로 복원합니다.

- 작업이 특정 노드 GPU 세트에 할당된 경우 Gpu 상태 확인은 해당 세트에서만 실행됩니다. 그렇지 않으면 노드의 모든 GPU에서 Gpu 상태 확인이 실행됩니다.
- 컴퓨팅 노드가 2개 이상의 Gpu 상태 확인 요청을 동시에 수신하면 첫 번째 상태 확인만 실행되고 나머지는 건너뛰게 됩니다. 노드 GPU를 대상으로 하는 상태 확인의 경우도 마찬가지입니다. 로그 파일에서 이 상황과 관련된 추가 정보를 확인할 수 있습니다.
- 특정 컴퓨팅 노드의 상태 확인 로그는 `/var/log/parallelcluster/slurm_health_check.log` 파일에서 확인할 수 있습니다. 이 파일은 CloudWatch Amazon의 클러스터 CloudWatch 로그 그룹에서 사용할 수 있으며, 여기에서 다음을 찾을 수 있습니다.
  - 서비스 및 지속성 모드의 활성화/비활성화를 포함하여 Gpu 상태 확인에서 실행한 작업에 대한 세부 정보
  - GPU 식별자, 시리얼 ID, UUID.

- 상태 확인 출력

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Note

HealthChecks AWS ParallelCluster 버전 3.6.0부터 지원됩니다.

## Networking

(필수) Slurm 대기열의 네트워킹 구성을 정의합니다.

```
Networking:
  SubnetIds:
    - string
  AssignPublicIp: boolean
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string
  PlacementGroup:
    Enabled: boolean
    Id: string
    Name: string
  Proxy:
    HttpProxyAddress: string
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

## Networking 속성

### SubnetIds(필수, [String])

Slurm 대기열을 프로비저닝하는 기존 서브넷의 ID

[SlurmQueues/ComputeResources/InstanceType](#)에서 인스턴스 유형을 구성하는 경우 서브넷을 하나만 정의할 수 있습니다.

[SlurmQueues/ComputeResources/Instances](#)에서 인스턴스 유형을 구성하는 경우 단일 서브넷 또는 여러 서브넷을 정의할 수 있습니다.

여러 서브넷을 사용하는 경우 대기열에 정의된 모든 서브넷은 동일한 VPC에 있어야 하며 각 서브넷은 별도의 가용 영역(AZ)에 있어야 합니다.

예를 들어 대기열에 서브넷-1과 서브넷-2를 정의한다고 가정해 보겠습니다.

subnet-1과 subnet-2 모두 AZ-1에 있을 수는 없습니다.

subnet-1은 AZ-1에 있을 수 있고 subnet-2는 AZ-2에 있을 수 있습니다.

인스턴스 유형을 하나만 구성하고 여러 서브넷을 사용하려면 InstanceType 대신에 Instances에서 인스턴스 유형을 정의하세요.

예를 들어 ComputeResources/InstanceType=instance.type 대신 ComputeResources/Instances/InstanceType=instance.type를 정의하세요.

#### Note

여러 가용 영역에 걸친 Elastic Fabric Adapter(EFA)는 지원되지 않습니다.

가용 영역을 여러 개 사용하면 스토리지 네트워킹 지연 시간이 늘어나고 AZ 간 데이터 전송 비용이 추가될 수 있습니다. 예를 들어 인스턴스가 다른 AZ에 있는 파일 스토리지에 액세스할 때 이런 현상이 발생할 수 있습니다. 자세한 내용은 [동일 AWS 리전내 데이터 전송](#)을 참조하세요.

단일 서브넷 사용에서 다중 서브넷으로 변경하기 위한 클러스터 업데이트:

- 클러스터의 서브넷 정의가 단일 서브넷과 AWS ParallelCluster 관리형 FSx for Lustre 파일 시스템으로 정의된다고 가정해 보겠습니다. 그러면 업데이트된 서브넷 ID 정의로 이 클러스터를 직접 업데이트할 수 없습니다. 클러스터를 업데이트하려면 먼저 관리 파일 시스템을 외부 파일 시스템으로 변경해야 합니다. 자세한 내용은 [AWS ParallelCluster 관리형 스토리지를 외부 스토리지로 전환합니다](#). 항목을 참조하세요.
- 추가하도록 정의된 여러 서브넷의 모든 AZ에 EFS 탑재 대상이 없는 경우 클러스터의 서브넷 정의가 단일 서브넷과 외부 Amazon EFS 파일 시스템으로 정의된다고 가정해 보겠습니다. 그러면 업데이트된 서브넷 ID 정의로 이 클러스터를 직접 업데이트할 수 없습니다. 클러스터를 업데이트하거나 클러스터를 만들려면 먼저 정의된 여러 서브넷의 모든 AZ에 대한 탑재 대상을 모두 만들어야 합니다.

가용 영역 및 클러스터 용량 예약은 다음에 정의되어 있습니다.

#### [CapacityReservationResourceGroupArn](#)

- 정의된 용량 예약 리소스 그룹에 포함되는 인스턴스 유형 및 가용 영역 집합과 대기열에 정의된 인스턴스 유형 및 가용 영역 집합 간에 중복이 없으면 클러스터를 생성할 수 없습니다.

- 정의된 용량 예약 자원 그룹에 포함되는 인스턴스 유형 및 가용 영역 집합과 대기열에 정의된 인스턴스 유형 및 가용 영역 집합 간에 부분적으로 겹치는 부분이 있는 경우 클러스터를 생성할 수 있습니다. AWS ParallelCluster 이 경우 부분 중복에 대한 경고 메시지를 보냅니다.
- 자세한 정보는 [온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작](#)을 참조하세요.

#### Note

AWS ParallelCluster 버전 3.4.0에 다중 가용 영역이 추가되었습니다.

#### Warning

이 경고는 버전 3.3.1 이전의 모든 3.x.y AWS ParallelCluster 버전에 적용됩니다. AWS ParallelCluster 이 매개변수가 변경되어도 버전 3.3.1은 영향을 받지 않습니다.

버전 3.3.1 이전 AWS ParallelCluster 3개 버전의 경우:

이 매개 변수를 변경하고 클러스터를 업데이트하면 새 관리형 FSx for Lustre 파일 시스템이 생성되고 기존 데이터를 보존하지 않고 기존 관리형 FSx for Lustre 파일 시스템이 삭제됩니다. 이로 인해 데이터가 손실됩니다. 데이터를 보존하려면 진행하기 전에 기존 FSx for Lustre 파일 시스템의 데이터를 백업해야 합니다. 자세한 내용은 FSx for Lustre 사용 설명서의 [백업 작업](#)을 참조하세요.

새 서브넷 값이 추가된 경우, [업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.](#)

서브넷 값이 제거된 경우, [업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.](#)

### AssignPublicIp(선택 사항, String)

퍼블릭 IP 주소를 생성하거나 Slurm 대기열의 노드에 할당합니다. 지원되는 값은 true 및 false입니다. 지정하는 서브넷에 따라 기본값이 결정됩니다. 퍼블릭 IP가 있는 서브넷은 기본적으로 퍼블릭 IP 주소를 할당합니다.

p4d또는 hpc6id 인스턴스 유형을 정의하거나 여러 네트워크 인터페이스 또는 네트워크 인터페이스 카드가 있는 다른 인스턴스 유형을 정의하는 경우 [HeadNodeNetworking](#)를 설정하여 퍼블릭 [ElasticIp](#)true 액세스를 제공해야 합니다. AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작한 인스턴스에만 할당할 수 있습니다. 이 경우 [NAT 게이트웨이](#)를 사용하여 클러스터 컴퓨팅 노드에 대한 퍼블릭 액세스를 제공하는 것이 좋습니다. 이 경우 AssignPublicIp를 false로 설정합

니다. IP 주소에 고나한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시 작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### SecurityGroups(선택 사항, [String])

Slurm 대기열에 사용할 보안 그룹의 목록입니다. 지정된 보안 그룹이 없는 경우 보안 그룹이 자동으로 AWS ParallelCluster 생성됩니다.

보안 그룹이 [SharedStorage](#)시스템에 맞게 구성되어 있는지 확인하십시오.

#### ⚠ Warning

이 경고는 3개 모두에 적용됩니다.  $x.y$  AWS ParallelCluster 버전 3.3.0 이전 버전 AWS ParallelCluster 이 매개변수가 변경되어도 버전 3.3.0은 영향을 받지 않습니다.

버전 3.3.0 이전의 AWS ParallelCluster 3개 버전의 경우:

이 매개 변수를 변경하고 클러스터를 업데이트하면 새 관리형 FSx for Lustre 파일 시스템 이 생성되고 기존 데이터를 보존하지 않고 기존 관리형 FSx for Lustre 파일 시스템이 삭제 됩니다. 이로 인해 데이터가 손실됩니다. 데이터를 보존하려면 기존 FSx for Lustre 파일 시스템의 데이터를 백업해야 합니다. 자세한 내용은 FSx for Lustre 사용 설명서의 [백업 작업](#)을 참조하세요.

#### ⚠ Warning

컴퓨팅 인스턴스에 Efa를 사용하도록 설정하는 경우 [EFA](#) 지원 인스턴스가 자체 내의 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹에 구성되어야 합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### AdditionalSecurityGroups(선택 사항, [String])

Slurm 대기열에 사용할 추가 보안 그룹의 목록입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### PlacementGroup(선택 사항)

Slurm 대기열의 배치 그룹 설정을 지정합니다.

[PlacementGroup](#):

Enabled: *boolean*

Id: *string*

Name: *string*

업데이트 정책: 관리형 배치 그룹 삭제를 위해서는 모든 컴퓨팅 노드를 중지해야 합니다. 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Enabled(선택 사항, **Boolean**)

Slurm대기열에 배치 그룹을 사용할지 여부를 나타냅니다. 기본값은 false입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Id(선택 사항, **String**)

Slurm 대기열이 사용하는 기존 클러스터 배치 그룹의 배치 그룹 이름 ID가 아닌 배치 그룹 이름을 입력해야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Name(선택 사항, **String**)

Slurm 대기열이 사용하는 기존 클러스터 배치 그룹의 배치 그룹 이름 ID가 아닌 배치 그룹 이름을 입력해야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

#### Note

- Name 또는 Id 정의되지 않은 상태로 PlacementGroup/Enabled가 true으로 지정된 경우, [ComputeResources/Networking/PlacementGroup](#)가 이 설정을 재정의하도록 정의되지 않는 한 각 컴퓨팅 리소스에 자체 관리형 배치 그룹이 할당됩니다.
- AWS ParallelCluster 버전 3.3.0부터 [SlurmQueues///](#)가 [NetworkingPlacementGroup//](#)의 기본 대안으로 [SlurmQueuesNetworkingPlacementGroup///](#)가 [Name](#)추가되었습니다. [Id PlacementGroup/Id](#)와 [PlacementGroup/Name](#)는 동일합니다. 어느 것이든 사용할 수 있습니다.



[PlacementGroupId](#)/와 [PlacementGroup](#)/를 모두 [Name](#)포함하면 AWS ParallelCluster 실패합니다. 둘 중 하나만 선택할 수 있습니다.

[PlacementGroup/Name](#)를 사용하기 위해 클러스터를 업데이트할 필요는 없습니다.

## Proxy(선택 사항)

Slurm 대기열의 프록시 설정을 지정합니다.

[Proxy](#):

[HttpProxyAddress](#): *string*

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)를 설정해야 합니다.

### HttpProxyAddress(선택 사항, String)

Slurm 대기열의 HTTP 또는 HTTPS 프록시 서버를 정의합니다. 보통은 `https://x.x.x.x:8080`입니다.

기본값이 없습니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)를 설정해야 합니다.

## Image

(선택 사항) Slurm 대기열에 사용할 이미지를 지정합니다. 모든 노드에 동일한 AMI를 사용하려면 [Image](#) 섹션의 [CustomAmi](#) 설정을 사용하십시오.

[Image](#):

[CustomAmi](#): *string*

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 [QueueUpdateStrategy](#)를 설정해야 합니다.

## Image 속성

### CustomAmi(선택 사항, String)

기본 AMI 대신 Slurm 대기열에 사용할 AMI입니다. pcluster CLI 명령을 사용하여 기본 AMI 목록을 볼 수 있습니다.

#### Note

AMI는 헤드 노드에서 사용하는 것과 동일한 운영 체제를 기반으로 해야 합니다.

#### `pcluster list-official-images`

사용자 지정 AMI를 시작하기 위해 추가 권한이 필요한 경우 헤드 노드 정책에 이러한 권한을 추가해야 합니다.

예를 들어, 사용자 지정 AMI에 암호화된 스냅샷이 연결된 경우 헤드 노드 정책에 다음과 같은 추가 정책이 필요합니다.

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

사용자 지정 AMI 검증 경고 문제를 해결하려면 [사용자 지정 AMI 문제 해결](#)을 참조하세요.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

## ComputeResources

(필수) Slurm 대기열의 ComputeResources 구성을 정의합니다.

### Note

클러스터 크기는 업데이트 중에 변경될 수 있습니다. 자세한 내용은 [클러스터 용량 크기 및 업데이트를 참조하십시오](#).

#### ComputeResources:

- Name: *string*
- InstanceType: *string*
- Instances:
  - InstanceType: *string*
- MinCount: *integer*
- MaxCount: *integer*
- DynamicNodePriority: *integer*
- StaticNodePriority: *integer*
- SpotPrice: *float*
- DisableSimultaneousMultithreading: *boolean*
- SchedulableMemory: *integer*
- HealthChecks:
  - Gpu:
    - Enabled: *boolean*
- Efa:
  - Enabled: *boolean*
  - GdrSupport: *boolean*
- CapacityReservationTarget:
  - CapacityReservationId: *string*
  - CapacityReservationResourceGroupArn: *string*
- Networking:
  - PlacementGroup:
    - Enabled: *boolean*
    - Name: *string*
- CustomSlurmSettings: *dict*
- Tags:
  - Key: *string*

**Value:** *string*

업데이트 정책: 이 목록 값 설정의 경우 업데이트 중에 새 값을 추가할 수 있으며, 또는 기존 값을 제거 할 때 컴퓨팅 플릿을 중지해야 합니다.

## ComputeResources 속성

### Name(필수, String)

Slurm 대기열 컴퓨팅 환경의 이름 계획 이름은 최대 25자까지 가능합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### InstanceType(필수, String)

이 Slurm 컴퓨팅 리소스에 사용되는 인스턴스 유형입니다. 클러스터의 모든 인스턴스 유형은 동일한 프로세서 아키텍처를 사용해야 합니다. 인스턴스는 x86\_64 또는 arm64 아키텍처 중 하나를 사용할 수 있습니다.

클러스터 구성은 [InstanceType](#) 또는 [인스턴스](#)를 정의해야 합니다. 둘 다 정의하면 AWS ParallelCluster 실패합니다.

InstanceType를 정의할 때는 여러 서브넷을 정의할 수 없습니다. 인스턴스 유형을 하나만 구성하고 여러 서브넷을 사용하려면 InstanceType 대신에 Instances에서 인스턴스 유형을 정의하세요. 자세한 내용은 [Networking/SubnetIds](#) 항목을 참조하세요.

p4d 또는 hpc6id 인스턴스 유형이나 여러 네트워크 인터페이스 또는 네트워크 인터페이스 카드가 있는 다른 인스턴스 유형을 정의하는 경우에 설명된 대로 프라이빗 서브넷에서 컴퓨팅 인스턴스를 시작해야 합니다. [AWS ParallelCluster 서브넷 2개 사용](#) AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작된 인스턴스에만 할당할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### Instances(필수)

컴퓨팅 리소스의 인스턴스 유형 목록을 지정합니다. 인스턴스 유형 목록에 대한 할당 전략을 지정하려면 [AllocationStrategy](#)을 참조하세요.

클러스터 구성은 [InstanceType](#) 또는 [Instances](#) 중 하나를 정의해야 합니다. 둘 다 정의하면 AWS ParallelCluster 은 실패합니다.

자세한 정보는 [Slurm을 사용하여 여러 인스턴스 유형 할당](#)을 참조하세요.

Instances:

- InstanceType: *string*

**i** Note

AWS ParallelCluster 버전 3.7.0부터 인스턴스에서 여러 인스턴스 유형을 구성하면 활성화할 EnableMemoryBasedScheduling 수 있습니다.

AWS ParallelCluster 버전 3.2.0~3.6의 경우. x, 인스턴스에서 여러 인스턴스 유형을 구성한 경우 활성화할 EnableMemoryBasedScheduling 수 없습니다.

업데이트 정책: 이 목록 값 설정의 경우 업데이트 중에 새 값을 추가할 수 있으며, 또는 기존 값을 제거할 때 컴퓨팅 플릿을 중지해야 합니다.

**InstanceType(필수, String)**

이 Slurm 컴퓨팅 리소스에 사용할 인스턴스 유형입니다. 클러스터의 모든 인스턴스 유형은 동일한 프로세서 아키텍처, x86\_64 또는 arm64를 사용해야 합니다.

Instances에 나열된 인스턴스 유형은 다음을 가지고 있어야 합니다.

- 동일한 수의 vCPU. DisableSimultaneousMultithreading이 true으로 설정되었다면 동일한 수의 코어.
- 동일한 제조사의 동일한 수의 액셀러레이터
- Efa/Enabled이 true로 설정되어 있으면 EFA 지원

Instances 목록에 있는 인스턴스 유형은 다음을 가지고 있을 수 있습니다:

- 메모리 용량이 다릅니다.

이 경우 최소 메모리를 사용 가능한 Slurm 리소스로 설정해야 합니다.

**i** Note

AWS ParallelCluster 버전 3.7.0부터 인스턴스에서 여러 인스턴스 유형을 구성하면 EnableMemoryBasedScheduling 활성화할 수 있습니다.

AWS ParallelCluster 버전 3.2.0~3.6의 경우. x, 인스턴스에서 여러 인스턴스 유형을 구성한 경우 활성화할 EnableMemoryBasedScheduling 수 없습니다.

- 다른 네트워크 카드

이 경우 컴퓨팅 리소스에 구성된 네트워크 인터페이스 수는 네트워크 카드 수가 가장 적은 인스턴스 유형에 따라 정의됩니다.

- 네트워크 대역폭 차이
- 다양한 인스턴스 스토어 크기

p4d 또는 hpc6id 인스턴스 유형이나 여러 네트워크 인터페이스 또는 네트워크 인터페이스 카드가 있는 다른 인스턴스 유형을 정의하는 경우에 설명된 대로 프라이빗 서브넷에서 컴퓨팅 인스턴스를 시작해야 합니다. [AWS ParallelCluster 서브넷 2개 사용](#) AWS 퍼블릭 IP는 단일 네트워크 인터페이스로 시작한 인스턴스에만 할당할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 중 퍼블릭 IPv4 주소 할당](#)을 참조하세요.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

#### Note

Instances AWS ParallelCluster 버전 3.3.0부터 지원됩니다.

### MinCount(선택 사항, Integer)

Slurm 컴퓨팅 리소스가 사용하는 최소 인스턴스 수입니다. 기본값은 0입니다.

#### Note

업데이트 중에 클러스터 크기가 변경될 수 있습니다. 자세한 내용은 [클러스터 용량 크기 및 업데이트](#)를 참조하십시오.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### MaxCount(선택 사항, Integer)

Slurm 컴퓨팅 리소스가 사용하는 최대 인스턴스 수입니다. 기본값은 10.

용량 블록 예약의 모든 인스턴스 부분이 정적 노드로 관리되므로 사용할 CapacityType = CAPACITY\_BLOCK 때는 0보다 크거나 MaxCount 같아야 합니다. MinCount

클러스터 생성 시 헤드 노드는 클러스터 생성 성공 신호를 보내기 전에 모든 정적 노드가 준비될 때까지 기다립니다. 하지만 사용 시 CapacityType = CAPACITY\_BLOCK Capacity Blocks와 관련

된 컴퓨팅 리소스 중 노드 부분은 이 검사 대상에서 제외됩니다. 구성된 모든 용량 블록이 활성 상태가 아니더라도 클러스터는 생성됩니다.

**Note**

업데이트 중에 클러스터 크기가 변경될 수 있습니다. 자세한 내용은 [클러스터 용량 크기 및 업데이트](#)를 참조하십시오.

### DynamicNodePriority(선택 사항, Integer)

대기열 컴퓨팅 리소스에 있는 동적 노드의 우선 순위 우선 순위는 컴퓨팅 리소스 동적 Slurm 노드의 노드 [Weight](#) 구성 파라미터에 매핑됩니다. 기본값은 1000입니다.

Slurm은 Weight 값이 가장 낮은 노드부터 우선 순위를 지정합니다.

**Warning**

Slurm 파티션(대기열)에 여러 Weight 값을 사용하면 대기열의 작업 예약 속도가 느려질 수 있습니다.

버전 3.7.0 이전 AWS ParallelCluster 버전에서는 정적 노드와 동적 노드 모두에 동일한 기본 가중치가 할당되었습니다. 1 이 경우 정적 및 동적 노드의 이름 지정 스키마로 인해 Slurm가 유틸리티 정적 노드보다 유틸리티 동적 노드를 우선할 수 있습니다. 다른 모든 조건이 같으면 Slurm는 노드 이름을 알파벳순으로 스케줄링합니다.

**Note**

DynamicNodePriority AWS ParallelCluster 버전 3.7.0에 추가되었습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### StaticNodePriority(선택 사항, Integer)

대기열 컴퓨팅 리소스에 있는 정적 노드의 우선 순위 우선 순위는 컴퓨팅 리소스 정적 Slurm 노드 [Weight](#) 구성 파라미터에 매핑됩니다. 기본값은 1입니다.

Slurm은 Weight 값이 가장 낮은 노드부터 우선 순위를 지정합니다.

**⚠ Warning**

Slurm 파티션(대기열)에 여러 Weight 값을 사용하면 대기열의 작업 예약 속도가 느려질 수 있습니다.

**i Note**

StaticNodePriority AWS ParallelCluster 버전 3.7.0에 추가되었습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**SpotPrice(선택 사항, Float)**

인스턴스가 시작되기 전에 Amazon EC2 스팟 인스턴스에 지불한 최고 가격입니다. 기본값은 온디맨드 요금입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

**DisableSimultaneousMultithreading(선택 사항, Boolean)**

true이면 Slurm 대기열에 있는 노드의 멀티스레딩이 비활성화됩니다. 기본값은 false입니다.

일부 인스턴스 유형은 멀티스레딩을 비활성화할 수 없습니다. 멀티스레딩 비활성화를 지원하는 인스턴스 유형 목록은 Amazon EC2 사용 설명서에서 [인스턴스 유형별 각 CPU 코어의 CPU 코어 및 스레드를 참조하십시오.](#)

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

**SchedulableMemory(선택 사항, Integer)**

컴퓨팅 리소스의 컴퓨팅 노드에 대한 Slurm 파라미터 RealMemory에 구성된 메모리 양(MiB) 이 값은 [SlurmSettings/EnableMemoryBasedScheduling](#)가 활성화된 경우 작업에 사용할 수 있는 노드 메모리의 상한값입니다. 기본값은 Amazon [EC2 인스턴스 유형에 나열되어 있고 Amazon EC2 API에서](#) 반환되는 메모리의 95%입니다. [DescribeInstanceTypes](#) GiB 단위로 지정된 값을 MiB로 변환해야 합니다.

지원되는 값: 1-EC2Memory

EC2Memory Amazon EC2 [인스턴스 유형에 나열되고 Amazon EC2 API에서](#) 반환되는 메모리 (MiB 단위)입니다. [DescribeInstanceTypes](#) GiB 단위로 지정된 값을 MiB로 변환해야 합니다.



이 옵션은 [SlurmSettings/EnableMemoryBasedScheduling](#)가 활성화된 경우에 가장 적합합니다. 자세한 내용은 [Slurm 메모리 기반 스케줄링](#) 항목을 참조하세요.

#### Note

SchedulableMemory은 AWS ParallelCluster 버전 3.2.0부터 지원됩니다. 버전 3.2.0부터 기본적으로 Amazon EC2 API에서 반환되는 메모리의 95% 까지 Slurm 컴퓨팅 노드를 AWS ParallelCluster 구성합니다RealMemory. DescribeInstanceTypes 이 구성은 EnableMemoryBasedScheduling의 값과는 무관합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### HealthChecks(선택 사항)

컴퓨팅 리소스의 상태 확인을 지정하세요.

#### Gpu(선택 사항)

컴퓨팅 리소스에 대한 GPU 상태 확인을 지정합니다.

#### Enabled(선택 사항, Boolean)

대기열에 있는 컴퓨팅 리소스에 대해 GPU 상태 검사를 AWS ParallelCluster 수행하는지 여부. 기본값은 false입니다.

#### Note

AWS ParallelCluster alinux2ARM 운영 체제를 사용하는 Gpu 노드에서는 HealthChecks /를 지원하지 않습니다. 이러한 플랫폼은 [NVIDIA 데이터 센터 GPU 관리자\(DCGM\)](#)를 지원하지 않습니다.

### Gpu 상태 확인 동작

- GpuEnabled/가 로 true 설정된 경우 컴퓨팅 리소스에서 GPU 상태 점검을 AWS ParallelCluster 수행합니다.
- Gpu 상태 확인은 컴퓨팅 리소스에서 상태 확인을 수행하여 성능이 저하된 GPU가 있는 노드에서 작업을 제출하지 못하도록 합니다.

- 컴퓨팅 노드가 Gpu 상태 확인에 실패하면 컴퓨팅 노드 상태가 DRAIN로 변경됩니다. 이 노드에서는 새 작업이 시작되지 않습니다. 기존 작업이 완료될 때까지 실행됩니다. 실행 중인 모든 작업이 완료된 후 동적 노드인 경우 컴퓨팅 노드가 종료되고 정적 노드인 경우 대체됩니다.
- Gpu 상태 확인 기간은 선택한 인스턴스 유형, 인스턴스의 GPU 수, Gpu 상태 확인 대상 수(작업 GPU 대상 수와 동일)에 따라 다릅니다. GPU가 8개인 인스턴스의 경우 일반적인 지속 시간은 3분 미만입니다.
- 지원되지 않는 인스턴스에서 Gpu가 상태 확인을 실행하면 인스턴스가 종료되고 작업이 컴퓨팅 노드에서 실행됩니다. 예를 들어 인스턴스에 GPU가 없거나 인스턴스에 GPU가 있지만 NVIDIA GPU가 아닌 경우 상태 확인이 종료되고 작업이 컴퓨팅 노드에서 실행됩니다. NVIDIA GPU만 지원됩니다.
- Gpu 상태 확인은 `dcgmi` 도구를 사용하여 노드에서 상태 확인을 수행하고 다음 단계를 수행합니다.

노드에서 Gpu 상태 확인이 시작되는 경우:

1. `nvidia-dcgm` 및 `nvidia-fabricmanager` 서비스가 실행 중인지 여부를 감지합니다.
2. 이러한 서비스가 실행되고 있지 않으면 Gpu 상태 확인이 시작됩니다.
3. 지속성 모드가 활성화되었는지 여부를 감지합니다.
4. 지속성 모드가 활성화되지 않은 경우 Gpu 상태 확인을 통해 활성화됩니다.

상태 확인이 끝나면 상태 확인은 Gpu 이러한 서비스와 리소스를 초기 상태로 복원합니다.

- 작업이 특정 노드 GPU 세트에 할당된 경우 Gpu 상태 확인은 해당 세트에서만 실행됩니다. 그렇지 않으면 노드의 모든 GPU에서 Gpu 상태 확인이 실행됩니다.
- 컴퓨팅 노드가 2개 이상의 Gpu 상태 확인 요청을 동시에 수신하면 첫 번째 상태 확인만 실행되고 나머지는 건너뛰게 됩니다. 노드 GPU를 대상으로 하는 상태 확인의 경우도 마찬가지입니다. 로그 파일에서 이 상황과 관련된 추가 정보를 확인할 수 있습니다.
- 특정 컴퓨팅 노드의 상태 확인 로그는 `/var/log/parallelcluster/slurm_health_check.log` 파일에서 확인할 수 있습니다. 이 파일은 CloudWatch Amazon의 클러스터 CloudWatch 로그 그룹에서 사용할 수 있으며, 여기에서 다음을 찾을 수 있습니다.
  - 서비스 및 지속성 모드의 활성화/비활성화를 포함하여 Gpu 상태 확인에서 실행한 작업에 대한 세부 정보
  - GPU 식별자, 시리얼 ID, UUID.
  - 상태 확인 출력

[업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.](#)

**Note**

HealthChecks AWS ParallelCluster 버전 3.6.0부터 지원됩니다.

**Efa(선택 사항)**

Slurm 대기열에 있는 노드의 Elastic Fabric Adapter(EFA) 설정을 지정합니다.

**Efa:**

**Enabled:** *boolean*

**GdrSupport:** *boolean*

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

**Enabled(선택 사항, Boolean)**

Elastic Fabric Adapter(EFA)를 활성화하도록 지정합니다. EFA를 지원하는 Amazon EC2 인스턴스 목록을 보려면 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [지원되는 인스턴스 유형](#)을 참조하십시오. 자세한 정보는 [Elastic Fabric Adapter](#)을 참조하세요. 클러스터 [SlurmQueues/Networking/PlacementGroup](#)를 사용하여 인스턴스 간 지연 시간을 최소화 하는 것이 좋습니다.

기본값은 false입니다.

**Note**

여러 가용 영역에 걸친 Elastic Fabric Adapter(EFA)는 지원되지 않습니다. 자세한 내용은 [SubnetIds](#)를 참조하세요.

**Warning**

에서 [SecurityGroups](#)사용자 지정 보안 그룹을 정의하는 경우 EFA 지원 인스턴스가 자체적으로 모든 인바운드 및 아웃바운드 트래픽을 허용하는 보안 그룹의 구성원인지 확인하십시오.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### GdrSupport(선택 사항, Boolean)

(선택 사항) AWS ParallelCluster 버전 3.0.2부터는 이 설정이 적용되지 않습니다. Slurm 컴퓨팅 리소스 및 운영 체제의 인스턴스 유형에서 지원되는 경우 GPUDirect RDMA(remote direct memory access)에 대한 Elastic Fabric Adapter(EFA) 지원이 항상 활성화됩니다.

#### Note

AWS ParallelCluster 버전 3.0.0에서 3.0.1까지: 컴퓨팅 리소스에 대한 GPUDirect RDMA 지원이 활성화되었습니다. Slurm EFA는 특정 운영 체제(OS가 alinux2, centos7, ubuntu1804, 또 ubuntu2004)의 특정 인스턴스 유형(p4d.24xlarge)에서 지원됩니다. 기본값은 false입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 풀릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### CapacityReservationTarget

CapacityReservationTarget:

CapacityReservationId: *string*

CapacityReservationResourceGroupArn: *string*

컴퓨팅 리소스에 사용할 온디맨드 용량 예약을 지정합니다.

### CapacityReservationId(선택 사항, String)

대기열의 컴퓨팅 리소스를 대상으로 하는 기존 용량 예약의 ID입니다. ID는 ML용 ODCR 또는 용량 블록을 참조할 수 있습니다.

이 매개 변수를 컴퓨팅 리소스 수준에서 지정하는 InstanceType 것이 선택 사항인 경우 예약에서 자동으로 검색됩니다.

### CapacityReservationResourceGroupArn(선택 사항, String)

컴퓨팅 리소스의 서비스 연결 용량 예약 그룹 역할을 하는 리소스 그룹에 Amazon 리소스 이름 (ARN)을 나타냅니다. AWS ParallelCluster 는 그룹에서 가장 적절한 용량 예약을 식별하여 사용합니다. 리소스 그룹에는 컴퓨팅 리소스에 대해 나열된 각 인스턴스 유형에 대해 하나 이상의 ODCR이 있어야 합니다. 자세한 정보는 온디맨드 용량 예약 (ODCR) 으로 인스턴스 시작을 참조하세요.

- [SlurmQueuesSlurmQueuesComputeResources/NetworkingNetworking](#) 또는 //에서 활성화된 경우 PlacementGroup 인스턴스 유형을 대상으로 하는 리소스 그룹을 AWS ParallelCluster 선택하고 컴퓨팅 리소스 (있는 경우) 를 PlacementGroup 대상으로 합니다.

PlacementGroup는 [ComputeResources](#)에 정의된 인스턴스 유형 중 하나를 대상으로 해야 합니다.

- [SlurmQueuesSlurmQueuesComputeResources/Networking](#) 또는 //에서 PlacementGroup 활성화되지 않은 경우 [Networking](#), 컴퓨팅 리소스의 인스턴스 유형만 대상으로 하는 리소스 그룹 (있는 경우) 을 AWS ParallelCluster 선택합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

#### Note

CapacityReservationTarget이 AWS ParallelCluster 버전 3.3.0에 추가되었습니다.

## Networking

[Networking](#):

[PlacementGroup](#):

[Enabled](#): *boolean*

[Name](#): *string*

업데이트 정책: 관리형 배치 그룹 삭제를 위해서는 모든 컴퓨팅 노드를 중지해야 합니다. 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### PlacementGroup(선택 사항)

컴퓨팅 리소스의 배치 그룹 설정을 지정합니다.

#### Enabled(선택 사항, Boolean)

컴퓨팅 리소스에 배치 그룹을 사용할지 여부를 나타냅니다.

- Name가 정의되지 않은 상태로 true로 설정하면 [SlurmQueues/Networking/PlacementGroup](#) 설정에 관계없이 해당 컴퓨팅 리소스에 자체 관리형 배치 그룹이 할당됩니다.

- Name가 정의된 상태로 true로 설정하면 SlurmQueues/Networking/PlacementGroup 설정에 관계없이 해당 컴퓨팅 리소스에 이름이 지정된 배치 그룹이 할당됩니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Name(선택 사항, String)

컴퓨팅 리소스에 사용되는 기존 클러스터 배치 그룹의 배치 그룹 이름입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

#### Note

- PlacementGroup/Enabled 및 Name가 모두 설정되지 않은 경우 각 값이 [SlurmQueues/Networking/PlacementGroup](#)로 디폴트됩니다.
- ComputeResourcesNetworkingPlacementGroup//는 AWS ParallelCluster 버전 3.3.0에 추가되었습니다.

### CustomSlurmSettings(선택 사항, Dict)

(선택 사항) 사용자 지정 Slurm 노드(컴퓨팅 리소스) 구성 설정을 정의합니다.

Slurm 노드(컴퓨팅 리소스)에 적용되는 사용자 지정 Slurm 구성 파라미터 키-값 쌍의 사전을 지정합니다.

각 개별 키-값 쌍(예: Param1: Value1)은 Slurm 노드 구성 라인 끝에 Param1=Value1 형식으로 별도로 추가됩니다.

CustomSlurmSettings 거부 목록에 없는 Slurm 구성 파라미터만 지정할 수 있습니다. 거부 목록에 있는 Slurm 구성 파라미터에 관한 자세한 내용은 [CustomSlurmSettings을 위한 거부 목록에 등록된 Slurm 구성 파라미터](#)를 참조하세요.

AWS ParallelCluster 매개 변수가 거부 목록에 있는지 여부만 확인합니다. AWS ParallelCluster 사용자 지정 Slurm 구성 매개변수 구문이나 의미 체계를 검증하지 않습니다. 사용자 지정 Slurm 구성 파라미터의 유효성을 검사하는 것은 사용자의 책임입니다. 잘못된 사용자 지정 Slurm 구성 파라미터로 인해 Slurm 대몬(daemon) 장애가 발생하여 클러스터 생성 및 업데이트 실패로 이어질 수 있습니다.

를 사용하여 사용자 지정 Slurm 구성 매개변수를 지정하는 방법에 대한 자세한 내용은 [AWS ParallelCluster](#) 참조하십시오. [Slurm 구성 사용자 지정](#)

Slurm 구성 파라미터에 대한 자세한 내용은 Slurm의 [slurm.conf](#)를 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### Note

CustomSlurmSettings은 AWS ParallelCluster 버전 3.6.0부터 지원됩니다.

### Tags(선택 사항, [문자열])

태그 키-값 쌍의 목록입니다. ComputeResource 태그는 [Tags 섹션](#) 또는 [SlurmQueues/Tags](#)에 지정된 중복 태그보다 우선합니다.

#### Key(선택 사항, **String**)

태그 키

#### Value(선택 사항, **String**)

태그 값

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### ComputeSettings

(필수) Slurm 대기열의 ComputeSettings 구성을 정의합니다.

#### ComputeSettings 속성

Slurm 대기열에 있는 ComputeSettings 노드의 속성을 지정합니다.

```

ComputeSettings:
  LocalStorage:
    RootVolume:
      Size: integer
      Encrypted: boolean
      VolumeType: string
      Iops: integer
  
```

```
Throughput: integer
EphemeralVolume:
MountDir: string
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### LocalStorage(선택 사항)

Slurm 대기열에 있는 LocalStorage 노드의 속성을 지정합니다.

```
LocalStorage:
RootVolume:
Size: integer
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
EphemeralVolume:
MountDir: string
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### RootVolume(선택 사항)

Slurm 대기열에 있는 노드의 루트 볼륨 세부 정보를 지정합니다.

```
RootVolume:
Size: integer
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Size(선택 사항, Integer)

Slurm 대기열에 있는 노드의 루트 볼륨 크기를 기비바이트(GiB) 단위로 지정합니다. 기본 크기는 AMI에서 가져옵니다. 다른 크기를 사용하려면 AMI에서 growroot를 지원해야 합니다.



업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Encrypted(선택 사항, Boolean)

true 경우 Slurm 대기열에 있는 노드의 루트 볼륨이 암호화됩니다. 기본값은 false입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### VolumeType(선택 사항, String)

Slurm 대기열에 있는 노드의 [Amazon EBS 볼륨 유형](#)을 지정합니다. 지원되는 값은 gp2, gp3, io1, io2, sc1, st1, standard입니다. 기본값은 gp3입니다.

자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 [Amazon EBS 볼륨 유형](#)을 참조하세요.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Iops(선택 사항, Boolean)

io1, io2, gp3 유형 볼륨의 IOPS 수를 정의합니다.

기본 값, 지원되는 값, volume\_iops:volume\_size 비율은 VolumeType 및 Size에 따라 달라집니다.

#### VolumeType = io1

기본 Iops = 100

지원되는 값 Iops = 100~64000+

최대 volume\_iops와 volume\_size의 비율 = 50IOPS/GiB. 5000 IOPS는 최소 100GiB의 volume\_size가 필요합니다.

#### VolumeType = io2

기본 Iops = 100

지원되는 값 Iops = 100~64000(io2 Block Express 볼륨의 경우 256000)+

최대 Iops와 Size 비율 = 500IOPS/GiB. 5000 IOPS에는 최소 10GiB의 Size가 필요합니다.

**VolumeType = gp3**

기본 Iops = 3000

지원되는 값 Iops = 3000-16000 †

최대 Iops:Size 비율 = IOPS가 3000보다 큰 볼륨의 경우 GiB당 500 IOPS입니다.

† 최대 IOPS는 32,000 IOPS 이상으로 프로비저닝된 [Nitro 시스템에 구축된 인스턴스에서](#)만 보장됩니다. 다른 인스턴스는 최대 32,000 IOPS를 가질 수 있습니다. 이전 io1 볼륨은 [볼륨을 수정](#)하지 않는 한 전체 성능에 도달할 수 없습니다. io2 블록 익스프레스 볼륨은 R5b 인스턴스 유형에서 최대 256000의 volume\_iops 값을 지원합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [io2 블록 익스프레스 볼륨](#)을 참조하십시오.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

**Throughput(선택 사항, Integer)**

gp3 볼륨 유형의 처리량을 MiB/s 단위로 정의합니다. 이 설정은 VolumeType가 gp3일 때만 유효합니다. 기본값은 125입니다. 지원되는 값: 125-1000MiB/s

Throughput:Iops의 비율은 0.25를 초과할 수 없습니다. 1000MiB/s의 최대 처리량을 위해서는 Iops 설정이 최소 4000이어야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

**EphemeralVolume(선택 사항, Boolean)**

임시 볼륨의 설정을 지정합니다. 임시 볼륨은 모든 인스턴스 스토어 볼륨을 ext4 파일 시스템으로 포맷된 단일 논리 볼륨으로 결합하여 생성됩니다. 기본값은 /scratch입니다. 인스턴스 스토어 볼륨이 없는 인스턴스 유형의 경우 임시 볼륨이 생성되지 않습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 스토어 볼륨](#)을 참조하세요.

EphemeralVolume:  
MountDir: *string*

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

## MountDir(선택 사항, String)

Slurm 대기열에 있는 각 노드에 대한 임시 볼륨의 탑재 디렉터리입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

## CustomActions

(선택 사항) Slurm 대기열 내 노드상에서 실행할 사용자 지정 스크립트를 지정합니다.

```
CustomActions:
  OnNodeStart:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
  OnNodeConfigured:
    Sequence:
      - Script: string
        Args:
          - string
    Script: string
    Args:
      - string
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

## CustomActions 속성

### OnNodeStart(선택 사항, String)

노드 배포 부트스트랩 작업이 시작되기 전에 Slurm 대기열의 노드에서 실행할 스크립트 시퀀스 또는 단일 스크립트를 지정합니다. AWS ParallelCluster 는 동일한 사용자 지정 작업에 단일 스크립트와 Sequence를 모두 포함하는 것은 지원하지 않습니다. 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

## Sequence(선택 사항)

실행할 스크립트 목록

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Script(필수, String)

사용할 파일 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Args(선택 사항, [String])

스크립트에 전달할 인수 목록

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Script(필수, String)

단일 스크립트에 사용할 파일 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Args(선택 사항, [String])

단일 스크립트에 전달할 인수 목록

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

## OnNodeConfigured(선택 사항, String)

노드 부트스트랩 작업이 모두 완료된 후 Slurm 대기열의 노드에서 실행할 스크립트 시퀀스 또는 단일 스크립트를 지정합니다. AWS ParallelCluster 는 동일한 사용자 지정 작업에 단일 스크립트와 Sequence를 모두 포함하는 것은 지원하지 않습니다. 자세한 내용은 [사용자 지정 부트스트랩 작업 항목](#)을 참조하세요.

## Sequence(선택 사항)

실행할 스크립트 목록

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Script(필수, String)

사용할 파일 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Args(선택 사항, [String])

스크립트에 전달할 인수 목록

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Script(필수, String)

단일 스크립트에 사용할 파일 파일 경로는 https:// 또는 s3://로 시작되어야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

### Args(선택 사항, [String])

단일 스크립트에 전달할 인수 목록

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

#### Note

Sequence AWS ParallelCluster 버전 3.6.0부터 추가되었습니다. 지정하는 Sequence 경우 사용자 지정 작업에 사용할 여러 스크립트를 나열할 수 있습니다. AWS ParallelCluster 스크립트는 포함하지 않고 단일 스크립트로 사용자 정의 액션을 구성할 수 있도록 계속 지원합니다Sequence.

AWS ParallelCluster 단일 스크립트와 Sequence 동일한 사용자 지정 작업을 모두 포함하는 것은 지원하지 않습니다.

## Iam

(선택 사항) Slurm 대기열의 선택적 IAM 설정을 정의합니다.

```
Iam:
  S3Access:
    - BucketName: string
      EnableWriteAccess: boolean
      KeyName: string
  AdditionalIamPolicies:
    - Policy: string
  InstanceProfile: string
  InstanceRole: string
```

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## Iam 속성

### InstanceProfile(선택 사항, String)

Slurm 대기열의 기본 인스턴스 역할 또는 인스턴스 프로파일을 재정의할 인스턴스 프로파일을 지정합니다. InstanceProfile 및 InstanceRole를 둘 다 지정할 수 없습니다. 형식은 `arn:${Partition}:iam:${Account}:instance-profile/${InstanceProfileName}`입니다.

이를 지정하면 S3Access 및 AdditionalIamPolicies 설정을 지정할 수 없습니다.

S3Access에 추가된 특성은 새로운 권한을 요구하는 경우가 많으므로 AdditionalIamPolicies 및 AWS ParallelCluster 중 하나 또는 모두를 지정하는 것이 좋습니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### InstanceRole(선택 사항, String)

Slurm 대기열의 기본 인스턴스 역할 또는 인스턴스 프로파일을 재정의할 인스턴스 역할을 지정합니다. InstanceProfile 및 InstanceRole를 둘 다 지정할 수 없습니다. 형식은 `arn:${Partition}:iam:${Account}:role/${RoleName}`입니다.

이를 지정하면 S3Access 및 AdditionalIamPolicies 설정을 지정할 수 없습니다.

S3Access에 추가된 특성은 새로운 권한을 요구하는 경우가 많으므로 AdditionalIamPolicies 및 AWS ParallelCluster 중 하나 또는 모두를 지정하는 것이 좋습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### S3Access(선택 사항)

Slurm 대기열의 버킷을 지정합니다. 이는 Slurm 대기열의 버킷에 지정된 액세스 권한을 부여하는 정책을 생성하는 데 사용됩니다.

이를 지정하면 InstanceProfile 및 InstanceRole 설정을 지정할 수 없습니다.

S3Access에 추가된 특성은 새로운 권한을 요구하는 경우가 많으므로 AdditionalIamPolicies 및 AWS ParallelCluster 중 하나 또는 모두를 지정하는 것이 좋습니다.

#### S3Access:

- `BucketName`: *string*
- `EnableWriteAccess`: *boolean*
- `KeyName`: *string*

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### BucketName(필수, String)

버킷의 이름입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### KeyName(선택 사항, String)

버킷의 키입니다. 기본값은 \*입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### EnableWriteAccess(선택 사항, Boolean)

버킷에 대해 쓰기 액세스가 활성화되어 있는지 여부를 나타냅니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### AdditionalIamPolicies(선택 사항)

Amazon EC2에 대한 IAM 정책의 Amazon 리소스 이름(ARN) 목록을 쉼표로 구분하여 지정합니다. 이 목록은 에서 요구하는 권한 외에도 Slurm 대기열에 사용되는 루트 역할에 첨부됩니다 AWS ParallelCluster.

IAM 정책 이름과 해당 ARN은 서로 다릅니다. 이름은 사용할 수 없습니다.

이를 지정하면 InstanceProfile 및 InstanceRole 설정을 지정할 수 없습니다.

AdditionalIamPolicies를 사용하는 것이 좋습니다. AWS ParallelCluster가 요구하는 권한에 AdditionalIamPolicies가 추가되며, InstanceRole에는 요구되는 권한이 모두 포함되어 있어야 하기 때문입니다. 기능이 추가됨에 따라 필요한 권한은 종종 릴리스마다 변경됩니다.

기본값이 없습니다.

#### AdditionalIamPolicies:

- Policy: *string*

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### **Policy(필수, [String])**

IAM 정책 목록

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## SlurmSettings

(선택 사항) 전체 클러스터에 적용되는 Slurm 설정을 정의합니다.

#### SlurmSettings:

ScaledownIdleTime: *integer*  
QueueUpdateStrategy: *string*  
EnableMemoryBasedScheduling: *boolean*  
CustomSlurmSettings: [*dict*]  
CustomSlurmSettingsIncludeFile: *string*  
Database:  
Uri: *string*  
UserName: *string*  
PasswordSecretArn: *string*  
ExternalSlurmdbd:  
Host: *string*  
Port: *integer*  
Dns:  
DisableManagedDns: *boolean*  
HostedZoneId: *string*  
UseEc2Hostnames: *boolean*



## SlurmSettings 속성

### ScaledownIdleTime(선택 사항, Integer)

작업이 없고 Slurm 노드가 종료되는 시간(분)을 정의합니다.

기본값은 10입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### MungeKeySecretArn(선택 사항, String)

클러스터에서 사용할 base64로 인코딩된 먼지 키가 포함된 일반 텍스트 Secrets AWS Manager 시크릿의 Amazon 리소스 이름 (ARN)입니다. Slurm 이 Munge 키는 Slurm 클라이언트 명령과 원격 서버 역할을 하는 Slurm 데몬 간의 RPC 호출을 인증하는 데 사용됩니다. MungeKeySecretArn 제공되지 AWS ParallelCluster 않으면 클러스터에 대한 임의의 Munge 키가 생성됩니다.

#### Note

MungeKeySecretArn AWS ParallelCluster 버전 3.8.0부터 지원됩니다.

#### Warning

MungeKeySecretArn 가 기존 클러스터에 새로 추가된 경우, ParallelCluster 롤백 시 또는 나중에 제거할 때 이전 Munge Key가 복원되지 않습니다. MungeKeySecretArn 대신 무작위 Munge 키가 새로 생성됩니다.

AWS ParallelCluster 사용자에게 해당 특정 비밀 리소스에 [DescribeSecret](#) 대한 권한이 있는지 여부가 MungeKeySecretArn 확인됩니다. MungeKeySecretArn 다음과 같은 경우에 유효합니다.

- 지정된 비밀이 존재하고
- 암호는 일반 텍스트이며 유효한 base64로 인코딩된 문자열을 포함합니다.
- 디코딩된 바이너리 먼지 키의 크기는 256비트에서 8192비트 사이입니다.

pcluster 사용자 IAM 정책에 이 포함되지 DescribeSecret 않은 경우 MungeKeySecretArn 유효성이 검사되지 않고 경고 메시지가 표시됩니다. 자세한 정보는 [기본 AWS ParallelCluster pcluster 사용자 정책](#)을 참조하세요.

업데이트할 MungeKeySecretArn 때 컴퓨팅 플릿과 모든 로그인 노드를 중지해야 합니다.

ARN이 동일하게 유지되는 동안 비밀 ARN의 보안 값을 수정하면 클러스터가 새 munge 키로 자동 업데이트되지 않습니다. 비밀 ARN의 새 munge 키를 사용하려면 컴퓨팅 플릿과 로그인 노드를 중지한 다음 헤드 노드에서 다음 명령을 실행해야 합니다.

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

명령을 실행한 후 컴퓨팅 플릿과 로그인 노드를 모두 재개할 수 있습니다. 새로 프로비저닝된 컴퓨팅 및 로그인 노드는 새 munge 키를 사용하여 자동으로 시작됩니다.

Base64로 인코딩된 사용자 지정 먼지 키를 생성하려면 Munge 소프트웨어와 함께 배포된 [Mungekey](#) 유틸리티를 사용한 다음 OS에서 일반적으로 사용할 수 있는 base64 유틸리티를 사용하여 이를 인코딩하면 됩니다. 또는 bash를 사용할 수도 있습니다 (bs 파라미터를 32에서 1024 사이로 설정하십시오).

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

또는 Python은 다음과 같습니다.

```
import random
import os
import base64

# key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

업데이트 정책: COMPUTE FLEET 및 로그인 노드가 중지된 새 업데이트 정책 (3.7.0에서 잘못 추가되지 않음).

### QueueUpdateStrategy(선택 사항, String)

다음 업데이트 정책이 있는 [SlurmQueues](#) 섹션 파라미터의 대체 전략을 지정합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy를 설정해야 합니다.

이 QueueUpdateStrategy 값은 클러스터 업데이트 프로세스가 시작될 때만 사용됩니다.

유효한 값: COMPUTE\_FLEET\_STOP | DRAIN | TERMINATE

기본 값: COMPUTE\_FLEET\_STOP

## DRAIN

파라미터 값이 변경된 대기열의 노드는 DRAINING로 설정됩니다. 이 상태의 노드는 새 작업을 수락하지 않으며 실행 중인 작업은 계속 완료됩니다.

노드가 idle(DRAINED)가 된 후 노드가 정적이면 노드가 교체되고 동적이면 노드가 종료됩니다. 파라미터 값이 변경되지 않은 다른 대기열의 다른 노드는 영향을 받지 않습니다.

이 전략으로 모든 대기열 노드를 변경된 파라미터 값으로 교체해야 하는 시간은 실행 중인 워크로드에 따라 달라집니다.

## COMPUTE\_FLEET\_STOP

QueueUpdateStrategy 파라미터의 기본값입니다. 이 설정을 사용하는 경우, [SlurmQueues](#) 섹션 아래의 파라미터를 업데이트하려면 클러스터 업데이트를 수행하기 전에 [컴퓨팅 플릿을 중지해야](#) 합니다.

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

## TERMINATE

파라미터 값이 변경된 대기열에서는 실행 중인 작업이 종료되고 노드의 전원이 즉시 꺼집니다.

정적 노드는 교체되고 동적 노드는 종료됩니다.

파라미터 값이 변경되지 않은 다른 대기열의 다른 노드는 영향을 받지 않습니다.

업데이트 정책: 이 설정은 업데이트 중에 분석되지 않습니다.

### Note

QueueUpdateStrategy은 AWS ParallelCluster 버전 3.2.0부터 지원됩니다.

## EnableMemoryBasedScheduling(선택 사항, Boolean)

true이면 Slurm에 메모리 기반 스케줄링이 활성화됩니다. 자세한 내용은 [SlurmQueues/ComputeResources/SchedulableMemory](#) 항목을 참조하세요.

기본값은 false입니다.

**⚠ Warning**

메모리 기반 스케줄링을 활성화하면 Slurm 스케줄러가 작업 및 노드 할당을 처리하는 방식에 영향을 줍니다.

자세한 내용은 [Slurm 메모리 기반 스케줄링](#) 항목을 참조하세요.

**i Note**

`EnableMemoryBasedScheduling`은 AWS ParallelCluster 버전 3.2.0부터 지원됩니다.

**i Note**

AWS ParallelCluster [버전 3.7.0부터 인스턴스에 여러 인스턴스 유형을 구성하면 `EnableMemoryBasedScheduling` 활성화할 수 있습니다.](#)

AWS ParallelCluster 버전 3.2.0~3.6의 경우, [x, 인스턴스에서 여러 인스턴스 유형을 구성한 경우 활성화할 `EnableMemoryBasedScheduling` 수 없습니다.](#)

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

**CustomSlurmSettings(선택 사항, [Dict])**

전체 클러스터에 적용되는 사용자 지정 Slurm 설정을 정의합니다.

AWS ParallelCluster 가 생성하는 `slurm.conf` 파일 끝에 추가할 키-값 쌍의 Slurm 구성 사전 목록을 지정합니다.

목록의 각 사전은 Slurm 구성 파일에 추가된 별도의 줄로 표시됩니다. 단순 또는 복합 파라미터를 지정할 수 있습니다.

단순 파라미터는 다음 예제와 같이 단일 키 쌍으로 구성됩니다.

```
- Param1: 100
- Param2: "SubParam1,SubParam2=SubValue2"
```

Slurm 구성으로 렌더링된 예제:

```
Param1=100
```

```
Param2=SubParam1,SubParam2=SubValue2
```

복잡한 Slurm 구성 파라미터는 다음 예제와 같이 공백으로 구분된 여러 키-값 쌍으로 구성됩니다.

```
- NodeName: test-nodes[1-10]
  CPUs: 4
  RealMemory: 4196
  ... # other node settings
- NodeSet: test-nodeset
  Nodes: test-nodes[1-10]
  ... # other nodeset settings
- PartitionName: test-partition
  Nodes: test-nodeset
  ... # other partition settings
```

Slurm 구성으로 렌더링된 예제:

```
NodeName=test-nodes[1-10] CPUs=4 RealMemory=4196 ... # other node settings
NodeSet=test-nodeset Nodes=test-nodes[1-10] ... # other nodeset settings
PartitionName=test-partition Nodes=test-nodeset ... # other partition settings
```

#### Note

사용자 지정 Slurm 노드의 이름에 `-st-` 또는 `-dy-` 패턴을 포함해서는 안 됩니다. 이러한 패턴은 AWS ParallelCluster가 관리하는 노드에만 사용됩니다.

CustomSlurmSettings에서 사용자 지정 Slurm 구성 파라미터를 지정하는 경우 CustomSlurmSettingsIncludeFile에 대한 사용자 지정 Slurm 구성 파라미터를 지정해서는 안 됩니다.

CustomSlurmSettings 거부 목록에 없는 Slurm 구성 파라미터만 지정할 수 있습니다. 거부 목록에 있는 Slurm 구성 파라미터에 관한 자세한 내용은 [CustomSlurmSettings을 위한 거부 목록에 등록된 Slurm 구성 파라미터](#)를 참조하세요.

AWS ParallelCluster 파라미터가 거부 목록에 있는지 여부만 확인합니다. AWS ParallelCluster 사용자 지정 Slurm 구성 매개변수 구문이나 의미 체계를 검증하지 않습니다. 사용자 지정 Slurm 구성 파라미터의 유효성을 검사하는 것은 사용자의 책임입니다. 잘못된 사용자 지정 Slurm 구성 파라미터로 인해 Slurm 대몬(daemon) 장애가 발생하여 클러스터 생성 및 업데이트 실패로 이어질 수 있습니다.

를 사용하여 사용자 지정 Slurm 구성 매개변수를 지정하는 방법에 대한 자세한 내용은 [AWS ParallelCluster](#) 참조하십시오. [Slurm 구성 사용자 지정](#)

Slurm 구성 파라미터에 대한 자세한 내용은 Slurm의 [slurm.conf](#)를 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### Note

CustomSlurmSettings은 AWS ParallelCluster 버전 3.6.0부터 지원됩니다.

### CustomSlurmSettingsIncludeFile(선택 사항, String)

전체 클러스터에 적용되는 사용자 지정 Slurm 설정을 정의합니다.

AWS ParallelCluster 가 생성하는 `slurm.conf` 파일 끝에 추가할 사용자 지정 Slurm 구성 파라미터로 구성된 사용자 지정 Slurm 파일을 지정합니다.

파일의 경로를 포함해야 합니다. 경로는 `https://` 또는 `s3://`로 시작되어야 합니다.

CustomSlurmSettingsIncludeFile의 사용자 지정 Slurm 구성 파라미터를 지정하는 경우 CustomSlurmSettings에 대한 사용자 지정 Slurm 구성 파라미터를 지정해서는 안 됩니다.

#### Note

사용자 지정 Slurm 노드의 이름에 `-st-` 또는 `-dy-` 패턴을 포함해서는 안 됩니다. 이러한 패턴은 AWS ParallelCluster가 관리하는 노드에만 사용됩니다.

CustomSlurmSettingsIncludeFile 거부 목록에 없는 Slurm 구성 파라미터만 지정할 수 있습니다. 거부 목록에 있는 Slurm 구성 파라미터에 관한 자세한 내용은 [CustomSlurmSettings을 위한 거부 목록에 등록된 Slurm 구성 파라미터](#)를 참조하세요.

AWS ParallelCluster 매개변수가 거부 목록에 있는지 여부만 확인합니다. AWS ParallelCluster 사용자 지정 Slurm 구성 매개변수 구문이나 의미 체계를 검증하지 않습니다. 사용자 지정 Slurm 구성 파라미터의 유효성을 검사하는 것은 사용자의 책임입니다. 잘못된 사용자 지정 Slurm 구성 파라미터로 인해 Slurm 대몬(daemon) 장애가 발생하여 클러스터 생성 및 업데이트 실패로 이어질 수 있습니다.

를 사용하여 사용자 지정 Slurm 구성 매개변수를 지정하는 방법에 대한 자세한 내용은 [AWS ParallelCluster 참조하십시오. Slurm 구성 사용자 지정](#)

Slurm 구성 파라미터에 대한 자세한 내용은 Slurm의 [slurm.conf](#)를 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### Note

CustomSlurmSettings은 AWS ParallelCluster 버전 3.6.0부터 지원됩니다.

## Database

(선택 사항) 클러스터에서 Slurm 회계를 활성화하기 위한 설정을 정의합니다. 자세한 내용은 [Slurm 회계 담당자: AWS ParallelCluster](#) 항목을 참조하세요.

#### Database:

Uri: *string*

UserName: *string*

PasswordSecretArn: *string*

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

## Database 속성

### Uri(필수, String)

Slurm 회계의 백엔드로 사용되는 데이터베이스 서버의 주소입니다. 이 URI는 `host:port` 형식을 따라야 하며 `mysql://`과 같은 체계를 포함해서는 안 됩니다. 호스트는 헤드 노드에서 확인할 수 있는 IP 주소 또는 DNS 이름일 수 있습니다. 포트가 제공되지 않은 경우 AWS ParallelCluster 은 MySQL 기본 포트 3306을 사용합니다.

AWS ParallelCluster Slurm계정 데이터베이스를 클러스터에 부트스트랩하므로 데이터베이스에 액세스해야 합니다.

다음과 같은 상황이 발생하기 전에 데이터베이스에 연결할 수 있어야 합니다.

- 클러스터가 생성됩니다.
- Slurm 회계가 클러스터 업데이트로 활성화됩니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### UserName(필수, String)

Slurm가 데이터베이스에 연결하고, 계정 로그를 작성하고, 쿼리를 수행하는 데 사용되는 ID 사용자에게 데이터베이스에 대한 읽기 및 쓰기 권한이 모두 있어야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### PasswordSecretArn(필수, String)

UserName 일반 텍스트 비밀번호가 포함된 AWS Secrets Manager 비밀번호의 Amazon 리소스 이름 (ARN). 이 비밀번호는 데이터베이스 서버에서 인증하기 위해 UserName 및 Slurm 회계와 함께 사용됩니다.

#### Note

AWS Secrets Manager 콘솔을 사용하여 암호를 생성할 때는 “기타 유형의 암호”를 선택하고 일반 텍스트를 선택한 다음 암호 텍스트만 암호에 포함해야 합니다.

암호를 만드는 AWS Secrets Manager 데 사용하는 방법에 대한 자세한 내용은 암호 [만들기](#)를 참조하십시오. AWS Secrets Manager

사용자에게 [DescribeSecret](#) 권한이 있는 경우 PasswordSecretArn 유효성이 확인됩니다. PasswordSecretArn 지정된 비밀이 존재할 경우 유효합니다. 사용자 IAM 정책이 DescribeSecret을 포함하지 않는 경우 PasswordSecretArn가 검증되지 않고 경고 메시지가 표시됩니다. 자세한 내용은 [기본 AWS ParallelCluster pcluster 사용자 정책](#) 항목을 참조하세요.

PasswordSecretArn를 업데이트할 때는 컴퓨팅 플릿을 중지해야 합니다. 암호 값이 변경되고 암호 ARN이 변경되지 않는 경우 클러스터는 새 데이터베이스 암호로 자동 업데이트되지 않습니다. 새 암호 값에 맞게 클러스터를 업데이트하려면 컴퓨팅 플릿이 중지된 후 헤드 노드 내에서 다음 명령을 실행해야 합니다.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

#### Warning

계정 데이터가 손실되지 않도록 컴퓨팅 플릿이 중지된 경우에만 데이터베이스 비밀번호를 변경하는 것이 좋습니다.



업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### DatabaseName(선택 사항, String)

Slurm어카운팅에 사용할 데이터베이스 서버의 데이터베이스 이름 (Uri 매개 변수로 정의됨).

데이터베이스 이름에는 소문자, 숫자, 밑줄이 포함될 수 있습니다. 이름은 64자를 초과할 수 없습니다.

이 매개변수는 [slurmdbd.conf의 StorageLoc](#) 매개변수에 매핑됩니다.

DatabaseName 지정되지 않은 경우 ParallelCluster 클러스터 이름을 사용하여 값을 정의합니다.  
StorageLoc

DatabaseName 업데이트는 허용되며, 다음 사항을 고려하면 됩니다.

- 이름이 있는 데이터베이스가 데이터베이스 서버에 아직 DatabaseName 없는 경우 slurmdbd는 데이터베이스를 생성합니다. 필요에 따라 새 데이터베이스를 재구성하는 것은 사용자의 책임입니다 (예: 클러스터, 계정, 사용자, 연결, QoS 등).
- 데이터베이스 서버에 이름이 있는 데이터베이스가 DatabaseName 이미 있는 경우 slurmdbd는 해당 데이터베이스를 계정 기능에 사용합니다. Slurm

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

#### Note

Database이 릴리스 3.3.0부터 추가됩니다.

### ExternalSlurmdbd

(선택 사항) 외부 slurmdbd 서버를 통한 Slurm 어카운팅을 활성화하기 위한 설정을 정의합니다. [자세한 내용은 어카운팅을 참조하십시오Slurm. AWS ParallelCluster](#)

ExternalSlurmdbd:

Host: *string*

Port: *integer*

## ExternalSlurmdbd 속성

### Host(필수,String)

어카운팅을 위한 외부 slurmdbd 서버의 주소. Slurm 호스트는 헤드 노드에서 확인할 수 있는 IP 주소 또는 DNS 이름일 수 있습니다.

[업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.](#)

### Port(선택 사항,) Integer

slurmdbd 서비스가 수신하는 포트입니다. 기본값은 6819입니다.

[업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.](#)

## Dns

(선택 사항) 전체 클러스터에 적용되는 Slurm 설정을 정의합니다.

### Dns:

[DisableManagedDns](#): *boolean*

[HostedZoneId](#): *string*

[UseEc2Hostnames](#): *boolean*

## Dns 속성

### DisableManagedDns(선택 사항, Boolean)

true이면 클러스터의 DNS 항목이 생성되지 않고 Slurm 노드 이름을 확인할 수 없습니다.

기본적으로 시작 시 노드가 등록되는 Route 53 호스팅 영역을 AWS ParallelCluster 생성합니다. 기본값은 false입니다. DisableManagedDns로 true 설정하면 에서 호스팅 영역을 생성하지 않습니다 AWS ParallelCluster.

이 설정을 사용하여 인터넷에 액세스할 수 없는 서브넷에 클러스터를 배포하는 방법을 알아보려면 [AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서](#)을 참조하세요.

### Warning

클러스터가 제대로 작동하려면 이름 확인 시스템이 필요합니다. DisableManagedDns이 true로 설정되어 있으면 이름 확인 시스템을 제공해야 합니다. Amazon EC2 기

본 DNS를 사용하려면 로 설정합니다 `UseEc2Hostnames: true` 또는 자체 DNS 해석기를 구성하고 인스턴스 시작 시 노드 이름이 등록되도록 하세요. 예를 들어 [CustomActions/OnNodeStart](#)를 구성하여 이 작업을 수행할 수 있습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### HostedZoneId(선택 사항, String)

클러스터의 DNS 이름 확인에 사용할 사용자 지정 Route 53 호스팅 영역 ID를 정의합니다. 제공된 경우, 지정된 호스팅 영역에 클러스터 노드를 AWS ParallelCluster 등록하고 관리형 호스팅 영역을 생성하지 않습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### UseEc2Hostnames(선택 사항, Boolean)

`true`이면 클러스터 컴퓨팅 노드가 기본 EC2 호스트 이름으로 구성됩니다. 또한 Slurm `NodeHostName`가 이 정보로 업데이트됩니다. 기본값은 `false`입니다.

이 설정을 사용하여 인터넷에 액세스할 수 없는 서브넷에 클러스터를 배포하는 방법을 알아보려면 [AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서](#)를 참조하세요.

#### Note

이 노트는 AWS ParallelCluster 버전 3.3.0부터는 관련이 없습니다.

3.3.0 이전 AWS ParallelCluster 지원 버전의 경우:

`UseEc2Hostnames`가 로 `true` 설정된 경우 Slurm 구성 파일은 및 스크립트와 함께 설정됩니다. AWS ParallelCluster `prolog` `epilog`

- `prolog`를 실행하여 각 작업이 할당될 때 컴퓨팅 노드의 `/etc/hosts`에 노드 정보를 추가합니다.
  - `epilog`를 실행하여 `prolog`가 작성한 내용을 정리합니다.
- 사용자 정의 `prolog` 또는 `epilog` 스크립트를 추가하려면 각각 `/opt/slurm/etc/pcluster/prolog.d/` 또는 `/opt/slurm/etc/pcluster/epilog.d/` 폴더에 추가하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## SharedStorage 섹션

(선택 사항) 클러스터의 공유 스토리지 설정

AWS ParallelCluster 는 [Amazon EBS, ONTAP용 FSx, OpenZFS 공유 스토리지 볼륨의 경우 FSx, Lustre 공유 스토리지 파일 시스템용 Amazon EFS 및 FSx 또는 파일 캐시를 사용할 수 있습니다.](#)

SharedStorage 섹션에서는 외부 또는 관리형 스토리지를 정의할 수 있습니다.

- 외부 스토리지는 관리하는 기존 볼륨 또는 파일 시스템을 말합니다. AWS ParallelCluster 생성하거나 삭제하지 않습니다.
- AWS ParallelCluster 관리형 스토리지는 AWS ParallelCluster 생성하여 삭제할 수 있는 볼륨 또는 파일 시스템을 말합니다.

[공유 저장소 할당량](#) 및 공유 저장소 구성에 대한 자세한 내용은 AWS ParallelCluster 사용에서 [공유 스토리지](#)를 참조하세요.

### Note

를 스케줄러로 사용하는 경우 FSx for AWS Batch Lustre는 클러스터 헤드 노드에서만 사용할 수 있습니다.

### SharedStorage:

- MountDir: *string*
- Name: *string*
- StorageType: Ebs
- EbsSettings:
  - VolumeType: *string*
  - Iops: *integer*
  - Size: *integer*
  - Encrypted: *boolean*
  - KmsKeyId: *string*
  - SnapshotId: *string*
  - Throughput: *integer*
  - VolumeId: *string*
  - DeletionPolicy: *string*
  - Raid:
    - Type: *string*
    - NumberOfVolumes: *integer*
- MountDir: *string*

```

Name: string
StorageType: Efs
EfsSettings:
  Encrypted: boolean
  KmsKeyId: string
  EncryptionInTransit: boolean
  IamAuthorization: boolean
  PerformanceMode: string
  ThroughputMode: string
  ProvisionedThroughput: integer
  FileSystemId: string
  DeletionPolicy: string
- MountDir: string
  Name: string
  StorageType: FsxLustre
  FsxLustreSettings:
    StorageCapacity: integer
    DeploymentType: string
    ImportedFileChunkSize: integer
    DataCompressionType: string
    ExportPath: string
    ImportPath: string
    WeeklyMaintenanceStartTime: string
    AutomaticBackupRetentionDays: integer
    CopyTagsToBackups: boolean
    DailyAutomaticBackupStartTime: string
    PerUnitStorageThroughput: integer
    BackupId: string
    KmsKeyId: string
    FileSystemId: string
    AutoImportPolicy: string
    DriveCacheType: string
    StorageType: string
    DeletionPolicy: string
    DataRepositoryAssociations:
      - Name: string
        BatchImportMetaDataOnCreate: boolean
        DataRepositoryPath: string
        FileSystemPath: string
        ImportedFileChunkSize: integer
        AutoExportPolicy: string
        AutoImportPolicy: string
- MountDir: string
  Name: string

```

```

StorageType: FsxOntap
FsxOntapSettings:
  VolumeId: string
- MountDir: string
  Name: string
StorageType: FsxOpenZfs
FsxOpenZfsSettings:
  VolumeId: string
- MountDir: string
  Name: string
StorageType: FileCache
FileCacheSettings:
  FileCacheId: string

```

## SharedStorage 업데이트 정책

- 관리형/외부 EBS, 관리형 EFS 및 관리형 FSx Lustre의 업데이트 정책은 다음과 같습니다. [업데이트 정책: 이 목록 값 설정의 경우 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy가 새 값을 추가 하도록 설정해야 합니다. 기존 값을 제거할 때는 컴퓨팅 플릿을 중지해야 합니다.](#)
- 외부 EFS, FSx Lustre, FSx ONTAP, FSx 및 파일 캐시의 경우 업데이트 정책은 다음과 같습니다. OpenZfs [업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.](#)

## SharedStorage 속성

MountDir(필수, String)

공유 스토리지가 탑재된 경로입니다.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

Name(필수, String)

공유 스토리지의 이름 설정을 업데이트할 때 이 이름을 사용합니다.

### Warning

AWS ParallelCluster 관리형 공유 스토리지를 지정하고 값을 변경하면 기존 관리형 공유 스토리지와 데이터가 Name 삭제되고 새 관리형 공유 스토리지가 생성됩니다. 클러스터 업데이트로 Name의 값을 변경하는 것은 기존의 관리형 공유 저장소를 새 저장소로 교체하는 것과 같습니다. 기존 공유 스토리지의 데이터를 보존해야 하는 경우 Name를 변경하기 전에 데이터를 백업해야 합니다.

업데이트 정책: 이 목록 값 설정의 경우 컴퓨팅 플릿을 중지하거나 QueueUpdateStrategy가 새 값을 추가하도록 설정해야 합니다. 기존 값을 제거할 때는 컴퓨팅 플릿을 중지해야 합니다.

## StorageType(필수, String)

공유 스토리지의 유형 지원되는 값은 Ebs, Efs, FsxLustre, FsxOntap, FsxOpenZfs입니다.

자세한 내용은 [FsxLustreSettings](#), [FsxOntapSettings](#), [FsxOpenZfsSettings](#) 항목을 참조하세요.

### Note

AWS Batch 스케줄러로 사용하는 경우 FSx for Lustre는 클러스터 헤드 노드에서만 사용할 수 있습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## EbsSettings

(선택 사항) Amazon EBS 볼륨 설정

```
EbsSettings:
  VolumeType: string
  Iops: integer
  Size: integer
  Encrypted: boolean
  KmsKeyId: string
  SnapshotId: string
  VolumeId: string
  Throughput: integer
  DeletionPolicy: string
  Raid:
    Type: string
    NumberOfVolumes: integer
```

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## EbsSettings 속성

로 Delete 설정하면 [DeletionPolicy](#) 클러스터가 삭제되거나 클러스터 업데이트로 볼륨이 제거되면 해당 데이터와 함께 관리 볼륨이 삭제됩니다.

자세한 내용은 AWS ParallelCluster 사용에서 [공유 스토리지](#)를 참조하세요.

VolumeType(선택 사항, String)

[Amazon EBS 볼륨 유형](#)을 지정합니다. 지원되는 값은 gp2, gp3, io1, io2, sc1, st1, standard입니다. 기본값은 gp3입니다.

자세한 내용을 알아보려면 Amazon EC2 사용 설명서의 [Amazon EBS 볼륨 유형](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

Iops(선택 사항, Integer)

io1, io2, gp3 유형 볼륨의 IOPS 수를 정의합니다.

기본 값, 지원되는 값, volume\_iops:volume\_size 비율은 VolumeType 및 Size에 따라 달라집니다.

VolumeType = io1

기본 Iops = 100

지원되는 값 Iops = 100-64000 †

최대 volume\_iops:volume\_size 비율은 GiB당 50 IOPS입니다. 5000 IOPS에는 최소 100GiB의 volume\_size 필요합니다.

VolumeType = io2

기본 Iops = 100

지원되는 값 Iops = 100-64000 (io2 블록 익스프레스 볼륨의 경우 256000) †

최대 Iops:Size 비율은 GiB당 500 IOPS입니다. 5000 IOPS에는 최소 10GiB의 Size가 필요합니다.

VolumeType = gp3

기본 Iops = 3000

지원되는 값 Iops = 3000-16000 †

최대 Iops:Size 비율은 GiB당 500 IOPS입니다. 5000 IOPS에는 최소 10GiB의 Size 필요합니다.



† 최대 IOPS는 32,000 IOPS 이상으로 프로비저닝된 [Nitro 시스템 기반 인스턴스](#)에서만 보장됩니다. 다른 인스턴스는 최대 32,000 IOPS를 보장합니다. [볼륨을 수정하지 않는 한](#) 이전 io1 볼륨은 전체 성능에 도달할 수 없습니다. io2 블록 익스프레스 볼륨은 R5b 인스턴스 유형에서 최대 256000의 volume\_iops 값을 지원합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [io2 블록 익스프레스 볼륨](#)을 참조하십시오.

[업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.](#)

Size(선택 사항, Integer)

볼륨 크기를 기비바이트(GiB) 단위로 지정합니다. 기본값은 35입니다.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

Encrypted(선택 사항, Boolean)

볼륨이 암호화되는지 여부를 지정합니다. 기본값은 true입니다.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

KmsKeyId(선택 사항, String)

암호화에 사용할 사용자 지정 AWS KMS 키를 지정합니다. 이 설정을 사용하려면 Encrypted 설정을 true로 설정해야 합니다.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

SnapshotId(선택 사항, String)

스냅샷을 볼륨의 소스로 사용하는 경우 Amazon EBS 스냅샷 ID를 지정합니다.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

VolumeId(선택 사항, String)

Amazon EBS 볼륨 ID를 지정합니다. 이가 EbsSettings 인스턴스에 대해 지정되어 있는 경우 MountDir 파라미터만 지정할 수도 있습니다.

볼륨은 HeadNode과 같은 가용 영역에 생성되어야 합니다.

#### Note

다중 가용 영역은 AWS ParallelCluster 버전 3.4.0에 추가되었습니다.

[업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.](#)

## Throughput(선택 사항, Integer)

볼륨에 대해 프로비저닝할 처리량(MiB/S 단위)입니다(최대 1000MiB/s).

이 설정은 VolumeType가 gp3일 때만 유효합니다. 지원되는 범위는 125에서 1000까지이며 기본 값은 125입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## DeletionPolicy(선택 사항, String)

클러스터를 삭제하거나 볼륨을 제거할 때 볼륨을 유지, 삭제 또는 스냅샷할지 여부를 지정합니다. 지원되는 값은 Delete, Retain, Snapshot입니다. 기본값은 Delete입니다.

로 [DeletionPolicy](#) Delete 설정하면 클러스터가 삭제되거나 클러스터 업데이트로 볼륨이 제거되면 관리형 볼륨과 해당 데이터가 삭제됩니다.

자세한 정보는 [공유 스토리지](#)을 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Note

DeletionPolicy AWS ParallelCluster 버전 3.2.0부터 지원됩니다.

## Raid

(선택 사항) RAID 볼륨의 구성을 정의합니다.

### Raid:

Type: *string*

NumberOfVolumes: *integer*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Raid 속성

### Type(필수, String)

RAID 어레이 유형을 정의합니다. 지원되는 값은 "0"(스트라이프) 및 "1"(미러링)입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## NumberOfVolumes(선택 사항, Integer)

RAID 어레이를 생성하는 데 사용할 Amazon EBS 볼륨의 수를 정의합니다. 지원되는 값 범위는 2~5입니다. 기본값(Raid설정이 정의된 경우)은 2입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## EfsSettings

(선택 사항) Amazon EFS 파일 시스템 설정

### EfsSettings:

Encrypted: *boolean*

KmsKeyId: *string*

EncryptionInTransit: *boolean*

IamAuthorization: *boolean*

PerformanceMode: *string*

ThroughputMode: *string*

ProvisionedThroughput: *integer*

FileSystemId: *string*

DeletionPolicy: *string*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## EfsSettings 속성

로 DeletionPolicyDelete설정하면 클러스터가 삭제되거나 클러스터 업데이트로 파일 시스템이 제거 되면 해당 데이터와 함께 관리되는 파일 시스템이 삭제됩니다.

자세한 내용은 AWS ParallelCluster사용에서 공유 스토리지를 참조하세요.

## Encrypted(선택 사항, Boolean)

Amazon EFS 파일 시스템이 암호화되는지 여부를 지정합니다. 기본값은 false입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## KmsKeyId(선택 사항, String)

암호화에 사용할 사용자 지정 AWS KMS 키를 지정합니다. 이 설정을 사용하려면 Encrypted 설정을 true로 설정해야 합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## EncryptionInTransit(선택 사항, Boolean)

true로 설정된 경우 Amazon EFS 파일 시스템이 전송 계층 보안(TLS)을 사용하여 탑재됩니다. 기본값은 false로 설정되어 있습니다.

### Note

스케줄러로 사용되는 경우 AWS Batch EncryptionInTransit 지원되지 않습니다.

### Note

EncryptionInTransit이 AWS ParallelCluster 3.4.0부터 추가됩니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## IamAuthorization(선택 사항, Boolean)

true로 설정된 경우 시스템의 IAM ID를 사용하여 Amazon EFS를 인증합니다. 기본값은 false로 설정되어 있습니다.

### Note

IamAuthorization이 true으로 설정된 경우 EncryptionInTransit도 true으로 설정되어야 합니다.

### Note

스케줄러로 AWS Batch 사용되며 지원되지 IamAuthorization 않습니다.

### Note

IamAuthorization AWS ParallelCluster 버전 3.4.0부터 추가되었습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## PerformanceMode(선택 사항, String)

Amazon EFS 파일 시스템의 성능 모드를 지정합니다. 지원되는 값은 `generalPurpose` 및 `maxIO`입니다. 기본값은 `generalPurpose`입니다. 자세한 내용을 알아보려면 Amazon Elastic File System User Guide(Amazon Elastic File System 사용 설명서)의 [Performance modes](#)(성능 모드)를 참조하세요.

대부분의 파일 시스템에 `generalPurpose` 성능 모드를 사용하는 것이 좋습니다.

`maxIO` 성능 모드를 사용하는 파일 시스템은 더 높은 수준의 집계 처리량 및 초당 작업으로 확장할 수 있습니다. 그러나 대부분의 파일 작업에서 대기 시간이 조금 더 길다는 단점이 있습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## ThroughputMode(선택 사항, String)

Amazon EFS 파일 시스템의 처리량 모드를 지정합니다. 지원되는 값은 `bursting` 및 `provisioned`입니다. 기본값은 `bursting`입니다. `provisioned`가 사용되면 `ProvisionedThroughput`을 지정해야 합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## ProvisionedThroughput(ThroughputMode가 `provisioned`, Integer면 필수)

MiB/s로 측정되는 Amazon EFS 파일 시스템의 프로비저닝된 처리량(MiB/s 단위)을 정의합니다. 이는 Amazon EFS API 참조의 [ProvisionedThroughputInMibps](#) 파라미터에 해당합니다.

이 파라미터를 사용할 경우 `ThroughputMode`를 `provisioned`로 설정해야 합니다.

지원되는 범위는 1~1024입니다. 제한 증가를 요청하려면 AWS Support에 문의하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## FileSystemId(선택 사항, String)

기존 파일 시스템의 Amazon EFS 파일 시스템 ID를 정의합니다.

클러스터가 여러 가용 영역에 걸쳐 있도록 구성된 경우 클러스터에서 사용하는 각 가용 영역에서 파일 시스템 탑재 대상을 정의해야 합니다.


이것이 지정되어 있으면 `MountDir`만 지정될 수 있습니다. 다른 `EfsSettings`은 지정할 수 없습니다.

이 옵션을 설정하는 경우 정의한 파일 시스템에 대해 다음이 true여야 합니다.

- 각 클러스터의 가용 영역에 기존 탑재 대상이 있으며 HeadNode 및 ComputeNodes에서 허용된 인바운드 및 아웃바운드 NFS 트래픽이 있는 파일 시스템 [스케줄링/SlurmQueues/SubnetIds네트워크](#)에 여러 가용 영역이 구성되어 있습니다.


클러스터와 파일 시스템 간에 트래픽이 허용되도록 하려면 다음 중 한 가지를 수행하세요.

- 클러스터 서브�트의 CIDR 또는 접두사 목록으로 들어오고 나가는 트래픽을 허용하도록 탑재 대상의 보안 그룹을 구성합니다.


 Note

AWS ParallelCluster 포트가 열려 있고 CIDR 또는 접두사 목록이 구성되어 있는지 확인합니다. AWS ParallelCluster CIDR 블록 또는 접두사 목록의 내용을 확인하지 않습니다.


- [SlurmQueues/Networking/SecurityGroups](#) 및 [HeadNode/Networking/SecurityGroups](#)를 사용하여 클러스터 노드의 사용자 지정 보안 그룹을 설정합니다. 클러스터와 파일 시스템 간의 트래픽을 허용하도록 사용자 지정 보안 그룹을 구성해야 합니다.

 Note

모든 클러스터 노드가 사용자 지정 보안 그룹을 사용하는 경우 포트가 열려 AWS ParallelCluster 있는지 확인합니다. AWS ParallelCluster 소스와 대상이 제대로 구성되었는지 검증하지 않습니다.

 Warning

OneZone EFS는 모든 컴퓨팅 노드와 헤드 노드가 동일한 가용 영역에 있는 경우에만 지원됩니다. EFS에는 탑재 대상이 하나만 있을 OneZone 수 있습니다.

 Note

AWS ParallelCluster 버전 3.4.0에 다중 가용 영역이 추가되었습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## DeletionPolicy(선택 사항, String)

클러스터에서 파일 시스템을 제거하거나 클러스터를 삭제할 때 파일 시스템을 유지할지 또는 삭제할지 여부를 지정합니다. 지원되는 값은 Delete 및 Retain입니다. 기본값은 Delete입니다.

DeletionPolicy가 로 설정된 경우Delete, 클러스터가 삭제되거나 클러스터 업데이트로 파일 시스템이 제거되면 관리되는 파일 시스템과 해당 데이터가 함께 삭제됩니다.

자세한 정보는 [공유 스토리지](#)을 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Note

DeletionPolicy AWS ParallelCluster 버전 3.3.0부터 지원됩니다.

## FsxLustreSettings

### Note

StorageType에 FsxLustre가 지정되어 있으면 FsxLustreSettings를 정의해야 합니다.

(선택 사항) FSx for Lustre 파일 시스템 설정

### FsxLustreSettings:

```

StorageCapacity: integer
DeploymentType: string
ImportedFileChunkSize: integer
DataCompressionType: string
ExportPath: string
ImportPath: string
WeeklyMaintenanceStartTime: string
AutomaticBackupRetentionDays: integer
CopyTagsToBackups: boolean
DailyAutomaticBackupStartTime: string
PerUnitStorageThroughput: integer
BackupId: string # BackupId cannot coexist with some of the fields
KmsKeyId: string

```

```

FileSystemId: string # FileSystemId cannot coexist with other fields
AutoImportPolicy: string
DriveCacheType: string
StorageType: string
DeletionPolicy: string

```

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### Note

를 스케줄러로 사용하는 경우 FSx for AWS Batch Lustre는 클러스터 헤드 노드에서만 사용할 수 있습니다.

## FsxLustreSettings 속성

[DeletionPolicy](#)가 로 설정된 경우Delete, 클러스터가 삭제되거나 클러스터 업데이트로 파일 시스템이 제거되면 해당 데이터와 함께 관리되는 파일 시스템이 삭제됩니다.

자세한 정보는 [공유 스토리지](#)을 참조하세요.

### StorageCapacity(필수, Integer)

FSx for Lustre 파일 시스템의 스토리지 용량을 GiB 단위로 설정합니다. 새로운 파일 시스템을 생성할 경우 StorageCapacity가 필요합니다. BackupId 또는 FileSystemId가 지정된 경우에는 StorageCapacity을 포함하지 마세요.

- SCRATCH\_2, PERSISTENT\_1, PERSISTENT\_2 배포 유형의 경우 유효한 값은 1200GiB, 2400GiB 및 2400GiB의 증분입니다.
- SCRATCH\_1 배포 유형의 경우 유효한 값은 1200GiB, 2400GiB 및 3600GiB의 증분입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### DeploymentType(선택 사항, String)

FSx for Lustre 파일 시스템의 배포 유형을 지정합니다. 지원되는 값은 SCRATCH\_1, SCRATCH\_2, PERSISTENT\_1 및 PERSISTENT\_2입니다. 기본값은 SCRATCH\_2입니다.

임시 스토리지 및 단기 데이터 처리가 필요한 경우 SCRATCH\_1 및 SCRATCH\_2 배포 유형을 선택합니다. SCRATCH\_2 배포 유형은 전송 중 데이터 암호화와 SCRATCH\_1보다 높은 버스트 처리량 용량을 제공합니다.



장기 스토리지와 지연 시간에 민감하지 않은 처리량 중심 워크로드의 경우 PERSISTENT\_1 배포 유형을 선택하세요. PERSISTENT\_1는 전송 중 데이터의 암호화를 지원합니다. FSx for Lustre를 사용할 수 있는 모든 곳에서 사용할 수 있습니다.

장기 스토리지와 최고 수준의 IOPS 및 처리량이 필요한 지연 시간에 민감한 워크로드의 경우 PERSISTENT\_2 배포 유형을 선택합니다. PERSISTENT\_2는 SSD 스토리지를 지원하며, 더 높은 PerUnitStorageThroughput(최대 1000MB/s/TiB)를 제공합니다. PERSISTENT\_2는 제한된 수의 AWS 리전에서 사용할 수 있습니다. 배포 유형 및 사용 가능한 AWS 리전 위치 [PERSISTENT\\_2 목록](#)에 대한 자세한 내용은 Amazon FSx for [Lustre 사용 설명서에서 FSx for Lustre의 파일 시스템 배포 옵션](#)을 참조하십시오.

전송 중 데이터 암호화는 [이 특성](#)을 지원하는 Amazon EC2 인스턴스에서 SCRATCH\_2, PERSISTENT\_1 또는 PERSISTENT\_2 파일 시스템에 액세스할 때 자동으로 활성화됩니다.

지원되는 AWS 리전의 지원되는 인스턴스 유형에서 액세스할 경우 SCRATCH\_2, PERSISTENT\_1 및 PERSISTENT\_2 배포 유형에 대해 전송 중 데이터 암호화가 지원됩니다. 자세한 내용을 알아보려면 Amazon FSx for Lustre 사용 설명서의 [전송 중 데이터 암호화](#)를 참조하세요.

#### Note

PERSISTENT\_2 배포 유형에 대한 지원이 AWS ParallelCluster 버전 3.2.0에 추가되었습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

ImportedFileChunkSize(선택 사항, Integer)

데이터 리포지토리에서 가져온 파일의 경우 이 값은 단일 물리적 디스크에 저장된 파일당 스트라이프 수 및 최대 데이터 양(MiB 단위)을 결정합니다. 단일 파일을 스트라이프할 수 있는 최대 디스크 수는 파일 시스템을 구성하는 총 디스크 수에 따라 제한됩니다.

체크 크기 기본값은 1,024MiB(1GiB)이며 최대 512,000MiB(500GiB)까지 가능합니다. Amazon S3 객체의 크기는 최대 5TB입니다.

#### Note

이 파라미터는 PERSISTENT\_2 배포 유형을 사용하는 파일 시스템에서 지원되지 않습니다. 데이터 리포지토리 연결을 구성하는 방법에 대한 지침은 Amazon FSx for Lustre 사용 설명서의 [S3 버킷에 파일 시스템 연결](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### DataCompressionType(선택 사항, String)

Amazon FSx for Lustre 파일 시스템의 데이터 압축 구성을 설정합니다. 지원되는 값은 LZ4입니다. LZ4는 LZ4 알고리즘으로 데이터 압축이 켜져 있음을 나타냅니다. DataCompressionType이 지정되지 않은 경우 파일 시스템이 생성될 때 데이터 압축이 꺼집니다.

자세한 내용은 [Lustre 데이터 압축](#)을 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### ExportPath(선택 사항, String)

FSx for Lustre 파일 시스템의 루트가 내보내지는 Amazon S3의 경로입니다. 이 설정은 ImportPath 파라미터가 지정된 경우에만 지원됩니다. 이 경로는 ImportPath에 지정된 것과 동일한 Amazon S3 버킷을 사용해야 합니다. FSx for Lustre 파일 시스템에서 새로운 데이터 및 변경된 데이터를 내보낼 선택적 접두사를 제공할 수 있습니다. ExportPath 값이 제공되지 않으면 FSx for Lustre는 기본 내보내기 경로를 s3://import-bucket/FSxLustre[creation-timestamp]로 설정합니다. 타임스탬프는 UTC 형식(예: s3://import-bucket/FSxLustre20181105T222312Z)입니다.

Amazon S3 내보내기 버킷은 ImportPath에서 지정한 가져오기 버킷과 같아야 합니다. 버킷 이름만 지정하면(예: s3://import-bucket) 파일 시스템 객체와 Amazon S3 버킷 객체가 1:1로 매핑됩니다. 이 매핑은 내보내기 시 Amazon S3의 입력 데이터를 덮어쓴다는 뜻입니다. 내보내기 경로에 사용자 지정 접두사(예: s3://import-bucket/[custom-optional-prefix])를 제공하면 FSx for Lustre는 파일 시스템의 콘텐츠를 Amazon S3 버킷의 해당 내보내기 접두사로 내보냅니다.

#### Note

이 파라미터는 PERSISTENT\_2 배포 유형을 사용하는 파일 시스템에서 지원되지 않습니다. Amazon FSx for Lustre 사용 설명서의 [S3 버킷에 파일 시스템 연결](#)에 설명된 대로 데이터 리포지토리 연결을 구성하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### ImportPath(선택 사항, String)

FSx for Luster 파일 시스템의 데이터 리포지토리로 사용 중인 Amazon S3 버킷(선택적 접두사 포함)의 경로입니다. FSx for Luster 파일 시스템의 루트는 선택한 Amazon S3 버킷의 루트에 매핑됨

니다. 예를 들면, `s3://import-bucket/optional-prefix`입니다. Amazon S3 버킷 이름 뒤에 접두사를 지정하면 해당 접두사가 있는 객체 키만 파일 시스템에 로드됩니다.

#### Note

이 파라미터는 PERSISTENT\_2 배포 유형을 사용하는 파일 시스템에서 지원되지 않습니다. Amazon FSx for Lustre 사용 설명서의 [S3 버킷에 파일 시스템 연결](#)에 설명된 대로 데이터 리포지토리 연결을 구성하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

WeeklyMaintenanceStartTime(선택 사항, String)

주별 유지 관리를 수행하기 위한 기본 시작 시간 UTC+0 시간대의 "d:HH:MM" 형식입니다. 이 형식에서 d는 월요일로 시작해 일요일로 끝나는 1부터 7까지의 요일 숫자입니다. 이 필드에는 따옴표가 필요합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

AutomaticBackupRetentionDays(선택 사항, Integer)

자동 백업을 보존할 일수입니다. 이 값을 0으로 설정하면 자동 백업이 비활성화됩니다. 지원되는 범위는 0~90입니다. 기본값은 0입니다. 이 설정은 PERSISTENT\_1 및 PERSISTENT\_2 배포 유형과 함께 사용할 때만 유효합니다. 자세한 내용은 Amazon FSx for Lustre 사용 설명서의 [백업 작업](#)을 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

CopyTagsToBackups(선택 사항, Boolean)

true면 FSx for Lustre 파일 시스템의 태그를 백업에 복사합니다. 기본값은 false입니다. true로 설정하면 파일 시스템의 모든 태그는 사용자가 태그를 지정하지 않은 모든 자동 및 사용자 시작 백업에 복사됩니다. 이 값이 true이고 하나 이상의 태그를 지정하면 지정된 태그만 백업에 복사됩니다. 사용자가 시작한 백업을 만들 때 하나 이상의 태그를 지정하면 이 값에 관계없이 파일 시스템에서 태그가 복사되지 않습니다. 이 설정은 PERSISTENT\_1 및 PERSISTENT\_2 배포 유형과 함께 사용할 때만 유효합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## DailyAutomaticBackupStartTime(선택 사항, String)

HH:MM 형식의 매일 반복되는 시간입니다. HH는 제로 패딩된 하루 중 시간(0~23)이고, MM은 제로 패딩된 시간의 분(00~59)입니다. 예를 들어, 05:00은 매일 오전 5시를 지정합니다. 이 설정은 PERSISTENT\_1 및 PERSISTENT\_2 배포 유형과 함께 사용할 때만 유효합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## PerUnitStorageThroughput(PERSISTENT\_1 및 PERSISTENT\_2 배포 유형 사용 시 필수, Integer)

스토리지의 1테비바이트당 읽기 및 쓰기 처리량(MB/s/TiB)을 설명합니다. 파일 시스템 처리량 용량은 파일 시스템 스토리지 용량(TiB)에 PerUnitStorageThroughput(MB/s/TiB)를 곱하여 계산됩니다. 2.4TiB 파일 시스템의 경우 50MB/s/TiB의 PerUnitStorageThroughput를 프로비저닝하여 120MB/s의 파일 시스템 처리량을 얻을 수 있습니다. 프로비저닝한 처리량에 대해 비용을 지불합니다. 이는 속성에 [PerUnitStorageThroughput](#) 해당합니다.

유효한 값:

PERSISTENT\_1 SSD 스토리지: 50, 100, 200MB/TiB

PERSISTENT\_1 HDD 스토리지: 12, 40 MB/TiB

PERSISTENT\_2 SSD 스토리지: 125, 250, 500, 1000MB/s/TiB

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## BackupId(선택 사항, String)

기존 백업에서 FSx for Lustre 파일 시스템을 복원하는 데 사용할 백업의 ID를 지정합니다. BackupId 설정이 지정되어 있으면 AutoImportPolicy, DeploymentType, ExportPath, KmsKeyId, ImportPath, ImportedFileChunkSize, StorageCapacity 및 PerUnitStorageThroughput 설정을 지정하지 않아야 합니다. 이러한 설정은 백업에서 읽습니다. 또한, AutoImportPolicy, ExportPath, ImportPath 및 ImportedFileChunkSize 설정을 지정하지 않아야 합니다. 이는 [BackupId](#) 속성에 해당합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## KmsKeyId(선택 사항, String)

저장된 FSx for Lustre 파일 시스템의 영구 FSx for Lustre 파일 시스템 데이터를 암호화하는 데 사용되는 AWS Key Management Service (AWS KMS) 키 ID입니다. 지정하지 않으면 FSx for Lustre의 관리형 키가 사용됩니다. SCRATCH\_1 및 SCRATCH\_2 FSx for Lustre 파일 시스템은 항

상 FSx for Lustre 관리형 키를 사용하여 유휴 상태에서 암호화됩니다. 자세한 내용은 AWS Key Management Service API 참조의 [암호화](#)를 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

FileSystemId(선택 사항, String)

기존 FSx for Lustre 파일 시스템의 ID를 지정합니다.

이 옵션을 지정하면 FsxLustreSettings의 MountDir 및 FileSystemId 설정만 사용됩니다. FsxLustreSettings의 다른 모든 설정은 무시됩니다.

**Note**

AWS Batch 스케줄러를 사용하는 경우 FSx for Lustre는 헤드 노드에서만 사용할 수 있습니다.

**Note**

파일 시스템은 포트 988, 1021, 1022 및 1023을 통한 인바운드 및 아웃바운드 TCP 트래픽을 허용하는 보안 그룹에 연결되어 있어야 합니다.

다음 중 하나를 수행하여 클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인합니다.

- 클러스터 서브넷의 CIDR 또는 접두사 목록으로 들어오고 나가는 트래픽을 허용하도록 파일 시스템의 보안 그룹을 구성합니다.

**Note**

AWS ParallelCluster 포트가 열려 있고 CIDR 또는 접두사 목록이 구성되어 있는지 확인합니다. AWS ParallelCluster CIDR 블록 또는 접두사 목록의 내용을 확인하지 않습니다.

- [SlurmQueues/Networking/SecurityGroups](#) 및 [HeadNode/Networking/SecurityGroups](#)를 사용하여 클러스터 노드의 사용자 지정 보안 그룹을 설정합니다. 클러스터와 파일 시스템 간의 트래픽을 허용하도록 사용자 지정 보안 그룹을 구성해야 합니다.

**Note**

모든 클러스터 노드가 사용자 지정 보안 그룹을 사용하는 경우 포트가 열려 AWS ParallelCluster 있는지 확인합니다. AWS ParallelCluster 소스와 대상이 제대로 구성되어 있는지 검증하지 않습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**AutoImportPolicy(선택 사항, String)**

FSx for Lustre 파일 시스템을 생성하면 기존 Amazon S3 객체가 파일 및 디렉터리 목록에 표시됩니다. 이 속성을 사용하여, 연결된 Amazon S3 버킷에서 객체를 추가하거나 수정할 때 FSx for Lustre가 파일 및 디렉터리 목록을 최신 상태로 유지하는 방법을 선택할 수 있습니다. AutoImportPolicy는 다음 값을 가질 수 있습니다.

- NEW - 자동 가져오기가 켜져 있습니다. FSx for Lustre가 현재 FSx for Lustre 파일 시스템에 존재하지 않는 연결된 Amazon S3 버킷에 추가된 새 객체의 디렉터리 목록을 자동으로 가져옵니다.
- NEW\_CHANGED - 자동 가져오기가 켜져 있습니다. FSx for Lustre가 이 옵션을 선택한 후 Amazon S3 버킷에 추가된 새 객체 및 Amazon S3 버킷에서 변경된 기존 객체의 파일 및 디렉터리 목록을 자동으로 가져옵니다.
- NEW\_CHANGED\_DELETED - 자동 가져오기가 켜져 있습니다. FSx for Lustre는 Amazon S3 버킷에 추가된 새 객체, Amazon S3 버킷에서 변경된 기존 객체, Amazon S3 버킷에서 삭제된 객체의 파일 및 디렉터리 목록을 자동으로 가져옵니다.

**Note**

NEW\_CHANGED\_DELETED에 대한 지원이 AWS ParallelCluster 버전 3.1.1에서 추가되었습니다.

AutoImportPolicy가 지정되어 있지 않으면 자동 가져오기가 꺼집니다. FSx for Lustre는 파일 시스템이 생성될 때 링크된 Amazon S3 버킷의 파일 및 디렉터리 목록만 업데이트합니다. FSx for Lustre는 이 옵션을 선택한 후 새 개체 또는 변경된 개체에 대한 파일 및 디렉터리 목록을 업데이트하지 않습니다.

자세한 내용은 Amazon FSx for Lustre 사용 설명서의 [S3 버킷에서 업데이트 자동 가져오기를](#) 참조하세요.

**Note**

이 파라미터는 PERSISTENT\_2 배포 유형을 사용하는 파일 시스템에서 지원되지 않습니다. 데이터 리포지토리 연결을 구성하는 방법에 대한 지침은 Amazon FSx for Lustre 사용 설명서의 [S3 버킷에 파일 시스템 연결](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**DriveCacheType**(선택 사항, String)

파일 시스템에 SSD 드라이브 캐시가 있는지 지정합니다. 이는 StorageType 설정이 HDD로 설정되어 있고, DeploymentType 설정이 PERSISTENT\_1로 설정되어 있을 때만 설정할 수 있습니다. 이는 [DriveCacheType](#) 속성에 해당합니다. 자세한 내용은 Amazon FSx for Lustre 사용 설명서의 [FSx for Lustre 배포 옵션](#)을 참조하세요.

유일한 유효 값은 READ입니다. SSD 드라이브 캐시를 비활성화하려면 DriveCacheType 설정을 지정하지 마세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**StorageType**(선택 사항, String)

생성 중인 FSx for Lustre 파일 시스템의 스토리지 유형을 설정합니다. 유효 값은 SSD 및 HDD입니다.

- 솔리드 스테이트 드라이브 스토리지를 사용하려면 SSD를 설정합니다.
- 하드 디스크 드라이브 스토리지를 사용하려면 HDD로 설정합니다. HDD는 PERSISTENT 배포 유형에서 지원됩니다.

기본값은 SSD입니다. 자세한 내용은 Amazon FSx for Windows 사용 설명서의 [스토리지 유형 옵션](#) 및 Amazon FSx for Lustre 사용 설명서의 [다중 스토리지 옵션](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**DeletionPolicy**(선택 사항, String)

클러스터에서 파일 시스템을 제거하거나 클러스터를 삭제할 때 파일 시스템을 유지할지 또는 삭제할지 여부를 지정합니다. 지원되는 값은 Delete 및 Retain입니다. 기본값은 Delete입니다.

[DeletionPolicy](#)가 Delete로 설정된 경우 클러스터가 삭제되거나 클러스터 업데이트로 파일 시스템이 제거되면 관리 파일 시스템이 해당 데이터와 함께 삭제됩니다.

자세한 정보는 [공유 스토리지](#)를 참조하세요.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**Note**

DeletionPolicy AWS ParallelCluster 버전 3.3.0부터 지원됩니다.

### DataRepositoryAssociations(선택 사항, String)

DRA 목록 (파일 시스템당 최대 8개)

각 데이터 리포지토리 연결에는 고유한 Amazon FSx 파일 시스템 디렉터리 및 이와 연결된 고유한 S3 버킷 또는 접두사가 있어야 합니다.

[ExportPath](#)DRA와 [ImportPath](#)를 FsxLustreSettings 동시에 사용할 수는 없습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Name(필수, String)

DRA의 이름. 설정을 업데이트할 때 이 이름을 사용합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### BatchImportMetaDataOnCreate(선택 사항, Boolean)

데이터 리포지토리 연결이 생성된 후 메타데이터를 가져오기 위한 데이터 리포지토리 가져오기 태스크를 실행해야 하는지 여부를 나타내는 부울 플래그입니다. 이 플래그가 true로 설정된 경우 태스크가 실행됩니다.

기본 값: false

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### DataRepositoryPath(필수, String)

파일 시스템에 연결될 Amazon S3 데이터 리포지토리의 경로입니다. 경로는 s3://myBucket/myPrefix/ 형식의 S3 버킷 또는 접두사일 수 있습니다. 이 경로는 S3 데이터 리포지토리에서 파일을 가져오거나 내보낼 위치를 지정합니다.

다른 DRA와 겹칠 수 없습니다.



패턴:  $^{\wedge}[\wedge\{3,4357\}]^{\$}$

최소: 3

최대: 4357

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

FileSystemPath(필수, String)

DataRepositoryPath와 1-1로 매핑될 상위 수준 디렉터리(예: /ns1/) 또는 하위 디렉터리(예: /ns1/subdir/)를 가리키는 Amazon FSx for Lustre 파일 시스템의 경로입니다. 이름 앞에 슬래시가 있어야 합니다. 두 개의 데이터 리포지토리 연결에 중복되는 파일 시스템 경로가 있을 수 없습니다. 예를 들어, 데이터 리포지토리가 파일 시스템 경로 /ns1/과 연결된 경우 다른 데이터 리포지토리를 파일 시스템 경로 /ns1/ns2와 연결할 수 없습니다.

이 경로는 파일 시스템에서 파일을 내보내거나 가져올 위치를 지정합니다. 이 파일 시스템 디렉터리는 하나의 Amazon S3 버킷에만 연결할 수 있으며 다른 S3 버킷은 디렉터리에 연결할 수 없습니다.

다른 DRA와 겹칠 수 없음

#### Note

슬래시(/)만 파일 시스템 경로로 지정하는 경우 파일 시스템에 하나의 데이터 리포지토리만 연결할 수 있습니다. 파일 시스템과 연결된 첫 번째 데이터 저장소의 파일 시스템 경로로 /"만 지정할 수 있습니다.

패턴:  $^{\wedge}[\wedge\{1,4096\}]^{\$}$

최소: 1

최대: 4096

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

ImportedFileChunkSize(선택 사항, Integer)

데이터 리포지토리에서 가져온 파일의 경우 이 값은 단일 물리적 디스크에 저장된 파일당 스트라이프 수 및 최대 데이터 양(MiB)을 결정합니다. 단일 파일을 스트라이프할 수 있는 최대 디스크 수는 파일 시스템 또는 캐시를 구성하는 총 디스크 수에 따라 제한됩니다.

체크 크기 기본값은 1,024MiB(1GiB)이며 최대 512,000MiB(500GiB)까지 가능합니다. Amazon S3 객체의 크기는 최대 5TB입니다.

최소: 1

최대: 4096

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

AutoExportPolicy(선택 사항, Array of strings)

목록에는 다음 값 중 하나 이상이 포함될 수 있습니다.

- NEW - 새 파일과 디렉토리는 파일 시스템에 추가될 때 자동으로 데이터 리포지토리로 내보내집니다.
- CHANGED - 파일 시스템의 파일 및 디렉토리에 대한 변경 사항은 자동으로 데이터 리포지토리로 내보내집니다.
- DELETED - 파일과 디렉토리는 파일 시스템에서 삭제될 때 데이터 리포지토리에서 자동으로 삭제됩니다.

AutoExportPolicy에 대한 이벤트 유형의 조합을 정의할 수 있습니다.

최대: 3

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

AutoImportPolicy(선택 사항, Array of strings)

목록에는 다음 값 중 하나 이상이 포함될 수 있습니다.

- NEW - Amazon FSx는 현재 FSx 파일 시스템에 존재하지 않는 연결된 S3 버킷에 추가된 파일의 메타데이터를 자동으로 가져옵니다.
- CHANGED - 데이터 리포지토리에서 파일이 변경될 때 Amazon FSx가 파일 메타데이터를 자동으로 업데이트하고 파일 시스템의 기존 파일 콘텐츠를 무효화합니다.
- DELETED - 데이터 리포지토리에서 파일이 삭제될 때 Amazon FSx가 파일 시스템의 해당 파일을 자동으로 삭제합니다.

AutoImportPolicy에 대한 이벤트 유형의 조합을 정의할 수 있습니다.

최대: 3

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## FsxOntapSettings

### Note

[StorageType](#)에 FsxOntap가 지정되어 있으면 FsxOntapSettings를 정의해야 합니다.

(선택 사항) FSx for ONTAP 파일 시스템 설정

[FsxOntapSettings](#):

[VolumeId](#): *string*

### FsxOntapSettings 속성

VolumeId(필수, String)

기존 FSx for ONTAP 시스템의 볼륨 ID를 지정합니다.

### Note

- AWS Batch 스케줄러를 사용하는 경우 ONTAP용 FSx는 헤드 노드에서만 사용할 수 있습니다.
- FSx for ONTAP 배포 유형이 Multi-AZ인 경우 헤드 노드 서브넷의 라우팅 테이블이 제대로 구성되었는지 확인하세요.
- ONTAP용 FSx 지원은 버전 3.2.0에 추가되었습니다. AWS ParallelCluster
- 파일 시스템은 포트 111, 635, 2049 및 4046을 통한 인바운드 및 아웃바운드 TCP 및 UDP 트래픽을 허용하는 보안 그룹에 연결되어 있어야 합니다.

다음 작업 중 하나를 수행하여 클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인합니다.

- 클러스터 서브넷의 CIDR 또는 접두사 목록으로 들어오고 나가는 트래픽을 허용하도록 파일 시스템의 보안 그룹을 구성합니다.

**Note**

AWS ParallelCluster 포트가 열려 있고 CIDR 또는 접두사 목록이 구성되어 있는지 확인합니다. AWS ParallelCluster CIDR 블록 또는 접두사 목록의 내용을 확인하지 않습니다.

- [SlurmQueues/Networking/SecurityGroups](#) 및 [HeadNode/Networking/SecurityGroups](#)를 사용하여 클러스터 노드의 사용자 지정 보안 그룹을 설정합니다. 클러스터와 파일 시스템 간의 트래픽을 허용하도록 사용자 지정 보안 그룹을 구성해야 합니다.

**Note**

모든 클러스터 노드가 사용자 지정 보안 그룹을 사용하는 경우 포트가 열려 AWS ParallelCluster 있는지만 확인합니다. AWS ParallelCluster 소스와 대상이 제대로 구성되었는지 검증하지 않습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**FsxOpenZfsSettings****Note**

[StorageType](#)에 FsxOpenZfs가 지정되어 있으면 FsxOpenZfsSettings를 정의해야 합니다.

(선택 사항) FSx for OpenZFS 파일 시스템 설정

[FsxOpenZfsSettings](#):

[VolumeId](#): *string*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**FsxOpenZfsSettings** 속성

VolumeId(필수, String)

기존 FSx for OpenZFS 시스템의 볼륨 ID를 지정합니다.

**Note**

- AWS Batch 스케줄러를 사용하는 경우 OpenZFS용 FSX는 헤드 노드에서만 사용할 수 있습니다.
- OpenZFS를 위한 FSX에 대한 지원은 버전 3.2.0에서 추가되었습니다. AWS ParallelCluster
- 파일 시스템은 포트 111, 2049, 20001, 20002 및 20003을 통한 인바운드 및 아웃바운드 TCP 및 UDP 트래픽을 허용하는 보안 그룹에 연결되어 있어야 합니다.

다음 중 하나를 수행하여 클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인합니다.

- 클러스터 서브넷의 CIDR 또는 접두사 목록으로 들어오고 나가는 트래픽을 허용하도록 파일 시스템의 보안 그룹을 구성합니다.

**Note**

AWS ParallelCluster 포트가 열려 있고 CIDR 또는 접두사 목록이 구성되어 있는지 확인합니다. AWS ParallelCluster CIDR 블록 또는 접두사 목록의 내용을 확인하지 않습니다.

- [SlurmQueues/Networking/SecurityGroups](#) 및 [HeadNode/Networking/SecurityGroups](#)를 사용하여 클러스터 노드의 사용자 지정 보안 그룹을 설정합니다. 클러스터와 파일 시스템 간의 트래픽을 허용하도록 사용자 지정 보안 그룹을 구성해야 합니다.

**Note**

모든 클러스터 노드가 사용자 지정 보안 그룹을 사용하는 경우 포트가 열려 AWS ParallelCluster 있는지 확인합니다. AWS ParallelCluster 소스와 대상이 제대로 구성되었는지 검증하지 않습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**FileCacheSettings****Note**

[StorageType](#)에 FileCache가 지정되어 있으면 FileCacheSettings를 정의해야 합니다.

## (선택 사항) 파일 캐시에 대한 설정

### FileCacheSettings:

FileCacheId: *string*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### FileCacheSettings 속성

FileCacheId(필수, String)

기존 파일 캐시의 파일 캐시 ID를 지정합니다.

#### Note

- 파일 캐시는 AWS Batch 스케줄러를 지원하지 않습니다.
- 파일 캐시 지원 AWS ParallelCluster 기능은 버전 3.7.0에 추가되었습니다.
- 파일 시스템은 포트 988을 통한 인바운드 및 아웃바운드 TCP 트래픽을 허용하는 보안 그룹에 연결되어 있어야 합니다.

다음 중 하나를 수행하여 클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인합니다.

- 클러스터 서브넷의 CIDR 또는 접두사 목록으로 들어오고 나가는 트래픽을 허용하도록 파일 캐시의 보안 그룹을 구성합니다.

#### Note

AWS ParallelCluster 포트가 열려 있고 CIDR 또는 접두사 목록이 구성되어 있는지 확인합니다. AWS ParallelCluster CIDR 블록 또는 접두사 목록의 내용을 확인하지 않습니다.

- [SlurmQueues/Networking/SecurityGroups](#) 및 [HeadNode/Networking/SecurityGroups](#)를 사용하여 클러스터 노드의 사용자 지정 보안 그룹을 설정합니다. 클러스터와 파일 시스템 간의 트래픽을 허용하도록 사용자 지정 보안 그룹을 구성해야 합니다.

**Note**

모든 클러스터 노드가 사용자 지정 보안 그룹을 사용하는 경우 포트가 열려 AWS ParallelCluster 있는지만 확인합니다. AWS ParallelCluster 소스와 대상이 제대로 구성되었는지 검증하지 않습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**Iam** 섹션

(선택 사항) 클러스터의 IAM 속성을 지정합니다.

```
Iam:
  Roles:
    LambdaFunctionsRole: string
    PermissionsBoundary: string
    ResourcePrefix: string
```

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

**Iam** 속성

PermissionsBoundary(선택 사항, String)

AWS ParallelCluster에서 생성한 모든 역할의 권한 경계로 사용할 IAM 정책의 ARN 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티의 권한 범위](#)를 참조하세요. 형식은 `arn:${Partition}:iam::${Account}:policy/${PolicyName}`입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

Roles(선택 사항)

클러스터에서 사용하는 IAM 역할의 설정을 지정합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

LambdaFunctionsRole(선택 사항, String)

사용할 IAM 역할의 ARN입니다. AWS Lambda는 사용자 지정 리소스를 지원하는 모든 Lambda 함수에 연결된 기본 역할을 재정의합니다. AWS CloudFormation Lambda를 역할을 맡도록 허용된 보안 주체로 구성해야 합니다. 이 경우 사용되는 Lambda 함수의 역할을 재

정의하지 않습니다. AWS Batch형식은 `arn:${Partition}:iam:${Account}:role/${RoleName}`입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### ResourcePrefix(선택 사항)

에서 생성한 IAM 리소스의 경로 또는 이름 접두사를 지정합니다. AWS ParallelCluster

리소스 접두사는 IAM에서 지정한 이름 지정 규칙을 따라야 합니다.

- 이름은 최대 30자를 포함할 수 있습니다.
- 이름은 슬래시(/) 문자가 없는 문자열만 사용할 수 있습니다.
- 경로는 최대 512자까지 포함할 수 있습니다.
- 경로는 슬래시(/)로 시작하고 끝나야 합니다. 시작 슬래시와 끝 슬래시(/) 사이에 여러 개의 슬래시(/)를 포함할 수 있습니다.
- 경로와 이름 `/path/name`을 조합할 수 있습니다.

이름을 지정합니다.

```
Iam:
  ResourcePrefix: my-prefix
```

경로를 지정합니다.

```
Iam:
  ResourcePrefix: /org/dept/team/project/user/
```

경로와 이름을 지정합니다.

```
Iam:
  ResourcePrefix: /org/dept/team/project/user/my-prefix
```

`/my-prefix`를 지정하면 오류가 반환됩니다.

```
Iam:
  ResourcePrefix: /my-prefix
```

구성 오류가 반환됩니다. 경로에는 `/`가 두 개 있어야 합니다. 접두사 자체로는 `/`를 가질 수 없습니다.



업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## LoginNodes 섹션

### Note

AWS ParallelCluster 버전 3.7.0에 대한 LoginNodes 지원이 추가되었습니다.

(선택 사항) 로그인 노드 풀의 구성을 지정합니다.

```

LoginNodes:
  Pools:
    - Name: string
      Count: integer
      InstanceType: string
      GracetimePeriod: integer
      Image:
        CustomAmi: string
      Ssh:
        KeyName: string
      Networking:
        SubnetIds:
          - string
        SecurityGroups:
          - string
        AdditionalSecurityGroups:
          - string
      Iam:
        InstanceRole: string
        InstanceProfile: string
        AdditionalIamPolicies:
          - Policy: string
  
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

## LoginNodes 속성

### Pools 속성

리소스 구성이 동일한 로그인 노드 그룹을 정의합니다. 풀 한 개만 지정할 수 있습니다.

Pools:

- Name: *string*
- Count: *integer*
- InstanceType: *string*
- GracetimePeriod: *integer*
- Image:
  - CustomAmi: *string*
- Ssh:
  - KeyName: *string*
- Networking:
  - SubnetIds:
    - *string*
  - SecurityGroups:
    - *string*
  - AdditionalSecurityGroups:
    - *string*
- Iam:
  - InstanceRole: *string*
  - InstanceProfile: *string*
  - AdditionalIamPolicies:
    - Policy: *string*

## Name(필수 String)

LoginNodes 풀 이름을 지정합니다. 이는 LoginNodes 리소스에 태그를 지정하는 데 사용됩니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Count(필수 Integer)

활성 상태를 유지할 로그인 노드 수를 지정합니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## InstanceType(필수 String)

로그인 노드에 사용되는 Amazon EC2 인스턴스 유형을 지정합니다. 인스턴스 유형의 아키텍처는 Slurm InstanceType 설정에 사용된 아키텍처와 동일해야 합니다.

업데이트 정책: 로그인 노드 풀이 중지된 경우 이 설정을 변경할 수 있습니다.

## GracetimePeriod(선택 사항 Integer)

로그인한 사용자에게 로그인 노드의 서비스 해제를 알리는 알림과 실제 중지 이벤트 사이의 최소 경과 시간(분)을 지정합니다. GracetimePeriod에 유효한 값은 3분에서 최대 120분 사이입니다. 기본값은 60분입니다.

### Note

트리거링 이벤트에는 여러 서비스 간의 상호 작용이 포함됩니다. AWS 경우에 따라 네트워크 지연 및 정보 전파에 시간이 걸릴 수 있으며, 서비스의 내부 지연으로 인해 유예 기간이 예상보다 오래 걸릴 수 있습니다. AWS

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

## Image(선택 사항)

로그인 노드의 이미지 구성을 정의합니다.

Image:  
CustomAmi: *String*

## CustomAmi(선택 사항 String)

로그인 노드를 프로비저닝하는 데 사용되는 사용자 지정 AMI를 지정합니다. 지정하지 않을 경우 값은 [HeadNode 섹션](#)에 지정된 값으로 디폴트됩니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Ssh(선택 사항)

로그인 노드의 ssh 구성을 정의합니다.

Ssh:  
KeyName: *string*

## KeyName(선택 사항 String)

로그인 노드에 로그인하는 데 사용되는 ssh 키를 지정합니다. 지정하지 않을 경우 값은 [HeadNode 섹션](#)에 지정된 값으로 디폴트됩니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Networking(필수)

```

Networking:
  SubnetIds:
    - string
  SecurityGroups:
    - string
  AdditionalSecurityGroups:
    - string

```

### SubnetIds(필수 [String])

로그인 노드 풀을 프로비저닝하는 기존 서브넷의 ID 하나의 서브넷만 정의할 수 있습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### SecurityGroups(선택 사항 [String])

로그인 노드 풀에 사용할 보안 그룹의 목록입니다. 지정된 보안 그룹이 없는 경우 보안 그룹을 AWS ParallelCluster 생성하십시오.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### AdditionalSecurityGroups(선택 사항 [String])

로그인 노드 풀에 사용할 추가 보안 그룹의 목록입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Iam(선택 사항)

로그인 노드에서 클러스터의 기본 인스턴스 역할 또는 인스턴스 프로파일을 재정의하는 데 사용할 인스턴스 역할 또는 인스턴스 프로파일을 지정합니다.

```

Iam:
  InstanceRole: string
  InstanceProfile: string
  AdditionalIamPolicies:
    - Policy: string

```

### InstanceProfile(선택 사항 String)

기본 로그인 노드 인스턴스 프로파일을 재정의할 인스턴스 프로파일을 지정합니다. InstanceProfile 및 InstanceRole를 둘 다 지정할 수 없습니다. 형식은

arn:Partition:iam::Account:instance-profile/*InstanceProfileName*입니다. 이를 지정하면 InstanceRole 및 AdditionalIamPolicies 설정을 지정할 수 없습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### InstanceRole(선택 사항 String)

기본 로그인 노드 인스턴스 역할을 재정의할 인스턴스 역할을 지정합니다.

InstanceProfile 및 InstanceRole를 둘 다 지정할 수 없습니다. 형식은

arn:Partition:iam::Account:role/RoleName입니다. 이를 지정하면 S3Access 및 AdditionalIamPolicies 설정을 지정할 수 없습니다. 이를 지정하면 InstanceProfile 및 AdditionalIamPolicies 설정을 지정할 수 없습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### AdditionalIamPolicies(선택 사항)

AdditionalIamPolicies:

- Policy: *string*

### IAM 정책 Amazon 리소스 이름(ARN)

Amazon EC2에 대한 IAM 정책의 Amazon 리소스 이름(ARN) 목록을 쉼표로 구분하여 지정합니다. 이 목록은 에서 요구하는 권한 외에도 로그인 노드에 사용되는 루트 역할에 연결됩니다 AWS ParallelCluster.

IAM 정책 이름과 해당 ARN은 서로 다릅니다. 이름은 사용할 수 없습니다.

이를 지정하면 InstanceProfile 및 InstanceRole 설정을 지정할 수 없습니다. 필요한 권한에 AdditionalIamPolicies 추가되고 필요한 모든 권한이 InstanceRole 포함되어야 AdditionalIamPolicies 하므로 사용하는 것이 좋습니다. AWS ParallelCluster 기능이 추가됨에 따라 필요한 권한은 종종 릴리스마다 변경됩니다.

기본값이 없습니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

### Policy(필수 [String])

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## Monitoring 섹션

(선택 사항) 클러스터의 모니터링 설정을 지정합니다.

Monitoring:

Logs:

CloudWatch:

Enabled: *boolean*

RetentionInDays: *integer*

DeletionPolicy: *string*

Rotation:

Enabled: *boolean*

Dashboards:

CloudWatch:

Enabled: *boolean*

DetailedMonitoring: *boolean*

Alarms:

Enabled: *boolean*

업데이트 정책: 이 설정은 업데이트 중에 분석되지 않습니다.

### Monitoring 속성

#### Logs(선택 사항)

클러스터의 로그 설정

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

#### CloudWatch(선택 사항)

클러스터의 CloudWatch 로그 설정.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

Enabled(필수, Boolean)

true경우 클러스터 로그가 CloudWatch 로그로 스트리밍됩니다. 기본값은 true입니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

RetentionInDays(선택 사항, Integer)

로그에 로그 이벤트를 보관하는 CloudWatch 기간 (일) 기본값은 180입니다. 지원되는 값은 0, 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653입니다. 값이 0이면 기본 CloudWatch 로그 보존 설정이 사용됩니다. 즉, 만료되지 않습니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

DeletionPolicy(선택 사항, String)

클러스터가 삭제될 때 CloudWatch 로그의 로그 이벤트를 삭제할지 여부를 나타냅니다. 가능한 값은 Delete와 Retain입니다. 기본값은 Retain입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

Rotation(선택 사항)

클러스터의 로그 로테이션 설정

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

Enabled(필수, Boolean)

true면 로그 로테이션이 활성화됩니다. 기본값은 true입니다. AWS ParallelCluster 구성된 로그 파일이 특정 크기에 도달하면 파일이 순환되고 단일 백업이 유지됩니다. 자세한 정보는 [AWS ParallelCluster 로그 교체 구성](#)을 참조하세요.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

Dashboards(선택 사항)

클러스터의 대시보드 설정

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

CloudWatch(선택 사항)

클러스터의 CloudWatch 대시보드 설정.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

Enabled(필수, Boolean)

true인 경우 CloudWatch 대시보드가 활성화됩니다. 기본값은 true입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

DetailedMonitoring(선택 사항, Boolean)

로 true 설정하면 컴퓨팅 플릿 Amazon EC2 인스턴스에 대한 세부 모니터링이 활성화됩니다. 활성화되면 Amazon EC2 콘솔에 1분 간격으로 인스턴스를 모니터링하기 위한 그래프가 표시됩니다. 이 기능을 활성화하면 추가 비용이 발생합니다. 기본값은 false입니다.

자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화](#)를 참조하세요.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

#### Note

DetailedMonitoring AWS ParallelCluster 버전 3.6.0부터 추가되었습니다.

## Alarms(선택 사항)

CloudWatch 클러스터에 대한 경보.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

### Enabled(선택 사항)

true인 경우 클러스터에 대한 CloudWatch 경보가 생성됩니다. 기본값은 true입니다.

업데이트 정책: 이 설정은 업데이트 중에 변경할 수 있습니다.

#### Note

AWS ParallelCluster 버전 3.8.0부터 헤드 노드에 대해 Amazon EC2 Health Check, CPU/메모리/디스크 사용률 및 기타 모든 경보를 포함한 복합 경보와 같은 경보가 생성됩니다.

## Tags 섹션

(선택 사항), Array는 모든 클러스터 리소스에서 사용하고 모든 클러스터 리소스에 전파되는 태그를 정의합니다. AWS CloudFormation 자세한 내용을 알아보려면AWS CloudFormation 사용 설명서의 [AWS CloudFormation 리소스 태그](#)를 참조하세요.

### Tags:

- Key: *string*
- Value: *string*

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.



## Tags 속성

Key(필수, String)

태그의 이름을 정의합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

Value(필수, String)

태그의 값을 정의합니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## AdditionalPackages 섹션

(선택 사항) 설치할 추가 패키지를 식별하는 데 사용됩니다.

```
AdditionalPackages:
  IntelSoftware:
    IntelHpcPlatform: boolean
```

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## IntelSoftware

(선택 사항) Intel 선택 솔루션의 구성을 정의합니다.

```
IntelSoftware:
  IntelHpcPlatform: boolean
```

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## IntelSoftware 속성

IntelHpcPlatform(선택 사항, Boolean)

true면 Intel Parallel Studio에 대한 최종 사용자 라이선스 계약에 동의했음을 나타냅니다. 이렇게 하면 Intel Parallel Studio가 헤드 노드에 설치되고 컴퓨팅 노드와 공유됩니다. 이로 인해 헤드 노드의 부트스트랩을 수행하는 데 걸리는 시간이 몇 분 더 추가됩니다. IntelHpcPlatform 설정은 CentOS 7에서만 지원됩니다.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

## DirectoryService 섹션

### Note

에 대한 지원은 AWS ParallelCluster 버전 3.1.1에 DirectoryService 추가되었습니다.

(선택 사항) 다중 사용자 액세스를 지원하는 클러스터의 디렉토리 서비스 설정

AWS ParallelCluster [시스템 보안 서비스 데몬 \(SSSD\)](#) 에서 지원하는 경량 디렉터리 액세스 프로토콜 (LDAP) 을 통해 Active Directory (AD) 를 사용하여 클러스터에 대한 다중 사용자 액세스를 지원하는 권한을 관리합니다. 자세한 내용은 AWS Directory Service 관리 가이드의 [AWS Directory Service란 무엇입니까?](#)를 참조하세요.

잠재적으로 민감한 정보가 암호화된 채널을 통해 전송되도록 하려면 TLS/SSL을 통한 LDAP(줄여서 LDAPS)를 사용하는 것이 좋습니다.

#### DirectoryService:

```

DomainName: string
DomainAddr: string
PasswordSecretArn: string
DomainReadOnlyUser: string
LdapTlsCaCert: string
LdapTlsReqCert: string
LdapAccessFilter: string
GenerateSshKeysForUsers: boolean
AdditionalSssdConfigs: dict

```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

## DirectoryService 속성

### Note

인터넷에 액세스할 수 없는 단일 AWS ParallelCluster 서브넷에서 사용하려는 경우 추가 요구 사항은 을 참조하십시오. [AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서](#)

DomainName(필수, String)

ID 정보로 사용하는 액티브 디렉터리(AD) 도메인

DomainName은 Fully Qualified Domain Name(FQDN)과 LDAP Distinguished Name(DN) 형식을 모두 허용합니다.

- FQDN 예제: corp.*example*.com
- LDAP DN 예제: DC=*corp*,DC=*example*,DC=*com*

이 속성은 ldap\_search\_base라는 sssd-ldap 파라미터에 대응합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### DomainAddr(필수, String)

LDAP 서버로 사용되는 AD 도메인 컨트롤러를 가리키는 URI 또는 URI입니다. URI는 ldap\_uri라는 SSSD-LDAP 파라미터에 대응합니다. 값은 쉼표로 구분된 URI 문자열일 수 있습니다. LDAP를 사용하려면 각 URI의 시작 부분에 ldap://를 추가해야 합니다.

예제 값:

```
ldap://192.0.2.0,ldap://203.0.113.0          # LDAP
ldaps://192.0.2.0,ldaps://203.0.113.0      # LDAPS without support for certificate
verification
ldaps://abcdef01234567890.corp.example.com # LDAPS with support for certificate
verification
192.0.2.0,203.0.113.0                       # AWS ParallelCluster uses LDAPS by
default
```

인증서 확인과 함께 LDAPS를 사용하는 경우 URI는 호스트 이름이어야 합니다.

인증서 확인 또는 LDAP 없이 LDAPS를 사용하는 경우 URI는 호스트 이름 또는 IP 주소일 수 있습니다.

TLS/SSL을 통한 LDAP(LDAPS)를 사용하여 암호와 기타 민감한 정보가 암호화되지 않은 채널을 통해 전송되지 않도록 하세요. AWS ParallelCluster 이 프로토콜을 찾지 못하면 각 URI 또는 호스트 이름의 시작 부분에 ldaps://이 추가됩니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### PasswordSecretArn(필수, String)

DomainReadOnlyUser일반 텍스트 비밀번호가 포함된 AWS Secrets Manager 비밀번호의 Amazon 리소스 이름 (ARN). 암호의 내용은 ldap\_default\_authtok라는 SSSD-LDAP 파라미터에 대응합니다.

**Note**

AWS Secrets Manager 콘솔을 사용하여 암호를 생성할 때는 “기타 유형의 암호”를 선택하고 일반 텍스트를 선택한 다음 암호 텍스트만 암호에 포함해야 합니다.

암호를 만드는 AWS Secrets Manager 데 사용하는 방법에 대한 자세한 내용은 암호 [만들기](#)를 참조하십시오. AWS Secrets Manager

LDAP 클라이언트는 ID 정보를 요청할 때 DomainReadOnlyUser로서 암호를 사용하여 AD 도메인을 인증합니다.

사용자에게 [DescribeSecret](#)에 대한 권한이 있는 경우 PasswordSecretArn의 유효성이 검사됩니다. 지정된 비밀이 존재할 경우 PasswordSecretArn가 유효합니다. 사용자 IAM 정책이 DescribeSecret을 포함하지 않는 경우 PasswordSecretArn가 검증되지 않고 경고 메시지가 표시됩니다. 자세한 내용은 [기본 AWS ParallelCluster pcluster 사용자 정책](#) 항목을 참조하세요.

시크릿 값이 변경되더라도 클러스터는 자동으로 업데이트되지 않습니다. 새 암호 값에 맞게 클러스터를 업데이트하려면 [the section called “pcluster update-compute-fleet”](#) 명령으로 컴퓨팅 플릿을 중지한 다음 헤드 노드 내에서 다음 명령을 실행해야 합니다.

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

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

DomainReadOnlyUser(필수, String)

클러스터 사용자 로그인을 인증할 때 AD 도메인에 ID 정보를 쿼리하는 데 사용되는 ID입니다. 이는 ldap\_default\_bind\_dn라는 SSSD-LDAP 파라미터에 대응합니다. 이 값에는 AD ID 정보를 사용하세요.

노드에 있는 특정 LDAP 클라이언트에서 요구하는 형식으로 ID를 지정합니다.

- MicrosoftAD:

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

- SimpleAD:

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### LdapTlsCaCert(선택 사항, String)

도메인 컨트롤러용 인증서를 발급한 인증 체인의 모든 인증 기관에 대한 인증서를 포함하는 인증서 번들의 절대 경로입니다. 이는 `ldap_tls_cacert`라는 SSSD-LDAP 파라미터에 대응합니다.

인증서 번들은 PEM 형식의 개별 인증서를 연결하여 구성된 파일로, Windows에서는 DER Base64 형식이라고도 합니다. LDAP 서버 역할을 하는 AD 도메인 컨트롤러의 ID를 확인하는 데 사용됩니다.

AWS ParallelCluster 노드에 인증서를 처음 배치하는 일은 책임지지 않습니다. 클러스터 관리자는 클러스터를 생성한 후 헤드 노드에 인증서를 수동으로 구성하거나 [부트스트랩 스크립트](#)를 사용할 수 있습니다. 또는 헤드 노드에 구성된 인증서가 포함된 Amazon Machine Image(AMI)를 사용할 수 있습니다.

[Simple AD](#)는 LDAPS를 지원하지 않습니다. Simple AD 디렉토리를 통합하는 [방법을 알아보려면 AWS 보안 블로그에서 Simple AD용 LDAPS 엔드포인트를 구성하는 방법](#)을 참조하십시오. AWS ParallelCluster

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### LdapTlsReqCert(선택 사항, String)

TLS 세션에서 서버 인증서에 대해 수행할 검사를 지정합니다. 이는 `ldap_tls_reqcert`라는 SSSD-LDAP 파라미터에 대응합니다.

유효한 값: never, allow, try, demand 및 hard.

never, allow 및 try는 인증서에 문제가 발견되더라도 연결을 계속할 수 있게 합니다.

demand 및 hard는 인증서에 문제가 없는 경우 통신을 계속할 수 있게 합니다.

클러스터 관리자가 인증서 유효성 검사에 성공할 필요가 없는 값을 사용하는 경우 관리자에게 경고 메시지가 반환됩니다. 보안상의 이유로 인증서 확인을 비활성화하지 않는 것이 좋습니다.

기본값은 hard입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

### LdapAccessFilter(선택 사항, String)

디렉터리 액세스를 사용자 하위 집합으로 제한하는 필터를 지정합니다. 이 속성은 `ldap_access_filter`라는 SSSD-LDAP 파라미터에 대응합니다. 이를 사용하여 많은 사용자를 지원하는 AD로 쿼리를 제한할 수 있습니다.

이 필터는 클러스터에 대한 사용자 액세스를 차단할 수 있습니다. 그러나 차단된 사용자의 검색 가능성에는 영향을 미치지 않습니다.

이 속성이 설정되어 있으면 SSSD 파라미터 `access_provider`은 AWS ParallelCluster에 의해 내부적으로 `ldap`로 설정되며, [DirectoryService/AdditionalSssdConfigs](#) 설정에 의해 변경되어서는 안 됩니다.

이 속성을 생략하고 [DirectoryService/AdditionalSssdConfigs](#)에 사용자 지정된 사용자 액세스를 지정하지 않으면 디렉터리의 모든 사용자가 클러스터에 액세스할 수 있습니다.

예:

```

"!(cn=SomeUser*)" # denies access to every user with alias starting with "SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias starting with "SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".

```

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

`GenerateSshKeysForUsers`(선택 사항, Boolean)

헤드 노드에서 초기 인증 후 클러스터 사용자를 위한 SSH 키를 즉시 AWS ParallelCluster 생성할지 여부를 정의합니다.

`true`로 설정하면 헤드 노드에서 첫 인증을 받은 후 SSH 키가 생성되어 존재하지 않는 경우 모든 사용자에게 대해 SSH 키가 생성되어 `USER_HOME_DIRECTORY/.ssh/id_rsa`에 저장됩니다.

헤드 노드에서 아직 인증되지 않은 사용자의 경우 다음과 같은 경우에 첫 번째 인증이 발생할 수 있습니다.

- 사용자가 자신의 비밀번호로 헤드 노드에 처음으로 로그인합니다.
- 헤드 노드에서 `sudoer`가 처음으로 사용자로 전환합니다: `su USERNAME`
- 헤드 노드에서 `sudoer`가 처음으로 사용자로서 명령을 실행합니다: `su -u USERNAME COMMAND`

사용자는 클러스터 헤드 노드와 컴퓨팅 노드에 대한 후속 로그인 시 SSH 키를 사용할 수 있습니다. 이를 사용하면 AWS ParallelCluster 클러스터 컴퓨팅 노드에 대한 암호 로그인이 설계상 비활성화됩니다. 사용자가 헤드 노드에 로그인하지 않은 경우 SSH 키는 생성되지 않으며 사용자는 컴퓨팅 노드에 로그인할 수 없습니다.

기본값은 `true`입니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

## AdditionalSssdConfigs(선택 사항, Dict)

클러스터 인스턴스의 SSSD 구성 파일에 기록하기 위한 SSSD 파라미터 및 값을 포함하는 키-값 쌍의 설명 SSSD 구성 파일에 대한 전체 설명은 SSSD 및 관련 구성 파일의 온 인스턴스 매뉴얼 페이지를 참조하세요.

SSSD 파라미터 및 값은 다음 목록에 설명된 대로 AWS ParallelCluster의 SSSD 구성과 호환되어야 합니다.

- `id_provider` `ldap` 내부적으로 사용하도록 설정되어 AWS ParallelCluster 있으며 수정해서는 안 됩니다.
- `access_provider` [DirectoryService](#) / [LdapAccessFilter](#) 를 지정할 AWS ParallelCluster 때 `ldap` 내부적으로 [LdapAccessFilter](#) 설정되며 이 설정을 수정해서는 안 됩니다.

[DirectoryService](#) / [LdapAccessFilter](#) 가 생략되면 그것의 `access_provider` 사양도 생략됩니다. 예를 들어 [AdditionalSssdConfigs](#) 에서 `access_provider` 를 `simple` 로 설정하면 [DirectoryService](#) / [LdapAccessFilter](#) 를 지정해서는 안 됩니다.

다음 구성 스니펫은 `AdditionalSssdConfigs` 에 대한 유효한 구성의 예제입니다.

이 예제는 SSSD 로그의 디버그 수준을 활성화하고, 검색 기준을 특정 조직 단위로 제한하고, 보안 인증 캐싱을 비활성화합니다.

```
DirectoryService:
  ...
AdditionalSssdConfigs:
  debug_level: "0xFFF0"
  ldap_search_base: OU=Users,OU=CORP,DC=corp,DC=example,DC=com
  cache_credentials: False
```

이 예제는 SSSD [simple](#) `access_provider` 의 구성을 지정합니다.

`EngineeringTeam` 의 사용자에게 디렉터리 액세스 권한이 제공됩니다. 이 경우 [DirectoryService](#) / [LdapAccessFilter](#) 를 설정해서는 안 됩니다.

```
DirectoryService:
  ...
AdditionalSssdConfigs:
  access_provider: simple
  simple_allow_groups: EngineeringTeam
```

**업데이트 정책:** 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿을 중지해야 합니다.

## DeploymentSettings 섹션

### Note

DeploymentSettings AWS ParallelCluster 버전 3.4.0부터 추가됩니다.

(선택 사항) 배포 설정 구성을 지정합니다.

```
DeploymentSettings:
  LambdaFunctionsVpcConfig:
    SecurityGroupIds
      - string
    SubnetIds
      - string
  DisableSudoAccessForDefaultUser: Boolean
  DefaultUserHome: string # 'Shared' or 'Local'
```

## DeploymentSettings 속성

### LambdaFunctionsVpcConfig

(선택 사항) VPC AWS Lambda 구성 기능을 지정합니다. 자세한 정보는 [AWS Lambda 에서 VPC 구성 AWS ParallelCluster](#)을 참조하세요.

```
LambdaFunctionsVpcConfig:
  SecurityGroupIds
    - string
  SubnetIds
    - string
```

### LambdaFunctionsVpcConfig properties

SecurityGroupIds(필수, [String])

Lambda 함수에 연결된 Amazon VPC 보안 그룹 ID의 목록

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

SubnetIds(필수, [String])

Lambda 함수에 연결된 서브넷 ID의 목록



업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**Note**

서브넷과 보안 그룹이 동일한 VPC에 있어야 합니다.

### DisableSudoAccessForDefaultUser 속성

**Note**

이 구성 옵션은 Slurm 클러스터에서만 지원됩니다.

(선택 사항) 이 True 경우 기본 사용자의 sudo 권한이 비활성화됩니다. 이는 클러스터의 모든 노드에 적용됩니다.

```
# Main DeploymentSettings section in config yaml(applyes to HN, CF and LN)
DeploymentSettings:
  DisableSudoAccessForDefaultUser: True
```

의 DisableSudoAccessForDefaultUser 값을 업데이트하려면 컴퓨팅 플릿과 모든 로그인 노드를 중지해야 합니다.

업데이트 정책: 업데이트를 위해 이 설정을 변경하려면 컴퓨팅 플릿과 로그인 노드를 중지해야 합니다.

### DefaultUserHome 속성

로 Shared 설정하면 클러스터는 기본 설정을 사용하고 다음을 통해 클러스터 전체에서 기본 사용자 디렉토리를 공유합니다/home/<default user>.

로 Local 설정하면 헤드 노드, 로그인 노드 및 컴퓨팅 노드에 각각 별도의 로컬 기본 사용자 디렉토리가 저장됩니다local/home/<default user>.

## 빌드 이미지 구성 파일

AWS ParallelCluster 버전 3은 빌드 이미지 구성 매개변수에 YAML 1.1 파일을 사용합니다. 구성 오류를 줄이려면 들여쓰기가 올바른지 확인하세요. 자세한 내용은 <https://yaml.org/spec/1.1/>에서 YAML 1.1 사양을 살펴보세요.

이러한 구성 파일은 EC2 Image Builder를 사용하여 사용자 지정 AWS ParallelCluster AMI를 구축하는 방법을 정의하는 데 사용됩니다. 사용자 지정 AMI 구축 프로세스는 [pcluster build-image](#) 명령을 사용하여 트리거됩니다. 구성 파일 예제는 [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test\\_imagebuilder\\_schema/test\\_imagebuilder\\_schema](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema)를 참조하세요.

## 주제

- [빌드 이미지 구성 파일 속성](#)
- [Build 섹션](#)
- [Image 섹션](#)
- [DeploymentSettings 섹션](#)

## 빌드 이미지 구성 파일 속성

Region(선택 사항, String)

작업에 사용할 AWS 리전 지정합니다. build-image 예를 들어 us-east-2입니다.

## Build 섹션

(필수) 이미지를 빌드할 구성을 지정합니다.

```
Build:
  Imds:
    ImdsSupport: string
    InstanceType: string
    SubnetId: string
    ParentImage: string
  Iam:
    InstanceRole: string
    InstanceProfile: string
    CleanupLambdaRole: string
    AdditionalIamPolicies:
      - Policy: string
    PermissionsBoundary: string
  Components:
    - Type: string
      Value: string
  Tags:
    - Key: string
```

```

Value: string
SecurityGroupIds:
- string
UpdateOsPackages:
Enabled: boolean

```

## Build 속성

InstanceType(필수, String)

이미지를 빌드하는 데 사용되는 인스턴스의 인스턴스 유형을 지정합니다.

SubnetId(선택 사항, String)

이미지를 빌드할 인스턴스를 프로비저닝할 기존 서브넷의 ID를 지정합니다. 제공된 서브넷에는 인터넷 액세스가 필요합니다.

### Warning

`pcluster build-image`는 기본 VPC를 사용합니다. AWS Control Tower 또는 AWS 랜딩 존을 사용하여 기본 VPC를 삭제한 경우 서브넷 ID를 지정해야 합니다.

ParentImage(필수, String)

기본 이미지를 지정합니다. 상위 이미지는 AWS ParallelCluster AMI가 아닌 이미지나 동일 버전의 공식 AWS ParallelCluster AMI일 수 있습니다. 다른 버전의 AWS ParallelCluster 공식 또는 사용자 지정 AMI를 사용할 수 없습니다 AWS ParallelCluster. 형식은 이미지 `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion`의 ARN 또는 AMI ID `ami-12345678`여야 합니다.

SecurityGroupIds(선택 사항, [String])

이미지의 보안 그룹 ID의 목록을 지정합니다.

## Imsds

### Imsds 속성

(선택 사항) Amazon EC2 ImageBuilder 빌드 및 테스트 인스턴스 메타데이터 서비스 (IMDS) 설정을 지정합니다.

**Imds:****ImdsSupport**: *string***ImdsSupport**(선택 사항, String)

Amazon ImageBuilder EC2 빌드 및 테스트 인스턴스에서 지원되는 IMDS 버전을 지정합니다. 지원되는 값은 v2.0 및 v1.0입니다. 기본값은 v2.0입니다.

ImdsSupport이 v1.0로 설정되어 있으면 IMDSv1 및 IMDSv2가 모두 지원됩니다.

ImdsSupport이 v2.0로 설정되어 있으면 IMDSv2만 지원됩니다.

자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [IMDSv2 사용](#)을 참조하십시오.

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

**Note**

AWS ParallelCluster 버전 3.7.0부터 기본값은 v2.0입니다. ImdsSupport v2.0 사용자 지정 작업 호출에서 IMDSv1을 IMDSv2로 교체하고 ImdsSupport을 v2.0로 설정하는 것이 좋습니다.

[ImdsImdsSupport](#)/에 대한 지원은 AWS ParallelCluster 버전 3.3.0에 추가되었습니다.

**Iam****Iam 속성**

(선택 사항) 이미지 빌드의 IAM 리소스를 지정합니다.

**Iam:****InstanceRole**: *string***InstanceProfile**: *string***CleanupLambdaRole**: *string***AdditionalIamPolicies**:- **Policy**: *string***PermissionsBoundary**: *string*

## InstanceProfile(선택 사항, String)

EC2 Image Builder 인스턴스의 기본 인스턴스 프로파일을 재정의할 인스턴스 프로파일을 지정합니다. InstanceProfile, InstanceRole, AdditionalIamPolicies는 함께 지정할 수 없습니다. 형식은 `arn:Partition:iam::Account:instance-profile/InstanceProfileName`입니다.

## InstanceRole(선택 사항, String)

EC2 Image Builder 인스턴스의 기본 인스턴스 역할을 재정의할 인스턴스 역할을 지정합니다. InstanceProfile, InstanceRole, AdditionalIamPolicies는 함께 지정할 수 없습니다. 형식은 `arn:Partition:iam::Account:role/RoleName`입니다.

## CleanupLambdaRole(선택 사항, String)

빌드 완료 시 빌드 아티팩트를 제거하는 AWS CloudFormation 사용자 지정 리소스를 지원하는 AWS Lambda 함수에 사용할 IAM 역할의 ARN입니다. Lambda를 역할을 맡도록 허용된 보안 주체로 구성해야 합니다. 형식은 `arn:Partition:iam::Account:role/RoleName`입니다.

## AdditionalIamPolicies(선택 사항)

사용자 지정 AMI를 생성하는 데 사용되는 EC2 Image Builder 인스턴스에 연결할 추가 IAM 정책을 지정합니다.

AdditionalIamPolicies:  
- Policy: *string*

## Policy(선택 사항, [String])

IAM 정책 목록 형식은 `arn:Partition:iam::Account:policy/PolicyName`입니다.

## PermissionsBoundary(선택 사항, String)

AWS ParallelCluster에서 생성한 모든 역할의 권한 경계로 사용할 IAM 정책의 ARN IAM 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티에 대한 권한 경계](#)를 참조하세요. 형식은 `arn:Partition:iam::Account:policy/PolicyName`입니다.

## Components

### Components 속성

(선택 사항) 에서 기본적으로 제공하는 ImageBuilder 구성 요소 외에 AMI 빌드 프로세스 중에 사용할 Amazon EC2 구성 요소를 지정합니다. AWS ParallelCluster이러한 구성 요소를 사용하여 AMI 빌드 프

로세스를 사용자 지정할 수 있습니다. 자세한 내용은 [AWS ParallelCluster AMI 사용자 지정](#) 항목을 참조하세요.

#### Components:

- Type: *string*  
Value: *string*

#### Type(선택 사항, String)

구성 요소의 유형-값 쌍의 유형을 지정합니다. 유형은 arn 또는 script일 수 있습니다.

#### Value(선택 사항, String)

구성 요소의 유형-값 쌍의 값을 지정합니다. 유형이 arn인 경우 이것이 EC2 Image Builder 구성 요소의 ARN입니다. 유형이 script인 경우 이것이 EC2 Image Builder 구성 요소를 생성할 때 사용할 스크립트를 가리키는 https 또는 s3 링크입니다.

## Tags

### Tags 속성

(선택 사항) AMI를 구축하는 데 사용되는 리소스에 설정할 태그 목록을 지정합니다.

#### Tags:

- Key: *string*  
Value: *string*

#### Key(선택 사항, String)

태그의 이름을 정의합니다.

#### Value(선택 사항, String)

태그의 값을 정의합니다.

## UpdateOsPackages

### UpdateOsPackages 속성


(선택 사항) AWS ParallelCluster 소프트웨어 스택을 설치하기 전에 운영 체제를 업데이트할지 여부를 지정합니다.

UpdateOsPackages:

Enabled: *boolean*

Enabled(선택 사항, Boolean)

소프트웨어를 설치하기 전에 OS가 업데이트되고 재부팅되는 경우 true AWS ParallelCluster 기본 값은 false입니다.

 Note

UpdateOsPackages가 활성화되면 커널을 포함하여 사용 가능한 모든 OS 패키지가 업데이트됩니다. 고객은 업데이트가 업데이트에 포함되지 않은 AMI 종속성과 호환되는지 확인할 책임이 있습니다.

예를 들어 커널 버전 Y.0 및 일부 구성 요소 AWS ParallelCluster 버전 Z.0과 함께 제공되는 버전 X.0용 AMI를 구축한다고 가정해 보겠습니다. 사용 가능한 업데이트에 구성 요소 Z.0에 대한 업데이트가 없는 업데이트된 커널 버전 Y.1이 포함되어 있다고 가정해 보겠습니다. UpdateOsPackages를 활성화하기 전에 구성 요소 Z.0이 커널 Y.1을 지원하는지 확인하는 것은 사용자의 책임입니다.

## Image 섹션

(선택 사항) 이미지 빌드의 이미지 속성을 정의합니다.

Image:

Name: *string*

RootVolume:

Size: *integer*

Encrypted: *boolean*

KmsKeyId: *string*

Tags:

- Key: *string*

Value: *string*

## Image 속성

Name(선택 사항, String)

AMI 이름을 지정합니다. 지정하지 않으면 `pcluster build-image` 명령을 호출할 때 사용한 이름이 사용됩니다.

## Tags

### Tags 속성

(선택 사항) 이미지의 키-값 쌍을 지정합니다.

#### Tags:

- Key: *string*
- Value: *string*

Key(선택 사항, String)

태그의 이름을 정의합니다.

Value(선택 사항, String)

태그의 값을 정의합니다.

## RootVolume

### RootVolume 속성

(선택 사항) 이미지의 루트 볼륨 속성을 지정합니다.

#### RootVolume:

- Size: *integer*
- Encrypted: *boolean*
- KmsKeyId: *string*

Size(선택 사항, Integer)

이미지의 루트 볼륨 크기를 GiB 단위로 지정합니다. 기본 크기는 [ParentImage](#)의 크기 더하기 27GiB입니다.

Encrypted(선택 사항, Boolean)

볼륨이 암호화되는지 여부를 지정합니다. 기본값은 `false`입니다.

KmsKeyId(선택 사항, String)

볼륨을 암호화하는 데 사용되는 AWS KMS 키의 ARN을 지정합니다. 형식은 `arn:Partition:kms:Region:Account:key/KeyId`입니다.



## DeploymentSettings 섹션

(선택 사항) 배포 설정 구성을 지정합니다.

```
DeploymentSettings:
  LambdaFunctionsVpcConfig:
    SecurityGroupIds
      - string
    SubnetIds
      - string
```

### DeploymentSettings 속성

#### LambdaFunctionsVpcConfig

(선택 사항) VPC AWS Lambda 구성 기능을 지정합니다. 자세한 정보는 [AWS Lambda 에서 VPC 구성 AWS ParallelCluster](#)을 참조하세요.

```
LambdaFunctionsVpcConfig:
  SecurityGroupIds
    - string
  SubnetIds
    - string
```

#### LambdaFunctionsVpcConfig properties

SecurityGroupIds(필수, [String])

Lambda 함수에 연결된 Amazon VPC 보안 그룹 ID의 목록

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

SubnetIds(필수, [String])

Lambda 함수에 연결된 서브넷 ID의 목록

업데이트 정책: 이 설정을 변경하면 업데이트가 허용되지 않습니다.

#### Note

서브넷과 보안 그룹이 동일한 VPC에 있어야 합니다.

**Note**

DeploymentSettings AWS ParallelCluster 버전 3.4.0부터 추가되었습니다.

## AWS ParallelCluster API 레퍼런스

이 섹션에는 각 AWS ParallelCluster API 작업에 대한 설명, 구문 및 사용 예제가 나와 있습니다.

### 주제

- [buildImage](#)
- [createCluster](#)
- [deleteCluster](#)
- [deleteClusterInstances](#)
- [deleteImage](#)
- [describeCluster](#)
- [describeClusterInstances](#)
- [describeComputeFleet](#)
- [describeImage](#)
- [getClusterLog이벤트](#)
- [getClusterStack이벤트](#)
- [getImageLog이벤트](#)
- [getImageStack이벤트](#)
- [listClusters](#)
- [listClusterLog스트림](#)
- [listImageLog스트림](#)
- [listImages](#)
- [listOfficialImages](#)
- [updateCluster](#)
- [updateComputeFleet](#)

# buildImage

에서 사용자 지정 AWS ParallelCluster 이미지를 생성합니다 AWS 리전.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
POST /v3/images/custom
{
  "imageConfiguration": "string",
  "imageId": "string",
  "dryrun": boolean,
  "region": "string",
  "rollbackOnFailure": boolean,
  "supressValidators": [ "string" ],
  "validationFailureLevel": "string"
}
```

## 요청 본문

### imageConfiguration

YAML 문서로서의 이미지 구성입니다.

유형: string

필수 항목 여부: 예

### imageId

빌드할 이미지의 ID입니다.

유형: string

필수 항목 여부: 예

#### dryrun

true로 설정하면 리소스를 만들지 않고 요청 검증만 수행합니다. 이 파라미터를 사용하여 이미지 구성을 확인합니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

#### region

그러면 명령을 실행하여 이미지를 빌드할 수 있습니다. AWS 리전

유형: string

필수사항: 아니요

#### rollbackOnFailure

true로 설정하면 이미지 생성에 실패할 경우 이미지 스택 롤백이 발생합니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

#### suppressValidators

억제할 구성 유효성 검사기를 하나 이상 식별하세요.

유형: 문자열 목록

형식: (ALL | type: [A-Za-z0-9]+)

필수 여부: 아니요

#### validationFailureLevel

이미지 빌드 실패를 유발하는 최소 검증 수준입니다. 기본값은 ERROR입니다.

유형: string

유효값: INFO | WARNING | ERROR

필수 항목 여부: 아니요

## 응답 구문

```
{
  "image": {
    "imageId": "string",
    "ec2AmiInfo": {
      "amiId": "string"
    },
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "imageBuildStatus": "BUILD_IN_PROGRESS",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS"
  },
  "validationMessages": [
    {
      "id": "string",
      "type": "string",
      "level": "INFO",
      "message": "string"
    }
  ]
}
```

## 응답 본문

### image

#### imageId

이미지의 ID입니다.

유형: string

#### cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

#### cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

#### ec2 AmiInfo

ami\_id

아마존 EC2 AMI ID.

유형: string

#### imageBuildStatus

이미지 빌드 상태입니다.

유형: string

유효값: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

#### region

이미지가 내장되어 AWS 리전 있다는 것입니다.

유형: string

#### version

이미지를 만드는 데 사용된 AWS ParallelCluster 버전.

유형: string

#### validationMessages

검증 수준이 validationFailureLevel보다 낮은 메시지 목록입니다. 메시지 목록은 구성 검증 중에 수집됩니다.

id

검사기의 ID입니다.

유형: string

level

검증 수준입니다.

유형: string

유효값: INFO | WARNING | ERROR

message

확인 메시지입니다.

유형: string

type

검사기의 유형입니다.

유형: string

## 예제

### Python

요청

```
$ build_image(custom-image-id, custom-image-config.yaml)
```

200 응답

```
{
  'image': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/custom-image-id/711b76b0-af81-11ec-a29f-0ee549109f1f',
    'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
    'image_build_status': 'BUILD_IN_PROGRESS',
    'image_id': 'custom-image-id',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

# createCluster

AWS 리전에서 관리형 클러스터를 생성하세요.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
POST /v3/clusters
{
  "clusterName": "string",
  "clusterConfiguration": "string",
  "dryrun": boolean,
  "region": "string",
  "rollbackOnFailure": boolean,
  "suppressValidators": [ "string" ],
  "validationFailureLevel": "string"
}
```

## 요청 본문

### clusterConfiguration

YAML 문서로서의 클러스터 구성입니다.

유형: string

필수 항목 여부: 예

### clusterName

생성할 클러스터의 이름입니다.

이름은 영문자로 시작해야 합니다. 계획 이름은 최대 60자까지 가능합니다. Slurm 회계가 활성화된 경우 이름은 최대 40자까지 사용할 수 있습니다.



유형: string

필수 항목 여부: 예

### dryrun

true로 설정하면 리소스를 만들지 않고 요청 검증만 수행합니다. 이 파라미터를 사용하여 클러스터 구성을 확인합니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

### rollbackOnFailure

true로 설정하면 클러스터 생성에 실패할 경우 클러스터 스택 롤백이 발생합니다. 기본값은 true입니다.

유형: boolean

필수 항목 여부: 아니요

### suppressValidators

억제할 구성 유효성 검사기를 하나 이상 식별하세요.

유형: 문자열 목록

형식: (ALL | type: [A-Za-z0-9]+)

필수 여부: 아니요

### validationFailureLevel

클러스터 생성 실패를 유발하는 최소 검증 수준입니다. 기본값은 ERROR입니다.

유형: string

유효값: INFO | WARNING | ERROR

필수 항목 여부: 아니요

## 응답 구문

```
{
  "cluster": {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "clusterStatus": "CREATE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  },
  "validationMessages": [
    {
      "id": "string",
      "type": "string",
      "level": "INFO",
      "message": "string"
    }
  ]
}
```

## 응답 본문

### clusterName

클러스터의 이름입니다.

유형: string

### cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

## cloudformationStackStatus

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## clusterStatus

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## region

AWS 리전 클러스터가 생성되는 위치.

유형: string

## scheduler

### metadata

스케줄러의 메타데이터입니다.

### name

스케줄러의 이름입니다.

유형: string

### version

스케줄러의 버전입니다.

유형: string

### type

스케줄러의 유형입니다.

유형: string

## version

클러스터를 생성하는 데 사용된 AWS ParallelCluster 버전.

유형: string

## validation\_messages

검증 수준이 `validationFailureLevel`보다 낮은 메시지 목록입니다. 메시지 목록은 구성 검증 중에 수집됩니다.

### id

검사기의 ID입니다.

유형: string

### level

유형: string

유효값: INFO | WARNING | ERROR

### message

확인 메시지입니다.

유형: string

### type

검사기의 유형입니다.

유형: string

## 예제

### Python

#### 요청

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

#### 200 응답

```
{
```

```
'cluster': {
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster-3x/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
  'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
  'cluster_name': 'cluster-3x',
  'cluster_status': 'CREATE_IN_PROGRESS',
  'region': 'us-east-1',
  'scheduler': {
    'type': 'slurm'
  },
  'version': '3.2.1'
}
```

## deleteCluster

클러스터 삭제를 시작합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
DELETE /v3/clusters/{clusterName}
{
  "region": "string"
}
```

### 요청 본문

clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

region

AWS 리전 클러스터가 삭제되는 위치.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "cluster": {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS",
    "clusterStatus": "DELETE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  }
}
```

## 응답 본문

cluster

클러스터 인스턴스 목록입니다.

clusterName

클러스터의 이름입니다.

유형: string

cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

cloudformationStackStatus

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

clusterStatus

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

region

AWS 리전 클러스터가 생성되는 위치.

유형: string

scheduler

metadata

스케줄러의 메타데이터입니다.

name

스케줄러의 이름입니다.

유형: string

version

스케줄러의 버전입니다.

유형: string

## type

스케줄러의 유형입니다.

유형: string

## version

클러스터를 생성하는 데 사용된 AWS ParallelCluster 버전.

유형: string

## 예제

### Python

#### 요청

```
$ delete_cluster(cluster_name_3x)
```

#### 200 응답

```
{
  'cluster': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
    'cloudformation_stack_status': 'DELETE_IN_PROGRESS',
    'cluster_name': 'cluster_name_3x',
    'cluster_status': 'DELETE_IN_PROGRESS',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

## deleteClusterInstances

모든 클러스터 컴퓨팅 노드의 강제 종료를 시작합니다. 이 작업은 AWS Batch 클러스터를 지원하지 않습니다.

### 주제

- [요청 구문](#)



- [요청 본문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
DELETE /v3/clusters/{clusterName}/instances
{
  "force": boolean,
  "region": "string"
}
```

## 요청 본문

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### force

true로 설정하면 지정된 이름의 클러스터를 찾을 수 없을 때 강제로 삭제합니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

### region

AWS 리전 클러스터가 속해 있는 대상.

유형: string

필수사항: 아니요

## 응답 본문

None

## 예제

### Python

#### 요청

```
$ delete_cluster_instances(cluster_name_3x)
```

#### 200 응답

None

## deleteImage

사용자 지정 AWS ParallelCluster 이미지 삭제를 시작합니다.

### 주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
DELETE /v3/images/custom/{imageId}
{
  "force": boolean,
  "region": "string"
}
```

### 요청 본문

#### imageId

이미지의 ID입니다.

유형: string

필수 항목 여부: 예

force

true로 설정된 경우 AMI를 강제로 삭제합니다. AMI를 사용하는 인스턴스가 있거나 AMI를 공유하는 경우 이 파라미터를 사용하세요. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

region

AWS 리전 이미지가 생성된 위치.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "image": {
    "imageId": "string",
    "ec2AmiInfo": {
      "amiId": "string"
    },
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "imageBuildStatus": "DELETE_IN_PROGRESS",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS"
  }
}
```

## 응답 본문

image

cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

## cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmilInfo

amild

아마존 EC2 AMI ID.

유형: string

## imageBuildStatus

이미지 빌드 상태입니다.

유형: string

유효값: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

## imageId

이미지의 ID입니다.

유형: string

## region

이미지가 AWS 리전 생성되는 위치입니다.

유형: string

## version

이미지를 만드는 데 사용된 AWS ParallelCluster 버전.

유형: string

## 예제

### Python

#### 요청

```
$ delete_image(custom-image-id)
```

#### 200 응답

```
{
  'image': {
    'image_build_status': 'DELETE_IN_PROGRESS',
    'image_id': 'custom-image-id',
    'region': 'us-east-1',
    'version': '3.2.1'
  }
}
```

## describeCluster

기존 클러스터에 대한 세부 정보를 가져옵니다.

### 주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
GET /v3/clusters/{clusterName}
{
```

```
"region": "string"
}
```

## 요청 본문

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

## 응답 구문

### Note

failureReason AWS ParallelCluster 버전 failures 3.5.0부터 시작하도록 변경되었습니다.

```
{
  "clusterName": "string",
  "region": "string",
  "version": "string",
  "cloudFormationStackStatus": "CREATE_IN_PROGRESS",
  "clusterStatus": "CREATE_IN_PROGRESS",
  "scheduler": {
    "type": "string",
    "metadata": {
      "name": "string",
      "version": "string"
    }
  }
}
```

```
  },
  "cloudformationStackArn": "string",
  "creationTime": "2019-08-24T14:15:22Z",
  "lastUpdatedTime": "2019-08-24T14:15:22Z",
  "clusterConfiguration": {
    "url": "string"
  },
  "computeFleetStatus": "START_REQUESTED",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "headNode": {
    "instanceId": "string",
    "instanceType": "string",
    "launchTime": "2019-08-24T14:15:22Z",
    "privateIpAddress": "string",
    "publicIpAddress": "string",
    "state": "pending"
  },
  "failures": [
    {
      "failureCode": "string",
      "failureReason": "string"
    }
  ]
  "loginNodes": {
    "status": "string",
    "address": "string",
    "scheme": "string",
    "healthyNodes": integer,
    "unhealthyNodes": integer
  }
}
```

## 응답 본문

### clusterName

클러스터의 이름입니다.

유형: string

## cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

## cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## clusterConfiguration

url

클러스터 구성 파일의 URL입니다.

유형: string

## clusterStatus

클러스터 상태입니다.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## computeFleetStatus

컴퓨팅 플릿 상태입니다.

유형: string

유효값: START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED |  
STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED



## creationTime

클러스터가 생성된 시간의 타임스탬프입니다.

유형: 날짜/시간

## lastUpdatedTime

클러스터가 마지막으로 업데이트된 시간의 타임스탬프입니다.

유형: 날짜/시간

## region

AWS 리전 클러스터가 생성되는 위치입니다.

유형: string

## tags

클러스터와 연관된 태그의 목록입니다.

키

태그 이름입니다.

유형: string

## tag

태그 값입니다.

유형: string

## version

클러스터를 생성하는 데 사용된 AWS ParallelCluster 버전.

유형: string

## failures

클러스터 스택이 CREATE\_FAILED 상태에 있을 때의 실패 목록입니다.

## failureCode

클러스터 스택이 CREATE\_FAILED 상태에 있을 때의 장애 코드입니다.

유형: string

failureReason

클러스터 스택이 CREATE\_FAILED 상태일 때 오류가 발생하는 이유입니다.

유형: string

head\_node

클러스터 헤드 노드입니다.

instanceId

아마존 EC2 인스턴스 ID.

유형: string

instanceType

Amazon EC2 인스턴스 유형

유형: string

launchTime

Amazon EC2 인스턴스가 시작된 시간입니다.

유형: 날짜/시간

privateIpAddress

클러스터 프라이빗 IP 주소입니다.

유형: string

publicIpAddress

클러스터 퍼블릭 IP 주소입니다.

유형: string

state

헤드 노드 인스턴스 상태입니다.

유형: string

유효값: pending | running | shutting-down | terminated | stopping | stopped

## scheduler

### metadata

스케줄러의 메타데이터입니다.

#### name

스케줄러의 이름입니다.

유형: string

#### version

스케줄러의 버전입니다.

유형: string

## loginNodes

### status

로그인 노드 상태입니다.

유형: string

유효값: PENDING | FAILED | ACTIVE

### address

로그인 노드 주소입니다.

유형: string

### scheme

로그인 노드 스키마입니다.

유형: string

### scheme

정상 노드의 수입니다.

유형: 정수

## scheme

비정상 노드의 수입니다.

유형: 정수

## type

스케줄러의 유형입니다.

유형: string

## 예제

### Python

#### 요청

```
$ describe_cluster(cluster_name_3x)
```

#### 200 응답

```
{
  'cloud_formation_stack_status': 'CREATE_COMPLETE',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
  'cluster_configuration': {
    'url': 'https://parallelcluster-....'
  },
  'cluster_name': 'cluster_name_3x',
  'cluster_status': 'CREATE_COMPLETE',
  'compute_fleet_status': 'RUNNING',
  'creation_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
  'head_node': {
    'instance_id': 'i-abcdef01234567890',
    'instance_type': 't2.micro',
    'launch_time': datetime.datetime(2022, 3, 28, 22, 21, 56, tzinfo=tzlocal()),
    'private_ip_address': '172.31.56.3',
    'public_ip_address': '107.23.100.164',
    'state': 'running'
  },
}
```

```
'last_updated_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [
  {
    'key': 'parallelcluster:version', 'value': '3.2.1'
  }
],
'version': '3.2.1'
}
```

## describeClusterInstances

클러스터에 속하는 인스턴스를 설명합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
GET /v3/clusters/{clusterName}/instances
{
  "nextToken": "string",
  "nodeType": "string",
  "queueName": "string",
  "region": "string"
}
```

### 요청 본문

clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

#### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

#### nodeType

노드 유형별로 인스턴스를 필터링합니다.

유형: string

유효한 값: HeadNode, ComputeNode, LoginNode

필수 여부: 아니요

#### queueName

대기열 이름을 기준으로 인스턴스를 필터링합니다.

유형: string

필수사항: 아니요

#### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "nextToken": "string",
  "instances": [
    {
      "instanceId": "string",
      "instanceType": "string",
```

```
    "launchTime": "2019-08-24T14:15:22Z",
    "privateIpAddress": "string",
    "publicIpAddress": "string",
    "state": "pending",
    "nodeType": "HeadNode",
    "queueName": "string"
  }
]
}
```

## 응답 본문

### instances

클러스터 인스턴스 목록입니다.

#### instanceId

아마존 EC2 인스턴스 ID.

유형: string

#### instanceType

Amazon EC2 인스턴스 유형

유형: string

#### launchTime

Amazon EC2 인스턴스가 시작된 시간입니다.

유형: 날짜/시간

#### nodeType

노드 유형입니다.

유형: string

유효한 값: HeadNode, ComputeNode, LoginNode

#### publicIpAddress

클러스터 퍼블릭 IP 주소입니다.

유형: string

queueName

Amazon EC2 인스턴스가 노드를 지원하는 대기열의 이름입니다.

유형: string

state

노드 Amazon EC2 인스턴스 상태.

유형: string

유효값: pending | running | shutting-down | terminated | stopping | stopped

nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

요청

```
$ describe_cluster_instances(cluster_name_3x)
```

200 응답

```
{
  'instances': [
    {
      'instance_id': 'i-abcdef01234567890',
      'instance_type': 't2.micro',
      'launch_time': datetime.datetime(2022, 3, 30, 14, 2, 7, tzinfo=tzlocal()),
      'node_type': 'HeadNode',
      'private_ip_address': '192.0.2.5',
      'public_ip_address': '198.51.100.180',
      'state': 'running'
    }
  ]
}
```



```
    }  
  ]  
}
```

## describeComputeFleet

컴퓨팅 플릿의 상태를 설명합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
GET /v3/clusters/{clusterName}/computefleet  
{  
  "region": "string"  
}
```

### 요청 본문

clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "status": "START_REQUESTED",
  "lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

## 응답 본문

### status

유형: string

유효값: START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED  
| STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdateTime

마지막 상태 업데이트 시간을 나타내는 타임스탬프입니다.

유형: 날짜/시간

## 예제

### Python

요청

```
$ describe_compute_fleet(cluster_name_3x)
```

200 응답

```
{
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
  tzinfo=tzlocal()),
  'status': 'RUNNING'
}
```

## describeImage

기존 이미지에 대한 세부 정보를 가져옵니다.

## 주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
GET /v3/images/custom/{imageId}
{
  "region": "string"
}
```

## 요청 본문

### imageId

이미지의 ID입니다.

유형: string

필수 항목 여부: 예

### region

이미지를 AWS 리전 만든 곳.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "imageId": "string",
  "region": "string",
  "version": "string",
```

```
"imageBuildStatus": "BUILD_IN_PROGRESS",
"imageBuildLogsArn": "string",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackStatusReason": "string",
"cloudformationStackArn": "string",
"creationTime": "2019-08-24T14:15:22Z",
"cloudformationStackCreationTime": "2019-08-24T14:15:22Z",
"cloudformationStackTags": [
  {
    "key": "string",
    "value": "string"
  }
],
"imageConfiguration": {
  "url": "string"
},
"imagebuilderImageStatus": "PENDING",
"imagebuilderImageStatusReason": "string",
"ec2AmiInfo": {
  "amiId": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "amiName": "string",
  "architecture": "string",
  "state": "PENDING",
  "description": "string"
}
}
```

## 응답 본문

### imageId

세부 정보를 검색할 이미지의 ID입니다.

유형: string

### imageBuildStatus

이미지 빌드 상태입니다.

유형: string

유효값: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

imageConfiguration

url

이미 구성 파일의 URL입니다.

유형: string

region

이미지가 AWS 리전 생성되는 위치.

유형: string

version

이미지를 만드는 데 사용된 AWS ParallelCluster 버전.

유형: string

cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

cloudformationStackCreation시간

CloudFormation 스택이 생성된 시점의 타임스탬프입니다.

유형: 날짜/시간

cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |

UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

cloudformationStackStatus이유

CloudFormation 스택 상태의 이유.

유형: string

cloudformationStackTags

CloudFormation 스택의 태그 목록.

키

태그 이름입니다.

유형: string

값

태그 값입니다.

유형: string

creationTime

이미지가 생성된 시간의 타임스탬프입니다.

유형: 날짜/시간

ec2 AmiInfo

amild

아마존 EC2 AMI ID.

유형: string

amiName

아마존 EC2 AMI 이름.

유형: string

architecture

아마존 EC2 AMI 아키텍처.

유형: string

state

아마존 EC2 AMI의 상태.

유형: string

유효값: PENDING | AVAILABLE | INVALID | DEREGISTERED | TRANSIENT | FAILED | ERROR

tags

아마존 EC2 AMI 태그 목록.

키

태그 이름입니다.

유형: string

값

태그 값입니다.

유형: string

imagebuilderImageStatus

ImageBuilder 상태.

유형: string

유효값: PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING | INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

imagebuilderImageStatus이유

ImageBuilder 이미지 상태의 이유.

유형: string

imageBuildLogsArn

이미지 빌드 프로세스에 대한 로그의 Amazon 리소스 이름(ARN)입니다.

유형: string

## 예제

### Python

#### 요청

```
$ describe_image(custom-image-id)
```

#### 200 응답

```
{
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-image-id/6accc570-b080-11ec-845e-0e2dc6386985',
  'cloudformation_stack_creation_time': datetime.datetime(2022, 3, 30, 23, 23, 33,
731000, tzinfo=tzlocal()),
  'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
  'cloudformation_stack_tags': [
    {
      'key': 'parallelcluster:version', 'value': '3.2.1'
    },
    {
      'key': 'parallelcluster:image_name',
      'value': 'custom-image-id'
    },
    {
      'key': 'parallelcluster:custom-image-id',
      'value': 'custom-image-id'
    },
    {
      'key': 'parallelcluster:s3_bucket',
      'value': 'parallelcluster-abcdef01234567890-v1-do-not-delete'
    },
    {
      'key': 'parallelcluster:s3_image_dir',
      'value': 'parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0'
    },
    {
      'key': 'parallelcluster:build_log',
      'value': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-image-id'
    },
    {
      'key': 'parallelcluster:build_config',
```



```

    'value': 's3://parallelcluster-abcdef01234567890-v1-do-not-delete/
parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-
config.yaml'
  }
],
  'image_build_logs_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image',
  'image_build_status': 'BUILD_IN_PROGRESS',
  'image_configuration': {
    'url': 'https://parallelcluster-abcdef01234567890-v1-do-not-
delete.s3.amazonaws.com/parallelcluster/3.2.1/images/custom-image-
id-1234567890abcdef0/configs/image-config.yaml?... '
  },
  'image_id': 'custom-image-id',
  'imagebuilder_image_status': 'PENDING',
  'region': 'us-east-1',
  'version': '3.2.1'
}

```

## getClusterLog이벤트

로그 스트림과 연결된 이벤트를 검색합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```

GET /v3/clusters/{clusterName}/logstreams/{logStreamName}
{
  "endTime": datetime,
  "limit": float,
  "nextToken": "string",
  "region": "string",
  "startFromHead": boolean,

```

```
"startTime": datetime
}
```

## 요청 본문

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### logStreamName

로그 스트림의 이름입니다.

유형: string

필수 항목 여부: 예

### endTime

시간 범위의 끝입니다. ISO 8601 형식으로 표시됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다.

유형: 날짜/시간

형식: 2021-01-01T20:00:00Z

필수 여부: 아니요

### 제한

반환된 로그 이벤트의 최대 수입니다. 값을 지정하지 않는 경우 최대값은 1MB의 응답 크기에 들어갈 수 있는 로그 이벤트 수(최대 10,000개의 로그 이벤트)입니다.

유형: float

필수 여부: 아니요

### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

### startFromHead

true로 설정하면 가장 이른 로그 이벤트가 먼저 반환됩니다. 값이 false인 경우 최신 로그 이벤트가 먼저 반환됩니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

### startTime

시간 범위의 시작입니다. ISO 8601 형식으로 표시됩니다. 타임스탬프가 이 시간 이후인 이벤트가 포함됩니다.

유형: 날짜/시간

형식: 2021-01-01T20:00:00Z

필수 항목 여부: 아니요

## 응답 구문

```
{
  "nextToken": "string",
  "prevToken": "string",
  "events": [
    {
      "timestamp": "2019-08-24T14:15:22Z",
      "message": "string"
    }
  ]
}
```

## 응답 본문

### 이벤트

필터링된 이벤트 목록입니다.

message

이벤트 메시지

유형: string

타임스탬프

이벤트 타임스탬프입니다.

유형: 날짜/시간

nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

prevToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

요청

```
$ get_cluster_log_events(cluster_name_3x, log_stream_name=ip-192-0-2-26.i-  
abcdef01234567890.cfn-init)
```

200 응답

```
"events": [  
  {
```

```

    "message": "2022-09-22 16:40:15,127 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.us-east-1.amazonaws.com",
    "timestamp": "2022-09-22T16:40:15.127Z"
  },
  {
    "message": "2022-09-22 16:40:15,127 [DEBUG] Describing resource
HeadNodeLaunchTemplate in stack cluster_name_3x",
    "timestamp": "2022-09-22T16:40:15.127Z"
  },
  ...
]

```

## getClusterStack이벤트

클러스터의 스택과 관련된 이벤트를 검색합니다.

### Note

버전 3.6.0부터 중첩된 스택을 AWS ParallelCluster 사용하여 큐 및 컴퓨팅 리소스와 관련된 리소스를 생성합니다. GetClusterStackEvents API와 `pcluster get-cluster-stack-events` 명령은 클러스터 메인 스택 이벤트만 반환합니다. 콘솔에서 대기열 및 컴퓨팅 리소스와 관련된 이벤트를 비롯한 클러스터 스택 이벤트를 볼 수 있습니다. CloudFormation

### 주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```

GET /v3/clusters/{clusterName}/stackevents
{
  "nextToken": "string",
  "region": "string"
}

```

```
}
```

## 요청 본문

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "nextToken": "string",
  "events": [
    {
      "stackId": "string",
      "eventId": "string",
      "stackName": "string",
      "logicalResourceId": "string",
      "physicalResourceId": "string",
      "resourceType": "string",
      "timestamp": "2019-08-24T14:15:22Z",
      "resourceStatus": "CREATE_IN_PROGRESS",
      "resourceStatusReason": "string",
```

```
    "resourceProperties": "string",  
    "clientRequestToken": "string"  
  }  
]  
}
```

## 응답 본문

### 이벤트

필터링된 이벤트 목록입니다.

#### clientRequestToken

이 이벤트를 생성한 액션에 전달된 토큰입니다.

유형: string

#### eventId

이 이벤트의 고유 ID입니다.

유형: string

#### logicalResourceId

템플릿에 지정된 리소스의 논리명입니다.

유형: string

#### physicalResourceId

리소스의 물리적 인스턴스와 연결된 이름 또는 고유 식별자입니다.

유형: string

#### resourceProperties

리소스를 만드는 데 사용되는 속성의 BLOB입니다.

유형: string

#### resourceStatus

리소스 상태입니다.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
 DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED  
 | UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED  
 | IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS |  
 IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

resourceStatusReason

리소스와 관련된 성공 또는 실패 메시지입니다.

유형: string

resourceType

리소스의 유형입니다.

유형: string

stackId

스택 인스턴스의 고유 ID 이름입니다.

유형: string

stackName

스택과 연결되어 있는 이름입니다.

유형: string

타임스탬프

상태가 업데이트된 시간입니다.

유형: 날짜/시간

nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

요청



```
$ get_cluster_stack_events(cluster_name_3x)
```

200 응답

```
{
  'events': [
    {
      'event_id': '590b3820-b081-11ec-985e-0a7af5751497',
      'logical_resource_id': 'cluster_name_3x',
      'physical_resource_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
      'resource_status': 'CREATE_COMPLETE',
      'resource_type': 'AWS::CloudFormation::Stack',
      'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
      'stack_name': 'cluster_name_3x',
      'timestamp': datetime.datetime(2022, 3, 30, 23, 30, 13, 268000,
tzinfo=tzlocal())
    },
    ...
  ]
}
```

## getImageLog이벤트

이미지 빌드와 관련된 이벤트를 검색합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
GET /v3/images/custom/{imageId}/logstreams/{logStreamName}
{
```

```

"endTime": datetime,
"limit": float,
"nextToken": "string",
"region": "string",
"startFromHead": boolean,
"startTime": datetime
}

```

## 요청 본문

### imageId

이미지의 ID입니다.

유형: string

필수 항목 여부: 예

### logStreamName

로그스트림의 이름입니다.

유형: string

필수 항목 여부: 예

### endTime

시간 범위의 끝입니다. ISO 8601 형식으로 표시됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다.

유형: 날짜/시간

형식: 2021-01-01T20:00:00Z

필수 여부: 아니요

### 제한

반환된 로그 이벤트의 최대 수입니다. 값을 지정하지 않는 경우 최대값은 1MB의 응답 크기에 들어갈 수 있는 로그 이벤트 수(최대 10,000개의 로그 이벤트)입니다.

유형: float

필수 여부: 아니요

## nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

## region

이미지가 들어 AWS 리전 있는 것이죠.

유형: string

필수사항: 아니요

## startFromHead

true로 설정된 경우 가장 이른 로그 이벤트를 먼저 반환합니다. false로 설정되어 있는 경우에는 최신 로그 이벤트를 먼저 반환합니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

## startTime

시간 범위의 시작입니다. ISO 8601 형식으로 표시됩니다. 타임스탬프가 이 시간 이후인 이벤트가 포함됩니다.

유형: 날짜/시간

형식: 2021-01-01T20:00:00Z

필수 항목 여부: 아니요

## 응답 구문

```
{
  "nextToken": "string",
  "prevToken": "string",
  "events": [
    {
      "timestamp": "2019-08-24T14:15:22Z",
```

```

    "message": "string"
  }
]
}

```

## 응답 본문

### 이벤트

필터링된 이벤트 목록입니다.

#### message

이벤트 메시지

유형: string

#### 타임스탬프

이벤트 타임스탬프입니다.

유형: 날짜/시간

#### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

#### prevToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

요청

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

200 응답

```

"events": [
  {
    "message": "ExecuteBash: STARTED EXECUTION",
    "timestamp": "2022-04-05T15:51:20.228Z"
  },
  {
    "message": "ExecuteBash: Created temporary directory: /tmp/1234567890abcdef0",
    "timestamp": "2022-04-05T15:51:20.228Z"
  },
  ...
]

```

## getImageStack이벤트

이미지 빌드의 스택과 관련된 이벤트를 검색합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```

GET /v3/images/custom/{imageId}/stackevents
{
  "nextToken": "string",
  "region": "string"
}

```

### 요청 본문

imageId

이미지의 ID입니다.

유형: string

필수 항목 여부: 예

#### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

#### region

이미지가 들어 AWS 리전 있는 것.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "nextToken": "string",
  "events": [
    {
      "stackId": "string",
      "eventId": "string",
      "stackName": "string",
      "logicalResourceId": "string",
      "physicalResourceId": "string",
      "resourceType": "string",
      "timestamp": "2019-08-24T14:15:22Z",
      "resourceStatus": "CREATE_IN_PROGRESS",
      "resourceStatusReason": "string",
      "resourceProperties": "string",
      "clientRequestToken": "string"
    }
  ]
}
```

## 응답 본문

### 이벤트

필터링된 이벤트 목록입니다.

### clientRequestToken

이 이벤트를 생성한 액션에 전달된 토큰입니다.

유형: string

### eventId

이 이벤트의 고유 ID입니다.

유형: string

### logicalResourceId

템플릿에 지정된 리소스의 논리명입니다.

유형: string

### physicalResourceId

리소스의 물리적 인스턴스와 연결된 이름 또는 고유 식별자입니다.

유형: string

### resourceProperties

리소스를 만드는 데 사용되는 속성의 BLOB입니다.

유형: string

### resourceStatus

리소스 상태입니다.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED  
| UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED  
| IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS |  
IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

### resourceStatusReason

리소스와 관련된 성공 또는 실패 메시지입니다.

유형: string

## resourceType

리소스의 유형입니다.

유형: string

## stackId

스택 인스턴스의 고유 ID 이름입니다.

유형: string

## stackName

스택과 연결되어 있는 이름입니다.

유형: string

## 타임스탬프

상태가 업데이트된 시간입니다.

유형: 날짜/시간

## nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

#### 요청

```
$ get_image_stack_events(image_id)
```

#### 200 응답

```
{
  'events': [
    {
      'event_id': 'ParallelClusterImage-
CREATE_IN_PROGRESS-2022-03-30T23:26:33.499Z',
      'logical_resource_id': 'ParallelClusterImage',
```



```

    'physical_resource_id': 'arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-alinux2-image/3.2.1/1',
    'resource_properties': {
        "InfrastructureConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
        "ImageRecipeArn": "arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-alinux2-image/3.2.1",
        "DistributionConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
        "EnhancedImageMetadataEnabled": "false",
        "Tags": {
            "parallelcluster:image_name": "alinux2-
image", "parallelcluster:image_id": "alinux2-image"
        }
    },
    'resource_status': 'CREATE_IN_PROGRESS',
    'resource_status_reason': 'Resource creation Initiated',
    'resource_type': 'AWS::ImageBuilder::Image',
    'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/alinux2-
image/6accc570-b080-11ec-845e-0e2dc6386985',
    'stack_name': 'alinux2-image',
    'timestamp': datetime.datetime(2022, 3, 30, 23, 26, 33, 499000,
tzinfo=tzlocal())
},
...
]
}

```

## listClusters

기존 클러스터의 목록을 검색합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
GET /v3/clusters
{
  "clusterStatus": "string",
  "nextToken": "string",
  "region": "string"
}
```

## 요청 본문

### clusterStatus

클러스터 상태별로 필터링합니다. 기본값은 모든 클러스터입니다.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_FAILED

필수 항목 여부: 아니요

### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

### region

AWS 리전 클러스터의.

유형: string

필수사항: 아니요

## 응답 구문

```
{
```

```
"nextToken": "string",
"clusters": [
  {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "clusterStatus": "CREATE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  }
]
}
```

## 응답 본문

### 클러스터

#### cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

#### cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

**clusterName**

클러스터의 이름입니다.

유형: string

**clusterStatus**

클러스터 상태입니다.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

**scheduler****metadata**

스케줄러의 메타데이터입니다.

**name**

스케줄러의 이름입니다.

유형: string

**version**

스케줄러의 버전입니다.

유형: string

**type**

스케줄러의 유형입니다.

유형: string

**region**

AWS 리전 클러스터가 생성되는 위치입니다.

유형: string

**version**

클러스터를 생성하는 데 사용된 AWS ParallelCluster 버전.

유형: string

nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

Python

요청

```
$ list_clusters()
```

200 응답

```
{
  'clusters':
  [
    {
      'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
      'cloudformation_stack_status': 'CREATE_COMPLETE',
      'cluster_name': 'cluster_name_3x',
      'cluster_status': 'CREATE_COMPLETE',
      'region': 'us-east-1',
      'version': '3.2.1'
    },
    ...
  ]
}
```

## listClusterLog스트림

클러스터와 연결된 로그 스트림의 목록을 검색합니다.

주제

- [요청 구문](#)

- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
GET /v3/clusters/{clusterName}/logstreams
{
  "filters": [ "string" ],
  "nextToken": "string",
  "region": "string"
}
```

## 요청 본문

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### 필터

로그 스트림을 필터링합니다.

허용되는 필터는 다음과 같습니다.

- private-dns-name: 인스턴스의 프라이빗 DNS 이름의 약식 (예: ip-10-0-0-101).
- node-type: 유효한 값: HeadNode

유형: 고유 문자열 배열

형식: Name=a, Values=1 Name=b, Values=2, 3

필수 여부: 아니요

### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

## region

클러스터가 속해 AWS 리전 있는 곳.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "nextToken": "string",
  "logStreams": [
    {
      "logStreamName": "string",
      "creationTime": "2019-08-24T14:15:22Z",
      "firstEventTimestamp": "2019-08-24T14:15:22Z",
      "lastEventTimestamp": "2019-08-24T14:15:22Z",
      "lastIngestionTime": "2019-08-24T14:15:22Z",
      "uploadSequenceToken": "string",
      "logStreamArn": "string"
    }
  ]
}
```

## 응답 본문

### logStreams

로그 스트림 목록입니다.

#### creationTime

스트림이 생성된 시간입니다.

유형: 날짜/시간

#### firstEventTimestamp

스트림 첫 번째 이벤트의 시간입니다.

유형: 날짜/시간

#### lastEventTimestamp

스트림 마지막 이벤트의 시간입니다. lastEventTime 값은 최종 일관성을 기준으로 업데이트됩니다. 일반적으로 수집 후 한 시간 이내에 업데이트되지만 드물게 그보다 더 오래 걸릴 수 있습니다.

유형: 날짜/시간

#### lastIngestionTime

마지막 수집 시간입니다.

유형: 날짜/시간

#### logStreamArn

로그 스트림의 Amazon 리소스 이름(ARN)입니다.

유형: string

#### logStreamName

로그 스트림의 이름입니다.

유형: string

#### uploadSequenceToken

시퀀스 토큰입니다.

유형: string

#### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

#### 요청

```
$ list_cluster_log_streams(cluster_name_3x)
```



## 200 응답

```
{
  'log_streams': [
    {
      'creation_time': datetime.datetime(2022, 3, 30, 14, 7, 34, 354000,
tzinfo=tzlocal()),
      'first_event_timestamp': datetime.datetime(2022, 3, 30, 14, 6, 41, 444000,
tzinfo=tzlocal()),
      'last_event_timestamp': datetime.datetime(2022, 3, 30, 14, 25, 55, 462000,
tzinfo=tzlocal()),
      'last_ingestion_time': datetime.datetime(2022, 3, 30, 14, 49, 50, 62000,
tzinfo=tzlocal()),
      'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
parallelcluster/cluster_name_3x:log-stream:ip-192-0-2-26.i-abcdef01234567890.cfn-
init',
      'log_stream_name': 'ip-192-0-2-26.i-abcdef01234567890.cfn-init',
      ...
      'upload_sequence_token': '####'
    },
    ...
  ]
}
```

## listImageLog스트림

이미지와 관련된 로그 스트림 목록을 검색합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
GET /v3/images/custom/{imageId}/logstreams
{
```

```
"nextToken": "string",  
"region": "string"  
}
```

## 요청 본문

### imageId

이미지의 ID입니다.

유형: string

필수 항목 여부: 예

### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

### region

이미지가 들어 AWS 리전 있는 그대로요.

유형: string

필수사항: 아니요

## 응답 구문

```
{  
  "nextToken": "string",  
  "logStreams": [  
    {  
      "logStreamName": "string",  
      "creationTime": "2019-08-24T14:15:22Z",  
      "firstEventTimestamp": "2019-08-24T14:15:22Z",  
      "lastEventTimestamp": "2019-08-24T14:15:22Z",  
      "lastIngestionTime": "2019-08-24T14:15:22Z",  
      "uploadSequenceToken": "string",  
      "logStreamArn": "string"  
    }  
  ]  
}
```

```
}  
]  
}
```

## 응답 본문

### logStreams

로그 스트림 목록입니다.

#### creationTime

스트림이 생성된 시간입니다.

유형: 날짜/시간

#### firstEventTimestamp

스트림 내 첫 번째 이벤트의 시간입니다.

유형: 날짜/시간

#### lastEventTimestamp

스트림 마지막 이벤트의 시간입니다. lastEventTime 값은 최종 일관성을 기준으로 업데이트됩니다. 일반적으로 수집 후 한 시간 이내에 업데이트되지만 드물게 그보다 더 오래 걸릴 수 있습니다.

유형: 날짜/시간

#### lastIngestionTime

마지막 수집 시간입니다.

유형: 날짜/시간

#### logStreamArn

로그 스트림의 Amazon 리소스 이름(ARN)입니다.

유형: string

#### logStreamName

로그 스트림의 이름입니다.

유형: string

uploadSequenceToken

시퀀스 토큰입니다.

유형: string

next\_token

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

요청

```
$ list_image_log_streams(custom-image-id)
```

200 응답

```
{
  'log_streams': [
    {
      'creation_time': datetime.datetime(2022, 3, 29, 20, 29, 24, 875000,
tzinfo=tzlocal()),
      'first_event_timestamp': datetime.datetime(2022, 3, 29, 20, 29, 24, 775000,
tzinfo=tzlocal()),
      'last_event_timestamp': datetime.datetime(2022, 3, 29, 20, 38, 23, 944000,
tzinfo=tzlocal()),
      'last_ingestion_time': datetime.datetime(2022, 3, 29, 20, 51, 56, 26000,
tzinfo=tzlocal()),
      'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image:log-stream:3.2.1/1',
      'log_stream_name': '3.2.1/1',
      'upload_sequence_token': '####'
    },
    ...
  ]
}
```

# listImages

기존 사용자 지정 이미지 목록을 검색합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
GET /images/custom
{
  "imageStatus": "string",
  "nextToken": "string",
  "region": "string"
}
```

## 요청 본문

imageStatus

제공된 상태별로 이미지를 필터링합니다.

유형: string

유효값: AVAILABLE | PENDING | FAILED

필수 항목 여부: 예

nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

필수사항: 아니요

## region

AWS 리전 해당 이미지가 들어 있습니다.

유형: string

필수사항: 아니요

## 응답 구분

```
{
  "nextToken": "string",
  "images": [
    {
      "imageId": "string",
      "ec2AmiInfo": {
        "amiId": "string"
      },
      "region": "string",
      "version": "string",
      "cloudformationStackArn": "string",
      "imageBuildStatus": "BUILD_IN_PROGRESS",
      "cloudformationStackStatus": "CREATE_IN_PROGRESS"
    }
  ]
}
```

## 응답 본문

### images

이미지의 목록입니다.

#### cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

#### cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

#### ec2 AmiInfo

ami\_id

아마존 EC2 AMI ID.

유형: string

#### imageBuildStatus

이미지 빌드 상태입니다.

유효값: BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

유형: string

#### imageId

이미지의 ID입니다.

유형: string

#### region

이미지가 AWS 리전 생성되는 위치입니다.

유형: string

#### version

이미지를 만드는 데 사용된 AWS ParallelCluster 버전.

유형: string

#### nextToken

페이지가 매겨진 요청에 사용되는 토큰입니다.

유형: string

## 예제

### Python

#### 요청

```
$ list_images("AVAILABLE")
```

#### 200 응답

```
{
  'images': [
    {
      'ec2_ami_info': {
        'ami_id': 'ami-abcdef01234567890'
      },
      'image_build_status': 'BUILD_COMPLETE',
      'image_id': 'custom-image',
      'region': 'us-east-1',
      'version': '3.2.1'
    }
  ]
}
```

## listOfficialImages

AWS ParallelCluster 공식 이미지 목록을 검색하세요.

#### 주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

#### 요청 구문

```
GET /v3/images/official
```



```
{
  "architecture": "string",
  "os": "string",
  "region": "string"
}
```

## 요청 본문

### architecture

아키텍처별로 필터링합니다. 기본값은 필터링 없음입니다.

유형: string

유효값: x86\_64 | arm64

필수 여부: 아니요

### os

OS 배포를 기준으로 필터링합니다. 기본값은 필터링 없음입니다.

유형: string

유효값: alinux2 | centos7 | ubuntu2204 | ubuntu2004 | rhel8

필수 여부: 아니요

### region

AWS 리전 해당 공식 이미지는 에 나열되어 있습니다.

유형: string

필수사항: 아니요

## 응답 구문

```
{
  "images": [
    {
      "architecture": "string",
      "amiId": "string",
      "name": "string",

```

```
    "os": "string",  
    "version": "string"  
  }  
]  
}
```

## 응답 본문

### images

#### amild

AMI의 ID입니다.

유형: string

#### architecture

AMI 아키텍처입니다.

유형: string

#### name

AMI의 이름입니다.

유형: string

#### os

AMI 운영 체제입니다.

유형: string

#### version

AWS ParallelCluster 버전.

유형: string

## 예제

### Python

#### 요청

```
$ list_official_images()
```

200 응답

```
{
  'images': [
    {
      'ami_id': 'ami-015cfefb4e0d6306b2',
      'architecture': 'x86_64',
      'name': 'aws-parallelcluster-3.2.1-ubuntu-2004-lts-hvm-x86_64-202202261505 '
      '2022-02-26T15-08-34.759Z',
      'os': 'ubuntu2004',
      'version': '3.2.1'
    },
    ...
  ]
}
```

## updateCluster

클러스터를 업데이트합니다.

주제

- [요청 구문](#)
- [요청 본문](#)
- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

### 요청 구문

```
PUT /v3/clusters/{clusterName}
{
  "clusterConfiguration": "string",
  "dryrun": boolean,
  "forceUpdate": boolean,
  "region": "string",
  "suppressValidators": "string",
```

```
"validationFailureLevel": "string"  
}
```

## 요청 본문

### clusterConfiguration

YAML 문서로서의 클러스터 구성입니다.

필수 여부: 예

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### dryrun

true로 설정하면 리소스를 만들지 않고 요청 검증만 수행합니다. 이 파라미터를 사용하여 클러스터 구성과 업그레이드 요구 사항을 확인합니다. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

### forceUpdate

true로 설정된 경우 업데이트 검증 오류를 무시하고 강제로 업데이트하세요. 기본값은 false입니다.

유형: boolean

필수 항목 여부: 아니요

### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

## suppressValidators

억제할 구성 유효성 검사기를 하나 이상 식별하세요.

유형: string

형식: (ALL|type:[A-Za-z0-9]+)

필수 여부: 아니요

유효한 값의 예제: currentValue, requestedValue, message

## validationFailureLevel

업데이트 실패의 원인이 되는 최소 검증 수준입니다.

유형: string

유효값: INFO | WARNING | ERROR

필수 항목 여부: 아니요

## 응답 구문

```
{
  "cluster": {
    "clusterName": "string",
    "region": "string",
    "version": "string",
    "cloudformationStackArn": "string",
    "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
    "clusterStatus": "UPDATE_IN_PROGRESS",
    "scheduler": {
      "type": "string",
      "metadata": {
        "name": "string",
        "version": "string"
      }
    }
  },
  "validationMessages": [
    {
      "id": "string",
```

```
    "type": "string",
    "level": "INFO",
    "message": "string"
  }
],
"changeSet": [
  {
    "parameter": "string",
    "currentValue": "string",
    "requestedValue": "string"
  }
]
}
```

## 응답 본문

### changeSet

클러스터 업데이트의 변경 세트입니다.

#### currentValue

업데이트할 파라미터의 현재 값입니다.

유형: string

#### parameter

업데이트할 파라미터입니다.

유형: string

#### requestedValue

업데이트할 파라미터에 요청된 값입니다.

유형: string

### cluster

#### cloudformationStackArn

메인 CloudFormation 스택의 Amazon 리소스 이름 (ARN).

유형: string

## cloudformationStackStatus

CloudFormation 스택 상태.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## clusterName

클러스터의 이름입니다.

유형: string

## clusterStatus

클러스터 상태입니다.

유형: string

유효값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## region

AWS 리전 클러스터가 생성되는 위치입니다.

유형: string

## scheduler

### metadata

스케줄러의 메타데이터입니다.

### name

스케줄러의 이름입니다.

유형: string

**version**

스케줄러의 버전입니다.

유형: string

**type**

스케줄러의 유형입니다.

유형: string

**version**

AWS ParallelCluster 클러스터를 생성하는 데 사용된 버전.

유형: string

**validationMessages**

검증 수준이 `validationFailureLevel`보다 낮은 메시지 목록입니다. 메시지 목록은 구성 검증 중에 수집됩니다.

**id**

검사기의 ID입니다.

유형: string

**level**

검증 수준입니다.

유형: string

유효값: INFO | WARNING | ERROR

**message**

확인 메시지입니다.

유형: string

**type**

검사기의 유형입니다.

유형: string



## 예제

### Python

#### 요청

```
$ update_cluster(cluster_name_3x, path/config-file.yaml)
```

#### 200 응답

```
{
  'change_set': [
    {
      'current_value': '10',
      'parameter':
      'Scheduling.SlurmQueues[queue1].ComputeResources[t2micro].MaxCount',
      'requested_value': '15'
    }
  ],
  'cluster': {
    'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/test-api-cluster/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
    'cloudformation_stack_status': 'UPDATE_IN_PROGRESS',
    'cluster_name': 'cluster-3x',
    'cluster_status': 'UPDATE_IN_PROGRESS',
    'region': 'us-east-1',
    'scheduler': {
      'type': 'slurm'
    },
    'version': '3.2.1'
  }
}
```

## updateComputeFleet

클러스터 컴퓨팅 플릿의 상태를 업데이트합니다.

#### 주제

- [요청 구문](#)
- [요청 본문](#)

- [응답 구문](#)
- [응답 본문](#)
- [예제](#)

## 요청 구문

```
PATCH /v3/clusters/{clusterName}/computefleet
{
  "status": "string",
  "region": "string"
}
```

## 요청 본문

### clusterName

클러스터의 이름입니다.

유형: string

필수 항목 여부: 예

### status

컴퓨팅 플릿 상태입니다.

유형: string

유효값: START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

필수 사항 여부: 예

### region

AWS 리전 클러스터가 속한 곳.

유형: string

필수사항: 아니요

## 응답 구문

```
{
```

```
"status": "START_REQUESTED",
"lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

## 응답 본문

### status

컴퓨팅 플릿 상태입니다.

유형: string

유효값: START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED  
| STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdateTime

마지막 상태 업데이트 시간을 나타내는 타임스탬프입니다.

유형: 날짜/시간

## 예제

### Python

#### 요청

```
$ update_compute_fleet(cluster_name_3x, "START_REQUESTED")
```

#### 200 응답

```
{
  'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
    tzinfo=tzlocal()),
  'status': 'START_REQUESTED'
}
```

## AWS ParallelCluster Python 라이브러리 API

AWS ParallelCluster 버전 3.5.0부터 AWS ParallelCluster Python 라이브러리로 AWS ParallelCluster 에 액세스할 수 있습니다. 사용자의 `pccluster` 환경에서 또는 AWS Lambda 런타임 내에서 AWS

ParallelCluster 라이브러리에 액세스할 수 있습니다. AWS ParallelCluster Python 라이브러리를 사용하여 AWS ParallelCluster API에 액세스하는 방법을 알아봅니다. AWS ParallelCluster Python 라이브러리는 AWS ParallelCluster API가 제공하는 것과 동일한 기능을 제공합니다.

AWS ParallelCluster Python 라이브러리 작업 및 파라미터는 대문자 없이 snake\_case로 변환될 때 API 파라미터의 작업과 파라미터를 반영합니다.

## 주제

- [AWS ParallelCluster Python 라이브러리 권한 부여](#)
- [AWS ParallelCluster Python 라이브러리를 설치합니다.](#)
- [클러스터 API 작업](#)
- [컴퓨팅 플릿 API 작업](#)
- [클러스터 및 스택 로그 작업](#)
- [이미지 API 작업](#)
- [이미지 및 스택 로그 작업](#)
- [예제](#)
- [AWS ParallelCluster Python 라이브러리를 AWS Lambda](#)

## AWS ParallelCluster Python 라이브러리 권한 부여

boto3에 유효한 표준 방법 중 하나를 사용하여 보안 인증을 지정하세요. 자세한 내용은 [boto3 설명서](#)를 참조하세요.

## AWS ParallelCluster Python 라이브러리를 설치합니다.

1. [설 AWS ParallelCluster](#)에 제공된 지침에 따라 pcluster CLI 버전 3.5.0 이상을 설치합니다.
2. 다음 예제와 같이 pcluster 모듈을 가져오고 라이브러리 사용을 시작합니다.

```
import pcluster.lib as pc
pc.create_cluster(cluster_name="mycluster", cluster_configuration="config.yaml")
```

## 클러스터 API 작업

### 주제

- [list\\_clusters](#)

- [create\\_cluster](#)
- [delete\\_cluster](#)
- [describe\\_cluster](#)
- [update\\_cluster](#)

## list\_clusters

```
list_clusters(region, next_token, cluster_status)
```

기존 클러스터의 목록을 가져옵니다.

파라미터:

### region

지정된 AWS 리전에 배포된 클러스터를 나열합니다.

### next\_token

페이지가 매겨진 요청에 사용할 토큰입니다.

### cluster\_status

클러스터 상태별로 필터링합니다. 기본값은 모든 클러스터를 나열하는 것입니다.

유효한 값: CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE |  
UPDATE\_FAILED

## create\_cluster

```
create_cluster(cluster_name, cluster_configuration, region, suppress_validators,  
validation_failure_level, dry_run, rollback_on_failure, wait)
```

지정된 리전에 클러스터를 생성합니다.

파라미터:

### cluster\_name(필수)

클러스터 이름입니다.

## **cluster\_configuration**(필수)

Python 데이터 유형으로서의 클러스터 구성입니다.

### **region**

클러스터 AWS 리전.

### **suppress\_validators**

억제할 클러스터 구성 검사기를 하나 이상 식별합니다.

형식: (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

클러스터 생성 실패의 원인이 되는 최소 검증 수준입니다. 기본값은 ERROR입니다.

유효한 값: INFO | WARNING | ERROR

### **dry\_run**

리소스를 만들지 않고 요청 검증을 수행합니다. 이를 사용하여 클러스터 구성을 검증할 수 있습니다. 기본값은 False입니다.

### **rollback\_on\_failure**

True로 설정하면 AWS ParallelCluster가 실패 시 클러스터 스택 롤백을 자동으로 시작합니다. 기본값은 True입니다.

### **wait**

True로 설정하면 AWS ParallelCluster가 작업이 완료될 때까지 기다립니다. 기본값은 False입니다.

## **delete\_cluster**

```
delete_cluster(cluster_name, region, wait)
```

지정된 리전의 클러스터를 삭제합니다.

파라미터:

### **cluster\_name**(필수)

클러스터 이름입니다.

## region

클러스터 AWS 리전.

## wait

True로 설정된 경우 작업이 완료되기를 기다립니다. 기본값은 False입니다.

## describe\_cluster

```
describe_cluster(cluster_name, region)
```

기존 클러스터에 대한 세부 정보를 가져옵니다.

파라미터:

### cluster\_name(필수)

클러스터 이름입니다.

### region

클러스터 AWS 리전.

## update\_cluster

```
update_cluster(cluster_name, cluster_configuration, suppress_validators,  
validation_failure_level, region, force_update, dry_run, wait)
```

지정된 리전의 클러스터를 업데이트합니다.

파라미터:

### cluster\_name(필수)

클러스터 이름입니다.

### cluster\_configuration(필수)

Python 데이터 유형으로서의 클러스터 구성입니다.

## suppress\_validators

억제할 클러스터 구성 검사기를 하나 이상 식별합니다.

형식: (ALL | type:[A-Za-z0-9]+)

## validation\_failure\_level

클러스터 업데이트 실패의 원인이 되는 최소 검증 수준입니다. 기본값은 ERROR입니다.

유효한 값: INFO | WARNING | ERROR

## region

클러스터 AWS 리전.

## dry\_run

리소스를 생성하거나 업데이트하지 않고 요청 검증을 수행합니다. 이를 사용하여 클러스터 구성을 검증할 수 있습니다. 기본값은 False입니다.

## force\_update

True로 설정하면 업데이트 유효성 검사 오류를 무시하여 업데이트를 강제로 수행합니다. 기본값은 False입니다.

## wait

True로 설정된 경우 작업이 완료되기를 기다립니다. 기본값은 False입니다.

## 컴퓨팅 플릿 API 작업

주제

- [describe\\_compute\\_fleet](#)
- [update\\_compute\\_fleet](#)
- [delete\\_cluster\\_instances](#)
- [describe\\_cluster\\_instances](#)

## describe\_compute\_fleet

```
describe_compute_fleet(cluster_name, region)
```

특정 클러스터의 클러스터 컴퓨팅 플릿 상태를 설명합니다.



파라미터:

**cluster\_name**(필수)

클러스터 이름입니다.

**region**

지정된 AWS 리전에 배포된 클러스터의 컴퓨팅 플릿 상태를 설명합니다.

## update\_compute\_fleet

```
update_compute_fleet(cluster_name, status, region)
```

클러스터 컴퓨팅 플릿의 상태를 업데이트합니다.

파라미터:

**cluster\_name**(필수)

클러스터 이름입니다.

**status**(필수)

업데이트할 상태입니다.

유효한 값: START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

**region**

클러스터 AWS 리전.

## delete\_cluster\_instances

```
delete_cluster_instances(cluster_name, region, force)
```

지정된 리전의 클러스터를 삭제합니다.

파라미터:

**cluster\_name**(필수)

클러스터 이름입니다.

## region

클러스터 AWS 리전.

## force

True로 설정하면 지정된 `cluster_name`의 클러스터를 찾을 수 없을 때 강제로 삭제합니다. 기본 값은 False입니다.

## describe\_cluster\_instances

```
describe_cluster_instances(cluster_name, region, next_token, node_type, queue_name)
```

클러스터 인스턴스를 설명합니다.

파라미터:

### cluster\_name(필수)

클러스터 이름입니다.

### region

클러스터 AWS 리전.

### next\_token

페이지가 매겨진 요청에 사용할 토큰입니다.

### node\_type

인스턴스를 `node_type`으로 필터링합니다.

유효한 값: HeadNode | ComputeNode

### queue\_name

대기열 이름을 기준으로 인스턴스를 필터링합니다.

## 클러스터 및 스택 로그 작업

주제

- [list\\_cluster\\_log\\_streams](#)
- [get\\_cluster\\_log\\_events](#)

- [get\\_cluster\\_stack\\_events](#)

## list\_cluster\_log\_streams

```
list_cluster_log_streams(cluster_name, region, filters, next_token)
```

지정된 클러스터에 대한 로그 스트림을 나열합니다.

파라미터:

**cluster\_name**(필수)

클러스터 이름입니다.

**region**

클러스터 AWS 리전.

**filters**

클러스터 로그 스트림을 필터링합니다.

형식: 'Name=a,Values=1 Name=b,Values=2,3'

허용되는 필터:

code-dns-name

인스턴스의 프라이빗 DNS 이름의 약식입니다(예: ip-10-0-0-101).

node-type

노드 유형입니다.

유효한 값: HeadNode

**next\_token**

페이지가 매겨진 요청에 사용할 토큰입니다.

## get\_cluster\_log\_events

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,  
start_from_head, limit, start_time, end_time)
```

지정된 클러스터 및 로그 스트림에 대한 로그 이벤트를 가져옵니다.

파라미터:

**cluster\_name**(필수)

클러스터 이름입니다.

**log\_stream\_name**(필수)

로그 스트림 이름입니다.

**region**

클러스터 AWS 리전.

**next\_token**

페이지가 매겨진 요청에 사용할 토큰입니다.

**start\_from\_head**

True로 설정하면 AWS ParallelCluster가 가장 빠른 로그 이벤트를 먼저 반환합니다. False로 설정하면 최신 로그 이벤트를 먼저 반환합니다. 기본값은 False입니다.

**limit**

반환된 로그 이벤트의 최대 수입니다. 값을 지정하지 않는 경우 최대값은 1MB의 응답 크기에 들어갈 수 있는 로그 수(최대 10,000개의 로그 이벤트)입니다.

**start\_time**

로그 이벤트 시간 범위의 시작입니다. ISO 8601 형식(예: '2021-01-01T20:00:00Z')으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트가 포함됩니다.

**end\_time**

로그 이벤트 시간 범위의 끝입니다. ISO 8601 형식(예: '2021-01-01T20:00:00Z')으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다.

## get\_cluster\_stack\_events

```
get_cluster_stack_events(cluster_name, region, next_token)
```

지정된 클러스터에 대한 스택 이벤트를 가져옵니다.

파라미터:

**cluster\_name**(필수)

클러스터 이름입니다.

**region**

클러스터 AWS 리전.

**next\_token**

페이지가 매겨진 요청에 사용할 토큰입니다.

## 이미지 API 작업

주제

- [list\\_images](#)
- [build\\_image](#)
- [delete\\_image](#)
- [describe\\_image](#)

### list\_images

```
list_images(image_status, region, next_token)
```

기존 이미지 목록을 가져옵니다.

파라미터:

**image\_status**(필수)

이미지 상태별로 필터링합니다.

유효한 값: AVAILABLE | PENDING | FAILED

**region**

지정된 AWS 리전에 빌드된 이미지를 나열합니다.

## next\_token

페이지를 매긴 요청에 사용할 토큰입니다.

## build\_image

```
build_image(image_configuration, image_id, suppress_validators,  
            validation_failure_level, dry_run, rollback_on_failure, region)
```

특정 리전에서 사용자 지정 AWS ParallelCluster 이미지를 만드세요.

파라미터:

### image\_configuration(필수)

Python 데이터로서의 이미지 구성입니다.

### image\_id(필수)

이미지 ID입니다.

### suppress\_validators

억제할 이미지 구성 검사기를 하나 이상 식별합니다.

형식: (ALL | type:[A-Za-z0-9]+)

### validation\_failure\_level

이미지 생성 실패의 원인이 되는 최소 검증 수준입니다. 기본값은 ERROR입니다.

유효한 값: INFO | WARNING | ERROR

### dry\_run

True로 설정하면 AWS ParallelCluster가 리소스를 생성하지 않고 요청 검증을 수행합니다. 이를 사용하여 이미지 구성을 검증할 수 있습니다. 기본값은 False입니다.

### rollback\_on\_failure

True로 설정하면 AWS ParallelCluster가 실패 시 이미지 스택 롤백을 자동으로 시작합니다. 기본값은 False입니다.

### region

AWS 리전 이미지입니다.

## delete\_image

```
delete_image(image_id, region, force)
```

지정된 리전의 이미지를 삭제합니다.

파라미터:

**image\_id**(필수)

이미지 ID입니다.

**region**

AWS 리전 이미지입니다.

**force**

True로 설정하면 AWS ParallelCluster는 인스턴스가 AMI를 사용 중이거나 AMI를 공유하는 경우 강제로 삭제합니다. 기본값은 False입니다.

## describe\_image

```
describe_image(image_id, region)
```

기존 이미지에 대한 세부 정보를 가져옵니다.

파라미터:

**image\_id**(필수)

이미지 ID입니다.

**region**

AWS 리전 이미지입니다.

## 이미지 및 스택 로그 작업

주제

- [list\\_image\\_log\\_streams](#)
- [get\\_image\\_log\\_events](#)

- [get\\_image\\_stack\\_events](#)
- [list\\_official\\_images](#)

## list\_image\_log\_streams

```
list_image_log_streams(image_id, region, next_token)
```

이미지의 로그 스트림을 나열합니다.

파라미터:

**image\_id**(필수)

이미지 ID입니다.

**region**

AWS 리전 이미지입니다.

**next\_token**

페이지가 매겨진 요청에 사용할 토큰입니다.

## get\_image\_log\_events

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head, limit, start_time, end_time)
```

지정된 이미지 및 로그 스트림에 대한 로그 이벤트를 가져옵니다.

파라미터:

**image\_id**(필수)

이미지 ID입니다.

**log\_stream\_name**(필수)

로그 스트림 이름입니다.

**region**

AWS 리전 이미지입니다.



## **next\_token**

페이지가 매겨진 요청에 사용할 토큰입니다.

## **start\_from\_head**

True로 설정하면 AWS ParallelCluster가 가장 빠른 로그 이벤트를 먼저 반환합니다. False로 설정하면 최신 로그 이벤트를 먼저 반환합니다. 기본값은 False입니다.

## **limit**

반환된 로그 이벤트의 최대 수입니다. 값을 지정하지 않는 경우 최대값은 1MB의 응답 크기에 들어갈 수 있는 로그 수(최대 10,000개의 로그 이벤트)입니다.

## **start\_time**

로그 이벤트 시간 범위의 시작입니다. ISO 8601 형식(예: '2021-01-01T20:00:00Z')으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트가 포함됩니다.

## **end\_time**

로그 이벤트 시간 범위의 끝입니다. ISO 8601 형식(예: '2021-01-01T20:00:00Z')으로 표현됩니다. 타임스탬프가 이 시간 이후인 이벤트는 포함되지 않습니다.

## **get\_image\_stack\_events**

```
get_image_stack_events(image_id, region, next_token)
```

지정된 이미지에 대한 스택 이벤트를 가져옵니다.

파라미터:

### **image\_id**(필수)

이미지 ID입니다.

### **region**

AWS 리전 이미지입니다.

### **next\_token**

페이지가 매겨진 요청에 사용할 토큰입니다.

## list\_official\_images

```
list_official_images(region,os, architecture)
```

공식 AWS ParallelCluster 이미지 목록을 검색하세요.

파라미터:

### region

AWS 리전 이미지입니다.

### os

운영 체제 배포를 기준으로 필터링합니다. 기본값은 필터링 없음입니다.

### architecture

아키텍처별로 필터링합니다. 기본값은 필터링 없음입니다.

## 예제

주제

- [클러스터 생성](#)

## 클러스터 생성

지정된 입력을 환경에 저장하여 다음 예제 스크립트를 실행하면 클러스터가 생성됩니다. 클러스터 구성은 [클러스터 구성 설명서](#)를 기반으로 Python 데이터 유형으로 생성됩니다.

```
import os
import pprint
import pcluster.lib as pc
pp = pprint.PrettyPrinter()

HEAD_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
COMPUTE_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
KEY_NAME = os.environ["KEY_NAME"]
CONFIG = {'Image': {'Os': 'alinux2'},
          'HeadNode': {'InstanceType': 't2.large',
                       'Networking': {'SubnetId': HEAD_NODE_SUBNET}},
```

```

        'Ssh': {'KeyName': KEY_NAME}},

    'Scheduling': {'Scheduler': 'slurm',
                  'SlurmQueues':
                    [{'Name': 'queue0',
                      'ComputeResources':
                        [{'Name': 'queue0-i0', 'InstanceType': 't2.micro',
                          'MinCount': 0, 'MaxCount': 10}],
                      'Networking': {'SubnetIds': [COMPUTE_NODE_SUBNET]}]}}}

pp.pprint(pc.create_cluster(cluster_name="mycluster", cluster_configuration=CONFIG))

```

출력:

```

{'cluster': {'cloudformationStackArn': 'arn:aws:cloudformation:us-
east-2:123456789012:stack/mycluster/00000000-aaaa-1111-999-000000000000',
            'cloudformationStackStatus': 'CREATE_IN_PROGRESS',
            'clusterName': 'mycluster',
            'clusterStatus': 'CREATE_IN_PROGRESS',
            'region': 'us-east-2',
            'scheduler': {'type': 'slurm'},
            'version': '3.7.0'}}

```

## AWS ParallelCluster Python 라이브러리를 AWS Lambda

Lambda 계층과 런타임을 배포하여 AWS ParallelCluster Python 라이브러리에 액세스할 수 있습니다. 다음 단계에 설명된 대로 AWS ParallelCluster zip 파일에 대한 링크를 입력하여 사용할 수 있는 zip 파일을 호스팅합니다. Lambda는 zip 파일을 사용하여 Python 라이브러리에 대한 액세스를 지원하는 런타임 환경을 준비합니다. AWS ParallelCluster Python 라이브러리는 AWS ParallelCluster 버전 3.5.0과 함께 추가되었습니다. 버전 3.5.0 이상에 한해 해당 라이브러리를 사용할 수 있습니다.

호스팅된 zip 파일 URL의 형식은 `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`입니다.

AWS Lambda로 AWS ParallelCluster Python 라이브러리에 액세스를 시작하세요.

### Lambda 계층 생성

1. AWS Management Console에 로그인하고 AWS Lambda 콘솔로 이동하세요.
2. 탐색 창에서 계층을 선택한 다음 계층 생성을 선택합니다.

- 계층 이름을 입력하고 Amazon S3에서 파일 업로드를 선택합니다.
- zip 파일의 URL을 입력합니다. `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`.
- 호환되는 아키텍처에서 x86\_64 아키텍처를 선택합니다.
- 호환되는 런타임에서 Python 3.9 런타임을 선택합니다.
- 생성을 선택합니다.

## Lambda 계층 사용

- Lambda 콘솔 탐색 창에서 함수를 선택한 다음 함수 생성을 선택합니다.
- 함수 이름을 입력합니다.
- 런타임에서 Python 3.9 런타임을 선택합니다.
- 아키텍처에서 x86\_64 아키텍처를 선택합니다.
- 함수 생성을 선택합니다.
- 함수를 생성한 후 레이어를 선택하고 레이어 추가를 선택합니다.
- 사용자 지정 레이어를 선택하고 이전 단계에서 만든 레이어를 선택합니다.
- 계층 버전을 선택합니다.
- 추가를 선택합니다.
- Lambda에는 AWS ParallelCluster로 생성된 클러스터를 관리할 권한이 필요합니다. [기본 AWS ParallelCluster pcluster 사용자 정책](#)에 나열된 권한을 사용하여 Lambda 역할을 생성합니다.

이제 [AWS ParallelCluster Python 라이브러리 API](#)에 설명된 대로 Python 라이브러리에서 AWS ParallelCluster에 액세스할 수 있습니다.

# AWS ParallelCluster 작동 방식

AWS ParallelCluster는 클러스터를 관리하는 방법뿐만 아니라 AWS 서비스를 사용하여 HPC 환경을 구축하는 방법에 대한 참조로 작성되었습니다.

주제

- [AWS ParallelCluster 프로세스](#)
- [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)
- [AWS ParallelCluster내부 디렉터리](#)

## AWS ParallelCluster 프로세스

이 섹션은 Slurm과 함께 배포되는 클러스터에 적용됩니다. 이 스케줄러와 함께 사용하면 기본 작업 스케줄러와 상호 작용하여 컴퓨팅 노드 프로비저닝 및 제거를 AWS ParallelCluster 관리합니다.

를 기반으로 하는 HPC 클러스터의 AWS Batch 경우 컴퓨팅 AWS ParallelCluster 노드 관리를 위해 에서 제공하는 기능을 사용합니다. AWS Batch

### clustermgtd

클러스터 관리 데몬(daemon)은 다음 작업을 수행합니다.

- 비활성 파티션 정리
- 용량 블록과 관련된 Slurm 예약 및 노드 관리 (다음 섹션 참조)
- 정적 용량 관리: 정적 용량이 항상 정상 상태인지 확인하세요.
- 스케줄러를 Amazon EC2와 동기화합니다.
- 분리된 인스턴스 정리
- 일시 중지 워크플로 외부에서 발생하는 Amazon EC2 종료 시 스케줄러 노드 상태 복원
- 비정상 Amazon EC2 인스턴스 관리(Amazon EC2 상태 확인 실패)
- 정기 유지 관리 이벤트 관리
- 비정상 스케줄러 노드 관리(스케줄러 상태 점검 실패)

## 용량 블록과 관련된 Slurm 예약 및 노드 관리

ParallelCluster 온디맨드 용량 예약 (ODCR) 및 기계 학습용 용량 블록 (CB) 을 지원합니다. ODCR과 달리 CB는 향후 시작 시간을 가질 수 있으며 기한이 정해져 있습니다.

Clustermgtd는 루프에서 비정상 노드를 검색하여 다운된 모든 Amazon EC2 인스턴스를 종료하고 정적 노드인 경우 새 인스턴스로 교체합니다.

ParallelCluster 용량 블록과 관련된 정적 노드를 다르게 관리합니다. AWS ParallelCluster CB가 아직 활성화되지 않은 경우에도 클러스터를 생성하고, CB가 활성화되면 인스턴스가 자동으로 시작됩니다.

아직 활성화되지 않은 CB와 관련된 컴퓨팅 리소스에 해당하는 Slurm 노드는 CB 시작 시간에 도달할 때까지 유지 관리 상태로 유지됩니다. Slurm노드는 Slurm 관리자 사용자와 연결된 예약/유지 관리 상태로 유지되므로 작업은 수락할 수 있지만 예약이 제거될 때까지 작업은 보류 상태로 유지됩니다.

### Slurm

Clustermgtd는 Slurm 예약을 자동으로 생성/삭제하여 CB 상태에 따라 관련 CB 노드를 유지 관리 상태로 전환합니다. CB가 활성화되면 Slurm 예약이 제거되고 노드가 시작되며 보류 중인 작업이나 새 작업 제출에 사용할 수 있게 됩니다.

CB 종료 시간에 도달하면 노드는 예약/유지 관리 상태로 다시 이동합니다. CB가 더 이상 활성화되지 않고 인스턴스가 종료된 경우 새 대기열/컴퓨팅 리소스에 작업을 다시 제출하거나 다시 등록하는 것은 사용자의 몫입니다.

## clusterstatusmgtd

클러스터 상태 관리 데몬(daemon)은 컴퓨팅 플릿 상태 업데이트를 관리합니다. 매 분마다 DynamoDB 테이블에 저장된 플릿 상태를 가져오고 모든 STOP/START 요청을 관리합니다.

## computemgtd

컴퓨팅 관리 데몬(daemon)(computemgtd) 프로세스는 각 클러스터 컴퓨팅 노드에서 실행됩니다. 컴퓨팅 관리 데몬(daemon)은 5분마다 헤드 노드에 연결할 수 있고 정상 상태인지 확인합니다. 헤드 노드에 도달할 수 없거나 정상이 아닌 상태로 5분이 경과하면 컴퓨팅 노드가 종료됩니다.

## AWS 에서 사용하는 서비스 AWS ParallelCluster

에서 사용하는 Amazon Web Services (AWS) 서비스는 다음과 같습니다 AWS ParallelCluster.

주제

- [Amazon API Gateway](#)
- [AWS Batch](#)
- [AWS CloudFormation](#)
- [아마존 CloudWatch](#)
- [아마존 CloudWatch 이벤트](#)
- [아마존 CloudWatch 로그](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic 컨테이너 레지스트리](#)
- [Amazon EFS](#)
- [Amazon FSx for Lustre](#)
- [ONTAP용 아마존 NetApp FSx](#)
- [Amazon FSx for OpenZFS](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon RDS](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service](#)
- [Amazon Simple Storage Service\(S3\)](#)
- [Amazon VPC](#)
- [Elastic Fabric Adapter](#)
- [EC2 Image Builder](#)
- [NICE DCV](#)

## Amazon API Gateway

Amazon API Gateway는 모든 규모에서 REST, HTTP 및 WebSocket API를 생성, 게시, 유지 관리, 모니터링 및 보호하는 AWS 서비스입니다.

AWS ParallelCluster API Gateway를 사용하여 AWS ParallelCluster API를 호스팅합니다.

에 대한 AWS Batch자세한 내용은 <https://aws.amazon.com/api-gateway/> 및 <https://docs.aws.amazon.com/apigateway/> 을 참조하십시오.

## AWS Batch

AWS Batch AWS 관리형 작업 스케줄러 서비스입니다. 최적의 양과 유형의 컴퓨팅 리소스 (예: CPU 또는 메모리 최적화 인스턴스) 를 클러스터에 동적으로 프로비저닝합니다. AWS Batch 이러한 리소스는 볼륨 요구 사항을 포함하여 일괄 작업의 특정 요구 사항을 기반으로 프로비저닝됩니다. 를 사용하면 작업을 효과적으로 실행하기 위해 추가 배치 컴퓨팅 소프트웨어 또는 서버 클러스터를 설치하거나 관리할 필요가 없습니다. AWS Batch

AWS Batch AWS Batch 클러스터에만 사용됩니다.

에 대한 AWS Batch자세한 내용은 <https://aws.amazon.com/batch/> 및 <https://docs.aws.amazon.com/batch/> 을 참조하십시오.

## AWS CloudFormation

AWS CloudFormation 클라우드 환경에서 타사 애플리케이션 리소스를 AWS 모델링하고 프로비저닝하기 위한 공통 언어를 제공하는 infrastructure-as-code 서비스입니다. 에서 주로 사용하는 AWS ParallelCluster서비스입니다. 의 각 AWS ParallelCluster 클러스터는 스택으로 표시되며 각 클러스터에 필요한 모든 리소스는 AWS ParallelCluster AWS CloudFormation 템플릿 내에 정의됩니다. 대부분의 경우 AWS ParallelCluster CLI 명령은 생성, 업데이트 및 삭제 명령과 같은 AWS CloudFormation 스택 명령에 직접 대응합니다. 클러스터 내에서 시작된 인스턴스는 클러스터가 시작된 AWS CloudFormation 엔드포인트에 HTTPS 호출을 수행합니다. AWS 리전

에 대한 AWS CloudFormation자세한 내용은 <https://aws.amazon.com/cloudformation/> 및 <https://docs.aws.amazon.com/cloudformation/> 을 참조하십시오.

## 아마존 CloudWatch

Amazon CloudWatch (CloudWatch) 은 데이터와 실행 가능한 통찰력을 제공하는 모니터링 및 관찰 서비스입니다. 이러한 인사이트를 사용하여 애플리케이션을 모니터링하고, 성능 변화 및 서비스 예외에 대응하고, 리소스 활용도를 최적화할 수 있습니다. AWS ParallelCluster에서는 대시보드로 CloudWatch 사용되어 Docker 이미지 빌드 단계와 작업 출력을 모니터링하고 기록합니다. AWS Batch

AWS ParallelCluster 버전 2.10.0 이전에는 클러스터에서만 CloudWatch 사용되었습니다. AWS Batch



에 대한 자세한 내용은 <https://aws.amazon.com/cloudwatch/> 및 <https://docs.aws.amazon.com/cloudwatch/> 을 참조하십시오. CloudWatch

## 아마존 CloudWatch 이벤트

Amazon CloudWatch CloudWatch Events (Events) 는 Amazon Web Services (AWS) 리소스의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 신속하게 설정할 수 있는 단순 규칙을 사용하여 일치하는 이벤트를 검색하고 하나 이상의 대상 함수 또는 스트림으로 이를 라우팅할 수 있습니다. AWS ParallelCluster에서는 CloudWatch 이벤트가 AWS Batch 작업에 사용됩니다.

CloudWatch 이벤트에 대한 자세한 내용은 <https://docs.aws.amazon.com//eventbridge/latest/userguide/> 을 참조하십시오 eb-cwe-now-eb.

## 아마존 CloudWatch 로그

아마존 CloudWatch CloudWatch 로그 (Logs) 는 아마존의 핵심 기능 중 하나입니다 CloudWatch. 이것은 AWS ParallelCluster에서 사용하는 많은 구성 요소에 대한 로그 파일을 모니터링, 저장, 확인 및 검색하는 데 사용됩니다.

AWS ParallelCluster 버전 2.6.0 이전에는 CloudWatch 로그가 클러스터에서만 사용되었습니다. AWS Batch

자세한 정보는 [Amazon CloudWatch Logs와 통합](#) 을 참조하세요.

## AWS CodeBuild

AWS CodeBuild (CodeBuild) 는 소스 코드를 컴파일하고, 테스트를 실행하고, 배포 준비가 완료된 소프트웨어 패키지를 생성하는 AWS 관리형 지속적 통합 서비스입니다. AWS ParallelCluster에서는 CodeBuild 클러스터를 생성할 때 Docker 이미지를 자동으로 투명하게 빌드하는 데 사용됩니다.

CodeBuild 클러스터에만 사용됩니다. AWS Batch

에 대한 CodeBuild 자세한 내용은 <https://aws.amazon.com/codebuild/> 및 <https://docs.aws.amazon.com/codebuild/> 을 참조하십시오.

## Amazon DynamoDB

Amazon DynamoDB(이하 DynamoDB)는 빠르고 유연한 NoSQL 데이터베이스 서비스입니다. 이것은 클러스터의 최소 상태 정보를 저장하는 데 사용됩니다. 헤드 노드는 DynamoDB 테이블의 프로비저닝된 인스턴스를 추적합니다.

DynamoDB는 클러스터와 함께 사용되지 않습니다. AWS Batch

DynamoDB에 대한 자세한 내용은 <https://aws.amazon.com/dynamodb/> 및 <https://docs.aws.amazon.com/dynamodb/>을 참조하세요.

## Amazon Elastic Block Store

Amazon Elastic Block Store(Amazon EBS)는 공유 볼륨을 위한 영구 스토리지를 제공하는 고성능 블록 스토리지 서비스입니다. 모든 Amazon EBS 설정을 구성을 통해 전달할 수 있습니다. Amazon EBS 볼륨은 빈 상태로 초기화하거나 기존 Amazon EBS 스냅샷에서 초기화할 수 있습니다.

Amazon EBS에 대한 자세한 내용은 <https://aws.amazon.com/ebs/> 및 <https://docs.aws.amazon.com/ebs/>을 참조하세요.

## Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) 는 에 필요한 컴퓨팅 용량을 제공합니다. AWS ParallelCluster헤드 및 컴퓨팅 노드는 Amazon EC2 인스턴스입니다. HVM을 지원하는 모든 인스턴스 유형을 선택할 수 있습니다. 헤드 노드와 컴퓨팅 노드는 서로 다른 인스턴스 유형일 수 있습니다. 또한 여러 대기열을 사용하는 경우 컴퓨팅 노드 일부 또는 전체를 스팟 인스턴스로 시작할 수도 있습니다. 인스턴스에 있는 인스턴스 스토어 볼륨은 스트라이프된 LVM 볼륨으로 탑재됩니다.

Amazon EC2에 대한 자세한 내용은 <https://aws.amazon.com/ec2/> 및 <https://docs.aws.amazon.com/ec2/> 을 참조하십시오.

## Amazon Elastic 컨테이너 레지스트리

Amazon Elastic Container Registry(Amazon ECR)는 개발자가 Docker 컨테이너 이미지를 간편하게 저장, 관리 및 배포할 수 있게 해주는 완전관리형 Docker 컨테이너 레지스트리입니다. 에서 AWS ParallelCluster Amazon ECR은 클러스터가 생성될 때 구축되는 Docker 이미지를 저장합니다. 그런 다음 Docker 이미지는 제출된 작업의 컨테이너를 실행하는 AWS Batch 데 사용됩니다.

Amazon ECR은 AWS Batch 클러스터에서만 사용됩니다.

자세한 내용은 <https://aws.amazon.com/ecr/> 및 <https://docs.aws.amazon.com/ecr/>을 참조하세요.

## Amazon EFS

Amazon Elastic File System(Amazon EFS)는 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 간단하고 확장 가능한 완전 관리형 탄력적 NFS 파일 시스템을 제공합니다.

[EfsSettings](#)가 지정된 경우 Amazon EFS가 사용됩니다. Amazon EFS에 대한 지원은 AWS ParallelCluster 버전 2.1.0에 추가되었습니다.

Amazon EFS에 대한 자세한 내용은 <https://aws.amazon.com/efs/> 및 <https://docs.aws.amazon.com/efs/>를 참조하세요.

## Amazon FSx for Lustre

FSx for Lustre는 오픈 소스 Lustre 파일 시스템을 사용하는 고성능 파일 시스템을 제공합니다.

[FsxLustreSettings](#) 속성이 지정되면 FSx for Lustre가 사용됩니다. FSx for Lustre에 대한 지원은 버전 2.2.1에 추가되었습니다. AWS ParallelCluster

FSx for Lustre에 대한 자세한 내용은 <https://aws.amazon.com/fsx/lustre/> 및 <https://docs.aws.amazon.com/fsx/>을 참조하세요.

## ONTAP용 아마존 NetApp FSx

FSx for ONTAP는 널리 사용되는 ONTAP 파일 시스템에 구축된 NetApp 완전 관리형 공유 스토리지 시스템을 제공합니다. [FsxOntapSettings](#) 속성이 지정되면 FSx for ONTAP이 사용됩니다. ONTAP용 FSx 지원은 버전 3.2.0에 추가되었습니다. AWS ParallelCluster

FSx for ONTAP에 대한 자세한 내용은 <https://aws.amazon.com/fsx/netapp-ontap/> 및 <https://docs.aws.amazon.com/fsx/>을 참조하세요.

## Amazon FSx for OpenZFS

FSx for OpenZFS는 인기 있는 OpenZFS 파일 시스템을 기반으로 구축된 완전 관리형 공유 스토리지 시스템을 제공합니다. [FsxOpenZfsSettings](#) 속성이 지정되면 FSx for OpenZFS가 사용됩니다. OpenZFS를 위한 FSX에 대한 지원은 버전 3.2.0에서 추가되었습니다. AWS ParallelCluster

FSx for OpenZFS에 대한 자세한 내용은 <https://aws.amazon.com/fsx/openzfs/> 및 <https://docs.aws.amazon.com/fsx/>을 참조하세요.

## AWS Identity and Access Management

AWS Identity and Access Management (IAM)은 각 개별 클러스터의 인스턴스에 대해 Amazon EC2에 대해 최소 권한 IAM 역할을 제공하는 AWS ParallelCluster 데 사용됩니다. AWS ParallelCluster 인스턴스는 클러스터를 배포하고 관리하는 데 필요한 특정 API 호출에 대한 액세스 권한만 부여됩니다.

AWS Batch 클러스터를 사용하면 클러스터를 생성할 때 Docker 이미지 구축 프로세스와 관련된 구성 요소에 대한 IAM 역할도 생성됩니다. 이러한 구성 요소에는 Amazon ECR 리포지토리에, 또는 그 리

포지토리로부터 도커 이미지를 추가 및 삭제할 수 있는 Lambda 함수가 포함됩니다. 또한 클러스터와 CodeBuild 프로젝트용으로 생성된 Amazon S3 버킷을 삭제할 수 있는 함수도 포함됩니다. AWS Batch 리소스, 인스턴스 및 작업에 대한 역할도 있습니다.

IAM에 대한 자세한 내용은 <https://aws.amazon.com/iam/> 및 <https://docs.aws.amazon.com/iam/>을 참조하세요.

## AWS Lambda

AWS Lambda (Lambda) 는 Docker 이미지 생성을 오케스트레이션하는 함수를 실행합니다. 또한 Lambda는 Amazon ECR 리포지토리 및 Amazon S3에 저장된 도커 이미지와 같은 사용자 지정 클러스터 리소스의 정리를 관리합니다.

Lambda에 대한 자세한 내용은 <https://aws.amazon.com/lambda/> 및 <https://docs.aws.amazon.com/lambda/>를 참조하세요.

## Amazon RDS

Amazon Relational Database Service (Amazon RDS) 는 클라우드에서 관계형 데이터베이스를 더 쉽게 설정, 운영 및 확장할 수 있게 해주는 웹 서비스입니다. AWS

AWS ParallelCluster AWS Batch 및 Slurm 에 Amazon RDS를 사용합니다.

Amazon RDS에 대한 자세한 내용은 <https://aws.amazon.com/rds/> 및 <https://docs.aws.amazon.com/rds/>를 참조하세요.

## Amazon Route 53

Amazon Route 53(Route 53)은 각 컴퓨팅 노드에 대해 호스트 이름과 정규화된 도메인 이름을 포함하는 호스팅 영역을 생성하는 데 사용됩니다.

Route 53에 대한 자세한 내용은 <https://aws.amazon.com/route53/> 및 <https://docs.aws.amazon.com/route53/>을 참조하세요.

## Amazon Simple Notification Service

Amazon SNS는 게시자에서 구독자(생산자 및 소비자라고도 함)로 메시지를 전송하는 관리형 서비스입니다.

AWS ParallelCluster API 호스팅에 Amazon SNS를 사용합니다.

Amazon SNS에 대한 자세한 내용은 <https://aws.amazon.com/sns/> 및 <https://docs.aws.amazon.com/sns/>를 참조하세요.

## Amazon Simple Storage Service(S3)

Amazon Simple Storage 서비스 (Amazon S3) 는 각 템플릿에 템플릿을 AWS ParallelCluster 저장합니다. AWS 리전 AWS ParallelCluster CLI/SDK 도구가 Amazon S3를 사용할 수 있도록 구성할 수 있습니다.

AWS ParallelCluster 또한 클러스터에서 사용하는 리소스 (예: 클러스터 구성 파일) 를 AWS 계정 저장하기 위해 Amazon S3 버킷을 생성합니다. AWS ParallelCluster 클러스터를 AWS 리전 생성하는 각 버킷에 Amazon S3 버킷 1개를 유지 관리합니다.

AWS Batch 클러스터를 사용하면 계정의 Amazon S3 버킷이 관련 데이터를 저장하는 데 사용됩니다. 예를 들어, 이 버킷은 도커 이미지와 스크립트가 제출된 작업에서 생성될 때 생성된 아티팩트를 저장합니다.

자세한 내용은 <https://aws.amazon.com/s3/> 및 <https://docs.aws.amazon.com/s3/>을 참조하세요.

## Amazon VPC

Amazon VPC는 클러스터의 노드가 사용하는 네트워크를 정의합니다.

아마존 VPC에 대한 자세한 내용은 <https://aws.amazon.com/vpc/> 및 <https://docs.aws.amazon.com/vpc/>을 참조하세요.

## Elastic Fabric Adapter

EFA (Elastic Fabric Adapter) 는 고객이 대규모 노드 간 통신이 필요한 애플리케이션을 실행하는 데 사용할 수 있는 인스턴스용 네트워크 인터페이스입니다. AWS

EC2 Image Builder에 대한 자세한 내용은 <https://aws.amazon.com/hpc/efa/>을 참조하세요.

## EC2 Image Builder

EC2 Image Builder는 사용자 지정되고 안전한 서버 이미지의 생성, 관리 및 배포를 자동화하는 데 도움이 되는 완전관리형 서비스입니다. up-to-date

AWS ParallelCluster Image Builder를 사용하여 AWS ParallelCluster 이미지를 만들고 관리합니다.

EC2 Image Builder에 대한 자세한 내용은 <https://aws.amazon.com/image-builder/> 및 <https://docs.aws.amazon.com/imagebuilder/>을 참조하세요.

## NICE DCV

NICE DCV는 다양한 네트워크 조건에서 모든 장치에 원격 데스크톱 및 애플리케이션 스트리밍을 제공하는 안전한 방법을 제공하는 고성능 원격 디스플레이 프로토콜입니다. [HeadNode 섹션/Dcv](#) 설정이 지정되면 NICE DCV가 사용됩니다. NICE DCV에 대한 지원은 버전 2.5.0에서 추가되었습니다 AWS ParallelCluster .

NICE DCV에 대한 자세한 내용은 <https://aws.amazon.com/hpc/dcv/> 및 <https://docs.aws.amazon.com/dcv/>을 참조하세요.

## AWS ParallelCluster내부 디렉터리

클러스터 내에서 데이터를 공유하는 데 AWS ParallelCluster 사용되는 여러 내부 디렉터리가 있습니다. 헤드 노드, 컴퓨팅 노드, 로그인 노드 간에 공유되는 디렉터리는 다음과 같습니다.

/opt/slurm

/opt/intel

/opt/parallelcluster/shared (only with compute nodes)

/opt/parallelcluster/shared\_login\_nodes (only with login nodes)

/home (unless specified in SharedStorage)

### Note

기본적으로 이러한 디렉터리는 헤드 노드 EBS 볼륨에 생성되며 컴퓨팅 및 로그인 노드로의 NFS 내보내기 시 공유됩니다. AWS ParallelCluster 3.8부터 파라미터를 AWS ParallelCluster `efs`로 설정하여 이러한 디렉터리를 호스팅하고 공유하는 Amazon EFS 파일 시스템을 생성 및 관리할 수 있습니다. [SharedStorageType](#)

클러스터가 확장되면 EBS 볼륨을 통한 NFS 내보내기가 성능 병목 현상을 일으킬 수 있습니다. EFS를 사용하면 클러스터가 확장될 때 NFS 내보내기를 피하고 클러스터와 관련된 성능 병목 현상을 피할 수 있습니다.

# 자습서

다음 자습서는 AWS ParallelCluster 버전 3을 시작하는 방법을 보여주고 몇 가지 일반적인 작업에 대한 모범 사례 지침을 제공합니다.

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 내용은 [AWS ParallelCluster UI 비용\(률\)](#) 참조하세요.

## 주제

- [에서 첫 작업 실행 AWS ParallelCluster](#)
- [사용자 지정 AWS ParallelCluster AMI 구축](#)
- [Active Directory 통합](#)
- [AWS KMS 키를 사용한 공유 스토리지 암호화 구성](#)
- [다중 대기열 모드 클러스터에서 작업 실행](#)
- [AWS ParallelCluster API 사용하기](#)
- [Slurm 화계를 사용하여 클러스터 생성](#)
- [외부 Slurmdbd 어카운팅을 사용하여 클러스터 생성](#)
- [이전 AWS Systems Manager 문서 버전으로 되돌리기](#)
- [를 사용하여 클러스터 생성 AWS CloudFormation](#)
- [테라폼으로 ParallelCluster API 배포하기](#)
- [테라폼으로 클러스터 만들기](#)
- [테라폼으로 사용자 지정 AMI 생성](#)
- [AWS ParallelClusterID 센터와의 UI 통합](#)

## 에서 첫 작업 실행 AWS ParallelCluster

이 튜토리얼에서는 Hello World에서 첫 작업을 실행하는 방법을 안내합니다. AWS ParallelCluster

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

## 사전 조건

- AWS ParallelCluster [설치되었습니다](#).
- AWS CLI [가 설치 및 구성되었습니다](#).
- [Amazon EC2 키](#) 페어가 있습니다.
- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.

## 설치 확인

먼저 Node.js 종속성을 포함하여 설치 및 구성이 AWS ParallelCluster 올바른지 확인합니다.

```
$ node --version
v16.8.0
$ pcluster version
{
  "version": "3.7.0"
}
```

그러면 실행 중인 버전이 반환됩니다 AWS ParallelCluster.

## 첫 번째 클러스터 생성

이제 첫 번째 클러스터를 생성할 시간입니다. 이 자습서의 워크로드는 성능 집약적이 아니기 때문에 t2.micro의 기본 인스턴스 크기를 사용합니다. (프로덕션 워크로드의 경우 더 적합한 인스턴스 크기를 선택해야 합니다.) 클러스터 hello-world를 호출해 보겠습니다.

```
$ pcluster create-cluster \
  --cluster-name hello-world \
  --cluster-configuration hello-world.yaml
```



**Note**

대부분의 `pcluster` 명령에 사용할 항목을 지정해야 합니다. AWS 리전 `AWS_DEFAULT_REGION` 환경 변수나 `~/.aws/config` 파일 [default] 섹션의 `region` 설정에 그것이 지정되지 않은 경우 `pcluster` 명령줄에 `--region` 파라미터를 제공해야 합니다.

구성에 대한 메시지가 출력되는 경우 다음을 실행하여 AWS ParallelCluster를 구성해야 합니다.

```
$ pcluster configure --config hello-world.yaml
```

[pcluster create-cluster](#) 명령이 제대로 실행되면 다음과 비슷한 출력이 표시됩니다.

```
{
  "cluster": {
    "clusterName": "hello-world",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:xxx:stack/xxx",
    "region": "...",
    "version": "...",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

다음을 사용하여 클러스터 생성을 모니터링합니다.

```
$ pcluster describe-cluster --cluster-name hello-world
```

`clusterStatus`는 클러스터를 생성하는 동안 “CREATE\_IN\_PROGRESS”를 보고합니다. 클러스터가 성공적으로 생성되면 `clusterStatus`이 “CREATE\_COMPLETE”로 전환됩니다. 출력은 헤드 노드의 `publicIpAddress` 및 `privateIpAddress`도 제공합니다.

## 헤드 노드에 로그인

OpenSSH pem 파일을 사용하여 헤드 노드에 로그인합니다.

```
$ pcluster ssh --cluster-name hello-world -i /path/to/keyfile.pem
```

로그인되면 `sinfo` 명령을 실행하여 컴퓨팅 노드가 설정되고 구성되어 있는지 확인합니다.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up       infinite   10    idle~ queue1-dy-queue1t2micro-[1-10]
```

출력은 클러스터에 최대 10개의 노드로 구성된 대기열이 하나 있음을 보여줍니다.

## Slurm을 사용하여 첫 번째 작업 실행

이제 잠시 동안 대기한 다음 고유의 호스트 이름을 출력하는 작업을 생성합니다. 다음 콘텐츠를 통해 `hellojob.sh`라는 파일을 생성합니다.

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

그런 다음 `sbatch`를 사용하여 작업을 제출하고 작업이 실행되는지 확인합니다.

```
$ sbatch hellojob.sh
Submitted batch job 2
```

이제 대기열을 보고 작업 상태를 확인할 수 있습니다. 새로운 Amazon EC2 인스턴스를 프로비저닝하는 작업은 백그라운드에서 시작됩니다. `sinfo` 명령을 사용하여 클러스터 인스턴스의 상태를 모니터링할 수 있습니다.

```
$ squeue
          JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           2     queue1 hellojob ec2-user CF       3:30      1 queue1-dy-
queue1t2micro-1
```

출력에는 작업이 `queue1`에 제출되었음을 보여줍니다. 작업이 완료될 때까지 30초 동안 기다린 후 `squeue`을 다시 실행합니다.

```
$ squeue
          JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

이제 대기열에 작업이 없으므로 현재 디렉터리에서 출력을 확인할 수 있습니다.

```
$ ls -l
```

```
total 8
-rw-rw-r-- 1 ec2-user ec2-user 57 Sep  1 14:25 hellojob.sh
-rw-rw-r-- 1 ec2-user ec2-user 43 Sep  1 14:30 slurm-2.out
```

출력에는 “out” 파일이 표시됩니다. 작업의 출력이 표시됩니다.

```
$ cat slurm-2.out
Hello World from queue1-dy-queue1t2micro-1
```

출력에서는 작업이 queue1-dy-queue1t2micro-1 인스턴스에서 성공적으로 실행되었음을 보여줍니다.

방금 생성한 클러스터에서는 클러스터의 모든 노드에서 홈 디렉터리만 공유됩니다.

클러스터 생성 및 사용에 대해 자세히 알아보려면 [모범 사례](#) 섹션을 참조하세요.

애플리케이션에 공유 소프트웨어, 라이브러리 또는 데이터가 필요한 경우 다음 옵션을 고려해 보세요.

- 에 설명된 대로 소프트웨어를 포함하는 AWS ParallelCluster 활성화된 사용자 지정 AMI를 구축하십시오. [사용자 지정 AWS ParallelCluster AMI 구축](#).
- AWS ParallelCluster 구성 파일의 [StorageSettings](#) 옵션을 사용하여 공유 파일 시스템을 지정하고 설치된 소프트웨어를 지정된 마운트 위치에 저장합니다.
- [사용자 지정 부트스트랩 작업](#)을 사용하여 클러스터 각 노드의 부트스트랩 절차를 자동화할 수 있습니다.

## 사용자 지정 AWS ParallelCluster AMI 구축

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

### Important

사용자 지정 AMI를 빌드한다면, 새로운 각 AWS ParallelCluster 릴리스에서 사용자 지정 AMI를 생성하는 데 사용된 단계를 반복해야 합니다.

자세한 내용을 읽기 전에 먼저 [사용자 지정 부트스트랩 작업](#) 섹션을 검토하는 것이 좋습니다. 수정하려는 내용이 스크립트로 작성되고 향후 AWS ParallelCluster 릴리스에서 지원할 수 있는지 확인하세요.

일반적으로 사용자 지정 AMI를 구축하는 것이 이상적이지는 않지만 사용자 지정 AMI를 구축해야 하는 특정 AWS ParallelCluster 시나리오가 있습니다. 이 자습서에서는 이러한 시나리오에 맞는 사용자 지정 AMI를 빌드하는 방법에 대해 설명합니다.

## 사전 조건

- AWS ParallelCluster [설치되었습니다](#).
- AWS CLI [가 설치 및 구성되었습니다](#).
- [Amazon EC2 키](#) 페어가 있습니다.
- [pcluster](#) CLI를 실행하고 이미지를 빌드하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.

## AWS ParallelCluster AMI를 사용자 지정하는 방법

사용자 지정 AWS ParallelCluster AMI를 구축하는 방법은 두 가지가 있습니다. 이 두 가지 방법 중 하나는 AWS ParallelCluster CLI를 사용하여 새 AMI를 구축하는 것입니다. 또 다른 방법을 사용하려면 수동으로 수정하여 AWS 계정에서 MI를 새로 빌드해야 합니다.

## 사용자 지정 AWS ParallelCluster AMI 구축

사용자 지정된 AMI와 소프트웨어가 있는 경우, 그 AWS ParallelCluster 위에 필요한 변경 사항을 적용할 수 있습니다. AWS ParallelCluster EC2 Image Builder 서비스를 사용하여 사용자 지정 AMI를 구축합니다. 자세한 내용은 [Image Builder 사용 설명서](#)를 참조하세요.

### 중요 사항:

- 이 프로세스에는 1시간 가량 소요됩니다. 이 시간은 빌드 시 추가로 [Build/Components](#)를 설치할 경우 달라질 수 있습니다.
- AMI에는 주요 구성 요소 버전으로 태그가 지정되어 있습니다. 여기에는 커널, 스케줄러, [EFA](#) 드라이버가 포함됩니다. 구성 요소 버전의 하위 집합도 AMI 설명에 보고됩니다.
- AWS ParallelCluster 3.0.0부터 새로운 CLI 명령 세트를 사용하여 이미지의 라이프사이클을 관리할 수 있습니다. 여기에 [build-image](#), [list-images](#), [describe-image](#) 및 [delete-image](#)도 추가되었습니다.
- 이 방법은 반복할 수 있습니다. 다시 실행하여 AMI를 최신 상태로 유지(예: OS 업데이트)한 다음 기존 클러스터를 업데이트할 때 사용할 수 있습니다.

**Note**

AWS 중국 파티션에서 이 방법을 사용하면 네트워크 오류가 발생할 수 있습니다. 예를 들어, OS 리포지토리에서 GitHub 또는 OS 리포지토리에서 패키지를 다운로드할 때 `pcluster build-image` 명령에서 이러한 오류가 표시될 수 있습니다. 이 오류가 발생하면 다음 대체 방법 중 하나를 사용하는 것이 좋습니다.

1. 이 명령을 우회하는 [AWS ParallelCluster AMI 수정](#) 방법을 따릅니다.
2. 이미지를 다른 파티션 및 리전(예: us-east-1)에 빌드한 다음 저장-복원하여 중국 리전으로 이동합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [S3를 사용하여 AMI 저장 및 복원](#)을 참조하십시오.

단계:

1. AWS ParallelCluster 클라이언트가 사용자를 대신하여 AWS API 작업을 호출할 수 있도록 AWS 계정 자격 증명을 구성하십시오. 필요한 권한 목록은 [AWS Identity and Access Management 권한 입력 AWS ParallelCluster](#) 섹션을 참조하세요.
2. 기본 이미지 빌드 구성 파일을 생성합니다. 이렇게 하려면 이미지 및 `ParentImage`를 빌드하는 데 사용할 `InstanceType`을 지정합니다. 이는 AMI를 생성하기 위한 시작점으로 사용됩니다. 선택적 빌드 파라미터에 대한 자세한 내용은 [이미지 구성](#)을 참조하세요.

Build:

```
InstanceType: <BUILD_INSTANCE_TYPE>
ParentImage: <BASE_AMI_ID>
```

3. CLI 명령을 `pcluster build-image` 사용하여 기본으로 제공하는 AWS ParallelCluster AMI에서 시작하여 AMI를 구축합니다.

```
$ pcluster build-image --image-id IMAGE_ID --image-configuration IMAGE_CONFIG.yaml --
region REGION
{
  "image": {
    "imageId": "IMAGE_ID",
    "imageBuildStatus": "BUILD_IN_PROGRESS",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
    "region": "us-east-1",
```

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

### Warning

`pcluster build-image`는 기본 VPC를 사용합니다. AWS Control Tower 또는 AWS 랜딩 존을 사용하여 기본 VPC를 삭제하는 경우 이미지 구성 파일에 서브넷 ID를 지정해야 합니다. 자세한 정보는 [SubnetId](#)을 참조하세요.

기타 파라미터 목록은 [pcluster build-image](#) 명령 참조 페이지를 참조하세요. 이전 명령의 결과는 다음과 같습니다.

- CloudFormation 스택은 이미지 구성을 기반으로 생성됩니다. 스택에는 빌드에 필요한 모든 EC2 Image Builder 리소스가 포함되어 있습니다.
  - 생성된 리소스에는 사용자 지정 Image Builder AWS ParallelCluster 구성 요소를 추가할 수 있는 공식 Image Builder 구성 요소가 포함됩니다. 사용자 지정 구성 요소를 만드는 방법을 알아보려면 공공 부문 고객 워크숍용 HPC의 [사용자 지정 AMI 예제](#)를 참조하세요.
  - EC2 Image Builder는 빌드 인스턴스를 시작하고 AWS ParallelCluster 쿡북을 적용하고 소프트웨어 스택을 설치하고 AWS ParallelCluster 필요한 구성 작업을 수행합니다. AWS ParallelCluster 쿡북은 빌드 및 부트스트랩에 사용됩니다. AWS ParallelCluster
  - 인스턴스가 중지되고 이 인스턴스에서 새 AMI가 생성됩니다.
  - 새로 생성한 AMI에서 다른 인스턴스가 시작됩니다. 테스트 단계에서 EC2 Image Builder는 Image Builder 구성 요소에 정의된 테스트를 실행합니다.
  - 빌드가 성공하면 스택이 삭제됩니다. 빌드가 실패하더라도 스택은 유지되며 검사할 수 있습니다.
4. 다음 명령을 실행하여 빌드 프로세스의 상태를 모니터링할 수 있습니다. 빌드가 완료되면 이를 실행하여 응답에 제공된 AMI ID를 검색할 수 있습니다.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION

# BEFORE COMPLETE
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/image-config.yaml?...",
  },
  "imageId": "IMAGE_ID",
```

```

"imagebuilderImageStatus": "BUILDING",
"imageBuildStatus": "BUILD_IN_PROGRESS",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
"region": "us-east-1",
"version": "3.7.0",
"cloudformationStackTags": [
  {
    "value": "3.7.0",
    "key": "parallelcluster:version"
  },
  {
    "value": "IMAGE_ID",
    "key": "parallelcluster:image_name"
  },
  ...
],
"imageBuildLogsArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-IMAGE_ID",
"cloudformationStackCreationTime": "2022-04-05T21:36:26.176Z"
}

# AFTER COMPLETE
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
  },
  "imageId": "IMAGE_ID",
  "imageBuildStatus": "BUILD_COMPLETE",
  "region": "us-east-1",
  "ec2AmiInfo": {
    "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
    "amiId": "ami-1234stuv5678wxyz",
    "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
    "state": "AVAILABLE",
    "tags": [
      {
        "value": "2021.3.11591-1.el7.x86_64",
        "key": "parallelcluster:dcv_version"
      }
    ]
  }
}

```

```

    },
    ...
  ],
  "architecture": "x86_64"
},
"version": "3.7.0"
}

```

5. 클러스터를 생성하려면 클러스터 구성 내의 [CustomAmi](#) 필드에 AMI ID를 입력합니다.

## AMI 생성 프로세스 문제 해결 및 모니터링

이미지 생성은 약 1시간 후에 완료됩니다. [pcluster describe-image](#) 명령 또는 로그 검색 명령을 실행하여 프로세스를 모니터링할 수 있습니다.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

이 [build-image](#) 명령은 이미지를 구축하는 데 필요한 모든 Amazon EC2 리소스가 포함된 CloudFormation 스택을 생성하고 EC2 Image Builder 프로세스를 시작합니다.

[build-image](#) 명령을 실행한 후에는 `awslogs`를 사용하여 CloudFormation 스택 이벤트를 검색할 수 있습니다. [pcluster get-image-stack-events](#) --query 파라미터로 결과를 필터링하여 최신 이벤트를 확인할 수 있습니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI 출력 필터링](#)을 참조하십시오.

```
$ pcluster get-image-stack-events --image-id IMAGE_ID --region REGION --query
"events[0]"
{
  "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
  "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-IMAGE_ID/3.7.0/1",
  "resourceStatus": "CREATE_IN_PROGRESS",
  "resourceStatusReason": "Resource creation Initiated",
  "resourceProperties": "{\"InfrastructureConfigurationArn\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-abcd1234-ef56-gh78-ij90-1234abcd5678\\\", \\\"ImageRecipeArn\\\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/parallelclusterimage-
IMAGE_ID/3.7.0\\\", \\\"DistributionConfigurationArn\\\": \\\"arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-abcd1234-ef56-
gh78-ij90-1234abcd5678\\\", \\\"Tags\\\": {\\\"parallelcluster:image_name\\\": \\\"IMAGE_ID\\\",
\\\"parallelcluster:image_id\\\": \\\"IMAGE_ID\\\"}}\",

```



```

"stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
"stackName": "IMAGE_ID",
"logicalResourceId": "ParallelClusterImage",
"resourceType": "AWS::ImageBuilder::Image",
"timestamp": "2022-04-05T21:39:24.725Z"
}

```

약 15분 후 Image Builder 생성과 관련된 로그 이벤트 항목에 스택 이벤트가 나타납니다. 이제 [pcluster list-image-log-streams](#) 및 [pcluster get-image-log-events](#) 명령을 사용하여 이미지 로그 스트림을 나열하고 Image Builder 단계를 모니터링할 수 있습니다.

```

$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
  --query 'logStreams[*].logStreamName'

"3.7.0/1"
]

$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
  --log-stream-name 3.7.0/1 --limit 3
{
  "nextToken": "f/36295977202298886557255241372854078762600452615936671762",
  "prevToken": "b/36295977196879805474012299949460899222346900769983430672",
  "events": [
    {
      "message": "ExecuteBash: FINISHED EXECUTION",
      "timestamp": "2022-04-05T22:13:26.633Z"
    },
    {
      "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/parallelclusterimage-test-abcd1234-ef56-gh78-ij90-1234abcd5678/3.7.0/1",
      "timestamp": "2022-04-05T22:13:26.741Z"
    },
    {
      "message": "TOE has completed execution successfully",
      "timestamp": "2022-04-05T22:13:26.819Z"
    }
  ]
}

```

BUILD\_COMPLETE 상태가 표시될 때까지 [describe-image](#) 명령을 사용하여 계속 확인하세요.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
{
  "imageConfiguration": {
    "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/image-config.yaml?Signature=..."
  },
  "imageId": "IMAGE_ID",
  "imageBuildStatus": "BUILD_COMPLETE",
  "region": "us-east-1",
  "ec2AmiInfo": {
    "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
    "amiId": "ami-1234stuv5678wxyz",
    "description": "AWS ParallelCluster AMI for alinux2, kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64, efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
    "state": "AVAILABLE",
    "tags": [
      {
        "value": "2021.3.11591-1.el7.x86_64",
        "key": "parallelcluster:dcv_version"
      },
      ...
    ],
    "architecture": "x86_64"
  },
  "version": "3.7.0"
}
```

사용자 지정 AMI 생성 문제를 해결해야 하는 경우 다음 단계에 설명된 대로 이미지 로그의 아카이브를 생성하세요.

--output 파라미터에 따라 Amazon S3 버킷 또는 로컬 파일에 로그를 보관할 수 있습니다.

```
$ pcluster export-image-logs --image-id IMAGE_ID --region REGION \
--bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER
{
  "url": "https://BUCKET_NAME.s3.us-east-1.amazonaws.com/BUCKET-FOLDER/IMAGE_ID-logs-202209071136.tar.gz?AWSAccessKeyId=..."
}

$ pcluster export-image-logs --image-id IMAGE_ID \
```

```
--region REGION --bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER --output-file /tmp/
archive.tar.gz
{
  "path": "/tmp/archive.tar.gz"
}
```

아카이브에는 Image Builder 프로세스 및 AWS CloudFormation 스택 이벤트와 관련된 CloudWatch 로그 스트림이 포함되어 있습니다. 이 명령을 실행하는 데 몇 분이 걸릴 수 있습니다.

## 사용자 지정 AMI 관리

AWS ParallelCluster 3.0.0부터 이미지 라이프사이클을 구축, 모니터링 및 관리하기 위한 새로운 명령 세트가 CLI에 추가되었습니다. 명령에 대한 자세한 내용은 [pcluster 명령](#)을 참조하세요.

## AWS ParallelCluster AMI 수정

이 방법은 공식 AMI 위에 사용자 지정을 추가하여 공식 AWS ParallelCluster AMI를 수정하는 것으로 구성됩니다. 기본 AWS ParallelCluster AMI는 새 릴리스로 업데이트됩니다. 이러한 AMI에는 설치 및 구성 시 작동하는 AWS ParallelCluster 데 필요한 모든 구성 요소가 있습니다. 이 중 하나를 기본으로 삼아 시작할 수 있습니다.

### 중요 사항:

- 이 방법은 [build-image](#) 명령보다 빠릅니다. 하지만 이 프로세스는 수동 프로세스이므로 자동으로 반복할 수 없습니다.
- 이 방법을 사용하면 CLI를 통해 사용할 수 있는 로그 검색 및 이미지 수명 주기 관리 명령에 액세스할 수 없습니다.

### 단계:

#### New Amazon EC2 console

- 사용 중인 AMI에 AWS 리전 해당하는 AMI를 찾으십시오. [pcluster list-official-images](#) 명령을 --region 파라미터와 함께 사용하여 원하는 AMI를 사용하려는 OS AWS 리전 --os 및 아키텍처에서 필터링할 특정 및 --architecture 파라미터를 선택합니다. 출력에서 Amazon EC2 이미지 ID를 검색합니다.
- AWS Management Console [로그인하고 https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/) 에서 [Amazon EC2 콘솔을 엽니다.](#)
- 탐색 창에서 이미지와 AMI를 선택합니다. 검색된 EC2 이미지 ID를 검색하고 AMI를 선택한 다음 AMI에서 인스턴스 시작을 선택합니다.

4. 아래로 스크롤하여 인스턴스 유형을 선택합니다.
5. 키 페어를 선택하고 인스턴스를 시작합니다.
6. OS 사용자와 SSH 키를 사용하여 인스턴스에 로그인합니다.
7. 요구 사항에 맞추어 인스턴스를 수동으로 사용자 지정합니다.
8. 다음 명령을 실행하여 AMI 생성을 위해 인스턴스를 준비합니다.

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. 콘솔에서 인스턴스 상태와 인스턴스 종지를 선택합니다.

인스턴스를 중지하려면 인스턴스를 선택하고 인스턴스 상태, 인스턴스 종지를 차례로 선택합니다.

10. [Amazon EC2 콘솔을 사용하여 인스턴스에서 새 AMI를 생성하거나 AWS CLI 이미지를 생성합니다.](#)

Amazon EC2 콘솔에서

- a. 탐색 창에서 인스턴스를 선택합니다.
- b. 생성하고 수정한 인스턴스를 선택합니다.
- c. 작업, 이미지, 이미지 생성을 차례로 선택합니다.
- d. 이미지 생성을 선택합니다.

11. 클러스터를 생성하려면 클러스터 구성 내의 [CustomAmi](#) 필드에 새 AMI ID를 입력합니다.

## Old Amazon EC2 console

1. 사용 중인 AWS ParallelCluster AMI에 AWS 리전 해당하는 AMI를 찾으십시오. [pcluster list-official-images](#) 명령을 --region 파라미터와 함께 사용하여 원하는 AMI를 사용하는 OS AWS 리전 --os 및 아키텍처에서 필터링할 특정 및 --architecture 파라미터를 선택하면 됩니다. 출력에서 Amazon EC2 이미지 ID를 검색할 수 있습니다.
2. AWS Management Console [로그인하고 https://console.aws.amazon.com/ec2/ 에서 Amazon EC2 콘솔을 엽니다.](#)
3. 탐색 창에서 이미지와 AMI를 선택합니다. 필터를 퍼블릭 이미지로 설정하고 검색된 EC2 이미지 ID를 검색하고 AMI를 선택한 다음 시작을 선택합니다.
4. 인스턴스 유형을 선택하고 다음: 인스턴스 세부 정보 구성 또는 검토 및 시작을 선택하여 인스턴스를 시작합니다.
5. 시작을 선택하고 키 페어를 선택한 다음 인스턴스를 시작합니다.

6. OS 사용자와 SSH 키를 사용하여 인스턴스에 로그인합니다. 자세한 내용을 보려면 인스턴스로 이동하여 새 인스턴스를 선택한 다음 연결을 선택하세요.
7. 요구 사항에 맞추어 인스턴스를 수동으로 사용자 지정합니다.
8. 다음 명령을 실행하여 AMI 생성을 위해 인스턴스를 준비합니다.

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Amazon EC2 콘솔의 탐색 창에서 [Instances] 를 선택하고 새 인스턴스를 선택한 다음 [작업], [인스턴스 상태], [Stop] 을 선택합니다.
10. [Amazon EC2 콘솔을 사용하여 인스턴스에서 새 AMI를 생성하거나 AWS CLI 이미지를 생성합니다.](#)

Amazon EC2 콘솔에서

- a. 탐색 창에서 인스턴스를 선택합니다.
  - b. 생성하고 수정한 인스턴스를 선택합니다.
  - c. 작업, 이미지, 이미지 생성을 차례로 선택합니다.
  - d. 이미지 생성을 선택합니다.
11. 클러스터를 생성하려면 클러스터 구성 내의 [CustomAmi](#) 필드에 새 AMI ID를 입력합니다.

## Active Directory 통합

이 자습서에서는 다중 사용자 환경을 생성합니다. 이 환경에는 AWS Managed Microsoft AD (Active Directory) 와 통합된 AWS ParallelCluster 사이트가 포함되어 `corp.example.com` 있습니다. Admin 사용자는 디렉터리를 관리하고, ReadOnlly 사용자는 디렉터리를 읽고, user000 사용자는 클러스터에 로그인하도록 구성합니다. 자동 경로 또는 수동 경로를 사용하여 AD를 구성하는 데 사용하는 네트워킹 리소스, Active Directory (AD) 및 Amazon EC2 인스턴스를 생성할 수 있습니다. 경로에 관계없이 생성하는 인프라는 다음 방법 중 하나를 AWS ParallelCluster 사용하여 통합되도록 미리 구성됩니다.

- 인증서 검증이 포함된 LDAPS(가장 안전한 옵션으로 권장)
- 인증서 검증이 없는 LDAPS
- LDAP

LDAP 자체로는 암호화를 제공하지 않습니다. 잠재적으로 민감한 정보를 안전하게 전송하려면 AD와 통합된 클러스터에는 LDAPS(LDAP over TLS/SSL)를 사용하는 것이 좋습니다. 자세한 내용은 관리 가이드의 [서버측 LDAPS 사용](#)을 참조하십시오 AWS Managed Microsoft AD. AWS Directory Service

이러한 리소스를 만든 후에는 Active Directory(AD)와 통합된 클러스터를 구성하고 생성하세요. 클러스터가 생성된 후에는 생성한 사용자로 로그인합니다. 이 자습서에서 생성하는 구성에 대한 자세한 내용은 [클러스터에 대한 다중 사용자 액세스](#) 및 [DirectoryService](#) 구성 섹션을 참조하세요.

이 자습서에서는 클러스터에 대한 다중 사용자 액세스를 지원하는 환경을 만드는 방법을 다룹니다. 이 자습서에서는 AD를 만들고 사용하는 방법을 다루지 않습니다. AWS Directory Service 이 AWS Managed Microsoft AD 튜토리얼에서 설정하는 단계는 테스트 목적으로만 제공됩니다. AWS Directory Service 관리 안내서의 [AWS Managed Microsoft AD](#) 및 [Simple AD](#)에서 찾을 수 있는 공식 설명서 및 모범 사례를 대체하기 위해 제공되는 것은 아닙니다.

### Note

디렉터리 사용자 암호는 디렉터리 암호 정책 속성 정의에 따라 만료됩니다. 자세한 내용은 [지원되는 정책 설정](#)을 참조하세요. 를 사용하여 디렉터리 암호를 AWS ParallelCluster재설정하려면 [사용자 암호 및 만료된 암호를 재설정하는 방법](#)을 참조하십시오.

### Note

디렉터리 도메인 컨트롤러 IP 주소는 도메인 컨트롤러 변경 및 디렉터리 유지 보수로 인해 변경될 수 있습니다. 디렉터리 인프라를 생성하기 위해 자동화된 빠른 생성 방법을 선택한 경우 디렉터리 IP 주소가 변경될 때 디렉터리 컨트롤러 앞에 로드 밸런서를 수동으로 정렬해야 합니다. 빠른 생성 메서드를 사용하는 경우 디렉터리 IP 주소가 로드 밸런서와 자동으로 정렬되지 않습니다.

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

### 사전 조건

- AWS ParallelCluster [설치되었습니다](#).
- AWS CLI [가 설치 및 구성되었습니다](#).
- [Amazon EC2 키](#) 페어가 있습니다.

- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.

자습서를 진행하면서 *region-id* 및 *d-abcdef01234567890*과 같은 *inputs highlighted in red*를 자신의 이름과 ID로 바꾸세요. *0123456789012* AWS 계정 번호로 바꾸십시오.

## 1단계: AD 인프라 생성

자동 탭을 선택하여 AWS CloudFormation 빠른 생성 템플릿으로 Active Directory (AD) 인프라를 만들 수 있습니다.

수동 탭을 선택하여 AD 인프라를 수동으로 생성합니다.

### 자동

1. AWS Management Console에 로그인합니다.
2. [CloudFormation 빠른 생성 \(지역 us-east-1\)](#) 을 열어 콘솔에서 다음 리소스를 생성합니다.

#### CloudFormation

- VPC가 지정되지 않은 경우 서브넷 2개와 퍼블릭 액세스를 위한 라우팅이 있는 VPC.
- AWS Managed Microsoft AD An.
- 디렉터리를 관리하는 데 사용할 수 있는 AD에 연결된 Amazon EC2 인스턴스입니다.

3. 빠른 스택 생성 페이지 파라미터 섹션에서 다음 파라미터의 암호를 입력합니다.

- AdminPassword
- ReadOnlyPassword
- UserPassword

암호를 기록해 둡니다. 이 자습서 뒷부분에서 이 정보가 필요합니다.

4. DomainName에 **corp.example.com**를 입력합니다.
5. 키페어에 Amazon EC2 키 페어의 이름을 입력합니다.
6. 페이지 하단에서 각 액세스 기능이 필요함을 확인하는 확인란을 선택합니다.
7. 스택 생성을 선택합니다.
8. CloudFormation 스택이 CREATE\_COMPLETE 상태에 도달하면 스택의 [Outputs] 탭을 선택합니다. 출력 리소스 이름과 ID는 이후 단계에서 사용해야 하므로 기록해 두세요. 출력은 클러스터를 생성하는 데 필요한 정보를 제공합니다.

CloudFormation > Stacks > PclusterAD-1234abc

**Stacks (3)**

Filter by stack name

Active View nested

PclusterAD  
2022-05-18 10:27:24 UTC-0700  
CREATE\_COMPLETE

**PclusterAD-abcd123**

Delete Update Stack actions Create stack

Stack info Events Resources **Outputs** Parameters Template Change sets

**Outputs (10)**

Search outputs

| Key                               | Value                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------|
| DomainAddrLdap                    | ldap://10.0.111.88,ldap://10.0.222.111                                         |
| DomainAddrLdaps                   | ldaps://corp.example.com                                                       |
| DomainCertificateArn              | arn:aws:acm:us-east-1:123456789012:certificate/1234abcd-ef56-78gh-ij90-abcd1   |
| DomainCertificateSecretArn        | arn:aws:secretsmanager:us-east-1:123456789012:secret:DomainCertificateSecret-f |
| DomainCertificateSecretReadPolicy | arn:aws:iam::123456789012:policy/DomainCertificateSecretReadPolicy-PclusterAD  |
| DomainName                        | corp.example.com                                                               |
| DomainReadOnlyUser                | cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com                     |
| PasswordSecretArn                 | arn:aws:secretsmanager:us-east-1:123456789012:secret>PasswordSecret-PclusterA  |
| PrivateSubnetIds                  | subnet-1234567890abcdef0,subnet-abcdef01234567890                              |
| VpcId                             | vpc-021345abcdef6789                                                           |

- (선택 사항)2단계: [AD 사용자 및 그룹 관리](#) 연습을 완료하려면 디렉터리 ID가 필요합니다. 리소스를 선택하고 아래로 스크롤하여 디렉터리 ID를 기록해 둡니다.
- (선택 사항)2단계: [AD 사용자 및 그룹 관리](#) 또는 [3단계: 클러스터 생성](#)에서 계속하세요.

## 수동

다양한 가용 영역 및 AWS Managed Microsoft AD에 있는 두 개의 서브넷으로 디렉터리 서비스용 VPC를 생성합니다.

## AD 생성

### Note

- 디렉터리 및 도메인 이름은 `corp.example.com`입니다. 짧은 이름은 `CORP`입니다.
- 스크립트에서 Admin 암호를 변경합니다.
- Active Directory(AD)를 생성하는 데 최소 15분이 걸립니다.



다음 Python 스크립트를 사용하여 로컬에 VPC, 서브넷, AD 리소스를 생성합니다. AWS 리전이 파일을 `ad.py`로 저장하고 실행하세요.

```
import boto3
import time
from pprint import pprint

vpc_name = "PclusterVPC"
ad_domain = "corp.example.com"
admin_password = "asdfASDF1234"

Amazon EC2 = boto3.client("ec2")
ds = boto3.client("ds")
region = boto3.Session().region_name

# Create the VPC, Subnets, IGW, Routes
vpc = ec2.create_vpc(CidrBlock="10.0.0.0/16")["Vpc"]
vpc_id = vpc["VpcId"]
time.sleep(30)
ec2.create_tags(Resources=[vpc_id], Tags=[{"Key": "Name", "Value": vpc_name}])
subnet1 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.0.0/17",
    AvailabilityZone=f"{region}a")["Subnet"]
subnet1_id = subnet1["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet1_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/
subnet1"}])
ec2.modify_subnet_attribute(SubnetId=subnet1_id, MapPublicIpOnLaunch={"Value": True})
subnet2 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.128.0/17",
    AvailabilityZone=f"{region}b")["Subnet"]
subnet2_id = subnet2["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet2_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/
subnet2"}])
ec2.modify_subnet_attribute(SubnetId=subnet2_id, MapPublicIpOnLaunch={"Value": True})
igw = ec2.create_internet_gateway()["InternetGateway"]
ec2.attach_internet_gateway(InternetGatewayId=igw["InternetGatewayId"], VpcId=vpc_id)
route_table = ec2.describe_route_tables(Filters=[{"Name": "vpc-id", "Values":
    [vpc_id]}])["RouteTables"][0]
ec2.create_route(RouteTableId=route_table["RouteTableId"],
    DestinationCidrBlock="0.0.0.0/0", GatewayId=igw["InternetGatewayId"])
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsSupport={"Value": True})
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsHostnames={"Value": True})
```

```

# Create the Active Directory
ad = ds.create_microsoft_ad(
    Name=ad_domain,
    Password=admin_password,
    Description="ParallelCluster AD",
    VpcSettings={"VpcId": vpc_id, "SubnetIds": [subnet1_id, subnet2_id]},
    Edition="Standard",
)
directory_id = ad["DirectoryId"]

# Wait for completion
print("Waiting for the directory to be created...")
directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
directory = directories[0]
while directory["Stage"] in {"Requested", "Creating"}:
    time.sleep(3)
    directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
    directory = directories[0]

dns_ip_addrs = directory["DnsIpAddrs"]

pprint({"directory_id": directory_id,
        "vpc_id": vpc_id,
        "subnet1_id": subnet1_id,
        "subnet2_id": subnet2_id,
        "dns_ip_addrs": dns_ip_addrs})

```

다음은 Python 스크립트의 출력 예제입니다.

```

{
  "directory_id": "d-abcdef01234567890",
  "dns_ip_addrs": ["192.0.2.254", "203.0.113.237"],
  "subnet1_id": "subnet-021345abcdef6789",
  "subnet2_id": "subnet-1234567890abcdef0",
  "vpc_id": "vpc-021345abcdef6789"
}

```

출력 리소스 이름 및 ID를 적어 두세요. 이후 단계에서 사용하게 됩니다.

스크립트가 완료되면 다음 단계를 계속합니다.

## Amazon EC2 인스턴스 생성

### New Amazon EC2 console

1. AWS Management Console에 로그인합니다.
2. 첨부된 4단계에 나열된 정책에 해당하는 역할이 없는 경우 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 여세요. 그렇지 않으면 5단계로 건너뛰세요.
3. ResetUserPassword 정책을 생성하여 빨간색으로 강조 표시된 콘텐츠를 AWS 리전 ID, 계정 ID, AD 생성 시 실행한 스크립트 출력의 디렉터리 ID로 대체합니다.

#### ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. 다음과 같은 정책을 연결하여 IAM 역할을 생성합니다.
  - AWS 관리형 정책: [AmazonSSM ManagedInstanceCore](#)
  - AWS 관리형 정책: [아마존/SSM DirectoryServiceAccess](#)
  - ResetUserPassword 정책
5. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
6. Amazon EC2 대시보드에서 인스턴스 시작을 선택합니다.
7. 애플리케이션 및 OS 이미지에서 최신 Amazon Linux 2 AMI를 선택합니다.
8. 인스턴스 유형에서 t2.micro를 선택합니다.
9. 키 페어(로그인)에서 키 페어를 선택합니다.
10. 네트워크 설정에서 편집을 선택합니다.
11. VPC에서 디렉터리 VPC를 선택합니다.
12. 아래로 스크롤하여 고급 세부 정보를 선택합니다.

13. 고급 세부 정보의 도메인 가입 디렉터리에서 **corp.example.com**을 선택합니다.
14. IAM 인스턴스 프로파일에서 1단계에서 생성한 역할 또는 4단계에서 나열된 정책이 연결된 역할을 선택합니다.
15. 요약에서 인스턴스 시작을 선택합니다.
16. 인스턴스 ID(예: i-1234567890abcdef0)를 기록하고 인스턴스 시작이 완료될 때까지 기다립니다.
17. 인스턴스가 시작된 후 이어서 다음 단계를 수행합니다.

## Old Amazon EC2 console

1. AWS Management Console에 로그인합니다.
2. 첨부된 4단계에 나열된 정책에 해당하는 역할이 없는 경우 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 여세요. 그렇지 않으면 5단계로 건너뛰세요.
3. ResetUserPassword 정책을 생성합니다. 빨간색으로 강조 표시된 콘텐츠를 Active Directory (AD) 를 생성하기 위해 실행한 스크립트 출력의 AWS 리전 AWS 계정 ID, ID 및 디렉터리 ID로 바꾸십시오.

### ResetUserPassword

```
{
  "Statement": [
    {
      "Action": [
        "ds:ResetUserPassword"
      ],
      "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
      "Effect": "Allow"
    }
  ]
}
```

4. 다음과 같은 정책을 연결하여 IAM 역할을 생성합니다.
  - AWS 관리형 정책: [AmazonSSM ManagedInstanceCore](#)
  - AWS [관리형 정책: 아마존/SSM DirectoryServiceAccess](#)
  - ResetUserPassword 정책
5. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.

6. Amazon EC2 대시보드에서 인스턴스 시작을 선택합니다.
7. 애플리케이션 및 OS 이미지에서 최신 Amazon Linux 2 AMI를 선택합니다.
8. 인스턴스 유형에서 t2.micro를 선택합니다.
9. 키 페어(로그인)에서 키 페어를 선택합니다.
10. 네트워크 설정에서 편집을 선택합니다.
11. 네트워크 설정, VPC에서 디렉터리 VPC를 선택합니다.
12. 아래로 스크롤하여 고급 세부 정보를 선택합니다.
13. 고급 세부 정보의 도메인 가입 디렉터리에서 **corp.example.com**을 선택합니다.
14. 고급 세부 정보, 인스턴스 프로파일에서 1단계에서 생성한 역할 또는 4단계에서 나열한 정책이 연결된 역할을 선택합니다.
15. 요약에서 인스턴스 시작을 선택합니다.
16. 인스턴스 ID(예:i-1234567890abcdef0) 를 메모하고 인스턴스 시작이 완료될 때까지 기다립니다.
17. 인스턴스가 시작된 후 이어서 다음 단계를 수행합니다.

인스턴스를 AD에 조인합니다.

1. 인스턴스에 연결하고 **admin**로서 AD Realm에 조인하세요.

인스턴스에 연결하고 다음 명령을 실행합니다.

```
$ INSTANCE_ID="i-1234567890abcdef0"
```

```
$ PUBLIC_IP=$(aws ec2 describe-instances \
--instance-ids $INSTANCE_ID \
--query "Reservations[0].Instances[0].PublicIpAddress" \
--output text)
```

```
$ ssh -i ~/.ssh/keys/keypair.pem ec2-user@$PUBLIC_IP
```

2. 필요한 소프트웨어를 설치하고 Realm에 조인하세요.

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-common-tools krb5-workstation openldap-clients policycoreutils-python
```

### 3. 관리자 암호를 **admin** 암호로 바꿉니다.

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com
Password for Admin:
```

위의 방법이 성공하면 Realm에 조인되어 다음 단계를 진행할 수 있습니다.

## AD에 사용자 추가

### 1. ReadOnlyUser 및 추가 사용자를 생성합니다.

이 단계에서는 이전 단계에서 설치한 [adcli](#) 및 [openldap-client](#) 도구를 사용합니다.

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=ReadOnlyUser ReadOnlyUser
```

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=user000 user000
```

### 2. 사용자가 생성되었는지 확인합니다.

디렉터리 DNS IP 주소는 Python 스크립트의 출력입니다.

```
$ DIRECTORY_IP="192.0.2.254"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=user000,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

기본적으로 `ad-cli`를 사용하여 사용자를 생성하면 해당 사용자는 비활성화됩니다.

### 3. 로컬 시스템에서 사용자 암호 재설정 및 활성화:

Amazon EC2 인스턴스에서 로그아웃합니다.

**Note**

- `ro-p@ssw0rd`에서 `ReadOnlyUser` 검색한 비밀번호입니다. AWS Secrets Manager
- `user-p@ssw0rd`는 클러스터에 연결(ssh)할 때 제공되는 클러스터 사용자의 암호입니다.

`directory-id`는 Python 스크립트의 출력입니다.

```
$ DIRECTORY_ID="d-abcdef01234567890"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "ReadOnlyUser" \
--new-password "ro-p@ssw0rd" \
--region "region-id"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "user000" \
--new-password "user-p@ssw0rd" \
--region "region-id"
```

#### 4. Secrets Manager 보안 암호에 암호를 추가합니다.

암호를 `ReadOnlyUser` 만들고 설정했으니 이제 로그인을 검증하는 데 AWS ParallelCluster 사용되는 암호에 저장하십시오.

Secrets Manager를 사용하여 `ReadOnlyUser`의 암호를 값으로 보관할 새 보안 암호를 생성합니다. 보안 암호 값 형식은 JSON 형식이 아닌 일반 텍스트만 사용해야 합니다. 향후 단계를 위해 보안 암호 ARN을 기록해 두세요.

```
$ aws secretsmanager create-secret --name "ADSecretPassword" \
--region region_id \
--secret-string "ro-p@ssw0rd" \
--query ARN \
--output text
arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
```

## 인증서 검증(권장)이 포함된 LDAPS 설정

리소스 ID를 기록해 두세요. 이후 단계에서 사용하게 됩니다.

1. 로컬에서 도메인 인증서를 생성합니다.

```
$ PRIVATE_KEY="corp-example-com.key"
CERTIFICATE="corp-example-com.crt"
printf ".\n.\n.\n.\n.\n.\ncorp.example.com\n.\n.\n" | openssl req -x509 -sha256 -nodes -
newkey rsa:2048 -keyout $PRIVATE_KEY -days 365 -out $CERTIFICATE
```

2. 인증서를 Secrets Manager에 저장하여 나중에 클러스터 내에서 검색할 수 있도록 합니다.

```
$ aws secretsmanager create-secret --name example-cert \
  --secret-string file://$CERTIFICATE \
  --region region-id
{
  "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
  "Name": "example-cert",
  "VersionId": "14866070-092a-4d5a-bcdd-9219d0566b9c"
}
```

3. Amazon EC2 인스턴스를 AD 도메인에 조인하기 위해 생성한 IAM 역할에 다음 정책을 추가합니
다.

### PutDomainCertificateSecrets

```
{
  "Statement": [
    {
      "Action": [
        "secretsmanager:PutSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
      ],
      "Effect": "Allow"
    }
  ]
}
```



#### 4. 인증서를 AWS Certificate Manager (ACM) 로 가져옵니다.

```
$ aws acm import-certificate --certificate fileb://$CERTIFICATE \
  --private-key fileb://$PRIVATE_KEY \
  --region region-id
{
  "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
}
```

#### 5. Active Directory 엔드포인트 앞에 배치되는 로드 밸런서를 생성합니다.

```
$ aws elbv2 create-load-balancer --name CorpExampleCom-NLB \
  --type network \
  --scheme internal \
  --subnets subnet-1234567890abcdef0 subnet-021345abcdef6789 \
  --region region-id
{
  "LoadBalancers": [
    {
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
      "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
      "CanonicalHostedZoneId": "Z2IF0LAFXWL04F",
      "CreatedTime": "2022-05-05T12:56:55.988000+00:00",
      "LoadBalancerName": "CorpExampleCom-NLB",
      "Scheme": "internal",
      "VpcId": "vpc-021345abcdef6789",
      "State": {
        "Code": "provisioning"
      },
      "Type": "network",
      "AvailabilityZones": [
        {
          "ZoneName": "region-idb",
          "SubnetId": "subnet-021345abcdef6789",
          "LoadBalancerAddresses": []
        },
        {
          "ZoneName": "region-ida",
          "SubnetId": "subnet-1234567890abcdef0",
          "LoadBalancerAddresses": []
        }
      ]
    }
  ]
}
```

```

    ],
    "IpAddressType": "ipv4"
  }
]
}

```

## 6. Active Directory 엔드포인트를 대상으로 하는 대상 그룹을 만드세요.

```

$ aws elbv2 create-target-group --name CorpExampleCom-Targets --protocol TCP \
  --port 389 \
  --target-type ip \
  --vpc-id vpc-021345abcdef6789 \
  --region region-id
{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
      "TargetGroupName": "CorpExampleCom-Targets",
      "Protocol": "TCP",
      "Port": 389,
      "VpcId": "vpc-021345abcdef6789",
      "HealthCheckProtocol": "TCP",
      "HealthCheckPort": "traffic-port",
      "HealthCheckEnabled": true,
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 10,
      "HealthyThresholdCount": 3,
      "UnhealthyThresholdCount": 3,
      "TargetType": "ip",
      "IpAddressType": "ipv4"
    }
  ]
}

```

## 7. Active Directory(AD) 엔드포인트를 대상 그룹에 등록합니다.

```

$ aws elbv2 register-targets --target-group-arn
arn:aws:elasticloadbalancing:region-id:123456789012:targetgroup/CorpExampleCom-
Targets/44577c583b695e81 \
  --targets Id=192.0.2.254,Port=389 Id=203.0.113.237,Port=389 \
  --region region-id

```

## 8. 인증서를 사용하여 LB 리스너를 생성합니다.

```

$ aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region-id:123456789012:loadbalancer/net/
CorpExampleCom-NLB/3afe296bf4ba80d4 \
  --protocol TLS \
  --port 636 \
  --default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 \
  --ssl-policy ELBSecurityPolicy-TLS-1-2-2017-01 \
  --certificates CertificateArn=arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 \
  --region region-id
"Listeners": [
  {
    "ListenerArn": "arn:aws:elasticloadbalancing:region-id:123456789012:listener/
net/CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b",
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
    "Port": 636,
    "Protocol": "TLS",
    "Certificates": [
      {
        "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
      }
    ],
    "SslPolicy": "ELBSecurityPolicy-TLS-1-2-2017-01",
    "DefaultActions": [
      {
        "Type": "forward",
        "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
        "ForwardConfig": {
          "TargetGroups": [
            {
              "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81"
            }
          ]
        }
      }
    ]
  }
]

```

```

    ]
  }
]
}

```

9. 클러스터 VPC 내에서 도메인을 검색할 수 있도록 호스팅 영역을 생성합니다.

```

$ aws route53 create-hosted-zone --name corp.example.com \
  --vpc VPCRegion=region-id,VPCId=vpc-021345abcdef6789 \
  --caller-reference "ParallelCluster AD Tutorial"
{
  "Location": "https://route53.amazonaws.com/2013-04-01/hostedzone/
Z09020002B5MZQNXMSJUB",
  "HostedZone": {
    "Id": "/hostedzone/Z09020002B5MZQNXMSJUB",
    "Name": "corp.example.com.",
    "CallerReference": "ParallelCluster AD Tutorial",
    "Config": {
      "PrivateZone": true
    },
    "ResourceRecordSetCount": 2
  },
  "ChangeInfo": {
    "Id": "/change/C05533343BF3IKSORW1TQ",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T13:21:53.863000+00:00"
  },
  "VPC": {
    "VPCRegion": "region-id",
    "VPCId": "vpc-021345abcdef6789"
  }
}

```

10. 다음 콘텐츠가 포함된 **recordset-change.json**라는 이름의 파일을 추가합니다.  
**HostedZoneId**는 로드 밸런서의 표준 호스팅 영역 ID입니다.

```

{
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "corp.example.com",
        "Type": "A",

```

```

    "Region": "region-id",
    "SetIdentifier": "example-active-directory",
    "AliasTarget": {
      "HostedZoneId": "Z2IF0LAFXWL04F",
      "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
      "EvaluateTargetHealth": true
    }
  }
}
]
}

```

11. 이번에는 호스팅 영역 ID를 사용하여 레코드세트 변경 내용을 호스팅 영역에 제출합니다.

```

$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNMSJUB \
--change-batch file://recordset-change.json
{
  "ChangeInfo": {
    "Id": "/change/C0137926I56R3GC7XW2Y",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
  }
}

```

12. 다음 콘텐츠가 포함된 정책 문서 **policy.json**을 생성합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-abc123"
      ],
      "Effect": "Allow"
    }
  ]
}

```

13. 다음 콘텐츠가 포함된 **policy.json**라는 정책 문서를 생성합니다.

```
$ aws iam create-policy --policy-name ReadCertExample \
  --policy-document file://policy.json
{
  "Policy": {
    "PolicyName": "ReadCertExample",
    "PolicyId": "ANPAUUXUVBC42VZSI4LDY",
    "Arn": "arn:aws:iam::123456789012:policy/ReadCertExample-efg456",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2022-05-05T13:42:18+00:00",
    "UpdateDate": "2022-05-05T13:42:18+00:00"
  }
}
```

14. [\(선택 사항\)2단계: AD 사용자 및 그룹 관리](#) 또는 [3단계: 클러스터 생성](#)의 단계를 계속 따르세요.

### (선택 사항)2단계: AD 사용자 및 그룹 관리

이 단계에서는 액티브 디리버리 (AD) 도메인에 연결된 Amazon EC2 Amazon Linux 2 인스턴스에서 사용자와 그룹을 관리합니다.

자동 경로를 따랐다면 자동화의 일부로 생성된 AD 조인 인스턴스를 다시 시작하고 로그인하세요.

수동 경로를 따랐다면 이전 단계에서 만들고 AD에 연결한 인스턴스를 다시 시작하고 로그인하세요.

이 단계에서는 이전 단계의 일부로 인스턴스에 설치된 [adcli](#) 및 [openldap-client](#) 도구를 사용합니다.

AD 도메인에 연결된 Amazon EC2 인스턴스에 로그인합니다.

1. Amazon EC2 콘솔에서 이전 단계에서 생성한 제목 없는 Amazon EC2 인스턴스를 선택합니다. 인스턴스가 중지된 상태여야 합니다.
2. 인스턴스 상태가 중지된 경우 [인스턴스 상태] 를 선택한 다음 [인스턴스 시작] 을 선택합니다.
3. 상태 확인이 통과되면 인스턴스를 선택하고 연결 및 SSH를 선택하고 인스턴스에 로그인합니다.

AD에 가입한 Amazon EC2 Amazon Linux 2 인스턴스에 로그인한 사용자 및 그룹을 관리합니다.

-U "Admin" 옵션을 사용하여 `adcli` 명령을 실행하면 AD Admin 암호를 입력하라는 메시지가 표시됩니다. AD Admin 암호를 `ldapsearch` 명령의 일부로 포함합니다.

1. 사용자를 생성합니다.

```
$ adcli create-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

2. 사용자 암호를 재설정합니다.

```
$ aws --region "region-id" ds reset-user-password --directory-id "d-abcdef01234567890" --user-name "clusteruser" --new-password "new-p@ssw0rd"
```

3. 그룹을 생성합니다.

```
$ adcli create-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

4. 그룹에 사용자를 추가합니다.

```
$ adcli add-member "clusterteam" "clusteruser" --domain "corp.example.com" -U "Admin"
```

5. 사용자 및 그룹을 설명합니다.

모든 사용자를 설명합니다.

```
$ ldapsearch "(&(objectClass=user))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

특정 사용자를 설명합니다.

```
$ ldapsearch "(&(objectClass=user)(cn=clusteruser))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

이름 패턴으로 모든 사용자를 설명합니다.

```
$ ldapsearch "(&(objectClass=user)(cn=user*))" -x -h "192.0.2.254" -b
"DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

특정 그룹에 속한 모든 사용자를 설명합니다.

```
$ ldapsearch "(&(objectClass=user)
(memberOf=CN=clusterteam,OU=Users,OU=CORP,DC=corp,DC=example,DC=com))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

모든 그룹을 설명합니다.

```
$ ldapsearch "objectClass=group" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com"
-D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

특정 그룹을 설명합니다.

```
$ ldapsearch "(&(objectClass=group)(cn=clusterteam))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

6. 그룹에서 사용자를 제거합니다.

```
$ adcli remove-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

7. 사용자를 삭제합니다.

```
$ adcli delete-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

8. 그룹을 삭제합니다.

```
$ adcli delete-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

### 3단계: 클러스터 생성

Amazon EC2 인스턴스를 종료하지 않았다면 지금 종료하십시오.



환경은 Active Directory(AD)에 대해 사용자를 인증할 수 있는 클러스터를 생성하도록 설정되어 있습니다.

간단한 클러스터 구성을 만들고 AD 연결과 관련된 설정을 제공합니다. 자세한 내용은 [DirectoryService\(을\)](#)를 참조하세요.

다음 클러스터 구성 중 하나를 선택하고 `ldaps_config.yaml`, `ldaps_nocert_config.yaml` 또는 `ldap_config.yaml`이라는 이름의 파일에 복사합니다.

인증서 검증이 있는 LDAPS 구성을 선택하는 것이 좋습니다. 이 구성을 선택하는 경우 부트스트랩 스크립트도 `active-directory.head.post.sh`로 이름이 지정된 파일에 복사해야 합니다. 그리고 구성 파일에 표시된 대로 Amazon S3 버킷에 저장해야 합니다.

인증서 검증 구성이 포함된 LDAPS(권장)

#### Note

다음 구성 요소를 변경해야 합니다.

- `KeyName`: Amazon EC2 키페어 중 하나입니다.
- `SubnetId` / `SubnetIds`: CloudFormation 빠른 생성 스택 (자동 자습서) 또는 python 스크립트 (수동 자습서) 의 출력에 제공된 서브넷 ID 중 하나입니다.
- `Region`: AD 인프라를 생성한 리전입니다.
- `DomainAddr`: 이 IP 주소는 AD 서비스의 DNS 주소 중 하나입니다.
- `PasswordSecretArn`: `DomainReadOnlyUser`의 암호가 포함된 보안 암호의 Amazon 리소스 이름(ARN)입니다.
- `BucketName`: 부트스트랩 스크립트가 들어 있는 버킷의 이름입니다..
- `AdditionalPolicies/Policy`: 읽기 도메인 인증 정책의 Amazon 리소스 이름 (ARN).  
`ReadCertExample`
- `CustomActions/OnNodeConfigured/Args`: 도메인 인증 정책을 보유하는 보안 암호의 Amazon 리소스 이름(ARN).

보안 태세를 강화하려면 `HeadNode/Ssh/AllowedIps` 구성을 사용하여 헤드 노드에 대한 SSH 액세스를 제한하는 것이 좋습니다.

Region: *region-id*

```
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: keypair
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::123456789012:policy/ReadCertExample
    S3Access:
      - BucketName: my-bucket
        EnableWriteAccess: false
        KeyName: bootstrap/active-directory/active-directory.head.post.sh
  CustomActions:
    OnNodeConfigured:
      Script: s3://my-bucket/bootstrap/active-directory/active-directory.head.post.sh
      Args:
        - arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc
        - /opt/parallelcluster/shared/directory_service/domain-certificate.crt
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue0
        ComputeResources:
          - Name: queue0-t2-micro
            InstanceType: t2.micro
            MinCount: 1
            MaxCount: 10
        Networking:
          SubnetIds:
            - subnet-abcdef01234567890
  DirectoryService:
    DomainName: corp.example.com
    DomainAddr: ldaps://corp.example.com
    PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
    DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
    LdapTlsCaCert: /opt/parallelcluster/shared/directory_service/domain-certificate.crt
    LdapTlsReqCert: hard
```

## 부트스트랩 스크립트

부트스트랩 파일을 생성한 후 S3 버킷에 업로드하기 전에 `chmod +x active-directory.head.post.sh` 실행하여 실행 권한을 AWS ParallelCluster 부여하십시오.

```
#!/bin/bash
set -e

CERTIFICATE_SECRET_ARN="$1"
CERTIFICATE_PATH="$2"

[[ -z $CERTIFICATE_SECRET_ARN ]] && echo "[ERROR] Missing CERTIFICATE_SECRET_ARN" &&
exit 1
[[ -z $CERTIFICATE_PATH ]] && echo "[ERROR] Missing CERTIFICATE_PATH" && exit 1

source /etc/parallelcluster/cfnconfig
REGION="${cfn_region:?}"

mkdir -p $(dirname $CERTIFICATE_PATH)
aws secretsmanager get-secret-value --region $REGION --secret-id
$CERTIFICATE_SECRET_ARN --query SecretString --output text > $CERTIFICATE_PATH
```

## 인증서 확인 구성이 없는 LDAPS

### Note

다음 구성 요소를 변경해야 합니다.

- KeyName: Amazon EC2 키페어 중 하나입니다.
- SubnetId / SubnetIds: CloudFormation 빠른 생성 스택 (자동 자습서) 또는 python 스크립트 (수동 자습서) 의 출력에 있는 서브넷 ID 중 하나입니다.
- Region: AD 인프라를 생성한 리전입니다.
- DomainAddr: 이 IP 주소는 AD 서비스의 DNS 주소 중 하나입니다.
- PasswordSecretArn: DomainReadOnlyUser의 암호가 포함된 보안 암호의 Amazon 리소스 이름(ARN)입니다.

더 나은 보안 태세를 위해 HeadNode /Ssh/ AllowedIps 구성을 사용하여 헤드 노드에 대한 SSH 액세스를 제한하는 것이 좋습니다.

```

Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: keypair
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
      ComputeResources:
        - Name: queue0-t2-micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: corp.example.com
  DomainAddr: ldaps://corp.example.com
  PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsReqCert: never

```

## LDAP 구성

### Note

다음 구성 요소를 변경해야 합니다.

- KeyName: Amazon EC2 키페어 중 하나입니다.
- SubnetId / SubnetIds: CloudFormation 빠른 생성 스택 (자동 자습서) 또는 python 스크립트 (수동 자습서) 의 출력에 제공된 서브넷 ID 중 하나입니다.
- Region: AD 인프라를 생성한 리전입니다.
- DomainAddr: 이 IP 주소는 AD 서비스의 DNS 주소 중 하나입니다.

- PasswordSecretArn: DomainReadOnlyUser의 암호가 포함된 보안 암호의 Amazon 리소스 이름(ARN)입니다.

더 나은 보안 태세를 위해 HeadNode /Ssh/ AllowedIps 구성을 사용하여 헤드 노드에 대한 SSH 액세스를 제한하는 것이 좋습니다.

```

Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: keypair
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue0
      ComputeResources:
        - Name: queue0-t2-micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://192.0.2.254,ldap://203.0.113.237
  PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True

```

다음 명령을 사용하여 클러스터를 생성합니다.

```

$ pcluster create-cluster --cluster-name "ad-cluster" --cluster-configuration "./ldaps_config.yaml"

```

```
{
  "cluster": {
    "clusterName": "pcluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
    "region": "region-id",
    "version": 3.7.0,
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

#### 4단계: 사용자로서 클러스터에 연결

다음 명령을 사용하여 클러스터의 상태를 확인할 수 있습니다.

```
$ pcluster describe-cluster -n ad-cluster --region "region-id" --query "clusterStatus"
```

출력값은 다음과 같습니다.

```
"CREATE_IN_PROGRESS" / "CREATE_COMPLETE"
```

상태가 "CREATE\_COMPLETE"가 되면 생성한 사용자 이름과 암호로 로그인합니다.

```
$ HEAD_NODE_IP=$(pcluster describe-cluster -n "ad-cluster" --region "region-id" --query headNode.publicIpAddress | xargs echo)
```

```
$ ssh user000@$HEAD_NODE_IP
```

/home/user000@HEAD\_NODE\_IP/.ssh/id\_rsa에서 새 사용자를 위해 만든 SSH 키를 입력하면 암호 없이 로그인할 수 있습니다.

ssh 명령이 성공하면 Active Director (AD)를 사용하도록 인증된 사용자로 클러스터에 성공적으로 연결한 것입니다.

#### 5단계: 정리

1. 로컬 시스템에서 클러스터를 삭제합니다.

```
$ pcluster delete-cluster --cluster-name "ad-cluster" --region "region-id"
{
```

```

"cluster": {
  "clusterName": "ad-cluster",
  "cloudformationStackStatus": "DELETE_IN_PROGRESS",
  "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
  "region": "region-id",
  "version": "3.7.0",
  "clusterStatus": "DELETE_IN_PROGRESS"
}
}

```

2. 삭제 중인 클러스터의 진행 상태를 확인합니다.

```

$ pcluster describe-cluster --cluster-name "ad-cluster" --region "region-id" --
query "clusterStatus"
"DELETE_IN_PROGRESS"

```

클러스터가 성공적으로 삭제되면 다음 단계를 진행합니다.

## 자동

Active Directory 리소스를 삭제합니다.

1. <https://console.aws.amazon.com/cloudformation/>으로 이동합니다.
2. 탐색 창에서 스택을 선택합니다.
3. 스택 목록에서 AD 스택(예:pcluster-ad)을 선택합니다.
4. 삭제를 선택합니다.

## 수동

1. Amazon EC2 인스턴스를 삭제합니다.
  - a. <https://console.aws.amazon.com/ec2/> 에서 탐색 창에서 인스턴스를 선택합니다.
  - b. 디렉터리에 사용자를 추가하기 위해 생성한 인스턴스를 인스턴스 목록에서 선택합니다.
  - c. 인스턴스 상태를 선택한 뒤 인스턴스 종료를 선택하세요.
2. 호스팅 영역을 삭제합니다.
  - a. 다음과 같은 콘텐츠로 recordset-delete.json을 생성합니다. 이 예제에서 HostedZoneId 는 로드 밸런서의 표준 호스팅 영역 ID입니다.

```
{
  "Changes": [
    {
      "Action": "DELETE",
      "ResourceRecordSet": {
        "Name": "corp.example.com",
        "Type": "A",
        "Region": "region-id",
        "SetIdentifier": "pcluster-active-directory",
        "AliasTarget": {
          "HostedZoneId": "Z2IF0LAFXWL04F",
          "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}
```

- b. 호스팅 영역 ID를 사용하여 레코드세트 변경 내용을 호스팅 영역에 제출하세요.

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXMSJUB \
  --change-batch file://recordset-delete.json
{
  "ChangeInfo": {
    "Id": "/change/C04853642A0TH2TJ5NLNI",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T14:25:51.046000+00:00"
  }
}
```

- c. 호스팅 영역을 삭제합니다.

```
$ aws route53 delete-hosted-zone --id Z09020002B5MZQNXMSJUB
{
  "ChangeInfo": {
    "Id": "/change/C0468051QFABTVHMDEG9",
    "Status": "PENDING",
    "SubmittedAt": "2022-05-05T14:26:13.814000+00:00"
  }
}
```



```
}
```

3. LB 리스너를 삭제합니다.

```
$ aws elbv2 delete-listener \  
  --listener-arn arn:aws:elasticloadbalancing:region-id:123456789012:listener/net/  
  CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b --region region-id
```

4. 대상 그룹을 삭제합니다.

```
$ aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:region-  
  id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 --  
  region region-id
```

5. 로드 밸런서를 삭제합니다.

```
$ aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:region-  
  id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4 --  
  region region-id
```

6. 클러스터가 Secrets Manager에서 인증서를 읽는 데 사용하는 정책을 삭제합니다.

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/  
  ReadCertExample
```

7. 도메인 인증서가 포함된 암호를 삭제합니다.

```
$ aws secretsmanager delete-secret \  
  --secret-id arn:aws:secretsmanager:region-id:123456789012:secret:example-  
  cert-123abc \  
  --region region-id  
{  
  "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",  
  "Name": "example-cert",  
  "DeletionDate": "2022-06-04T16:27:36.183000+02:00"  
}
```

8. ACM에서 인증서를 삭제합니다.

```
$ aws acm delete-certificate \  

```

```
--certificate-arn arn:aws:acm:region-id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 --region region-id
```

## 9. Active Directory(AD) 리소스를 삭제합니다.

a. Python 스크립트 `ad.py`의 출력에서 다음 리소스 ID를 가져옵니다.

- AD ID
- AD 서브넷 ID
- AD VPC ID

b. 다음 명령을 실행하여 디렉터리를 삭제합니다.

```
$ aws ds delete-directory --directory-id d-abcdef0123456789 --region region-id
{
  "DirectoryId": "d-abcdef0123456789"
}
```

c. VPC의 보안 그룹을 나열합니다.

```
$ aws ec2 describe-security-groups --filters '[{"Name":"vpc-id","Values":["vpc-07614ade95ebad1bc"]}]' --region region-id
```

d. 사용자 지정 보안 그룹을 삭제합니다.

```
$ aws ec2 delete-security-group --group-id sg-021345abcdef6789 --region region-id
```

e. 서브넷을 삭제합니다.

```
$ aws ec2 delete-subnet --subnet-id subnet-1234567890abcdef --region region-id
```

```
$ aws ec2 delete-subnet --subnet-id subnet-021345abcdef6789 --region region-id
```

f. 인터넷 게이트웨이를 설명합니다.

```
$ aws ec2 describe-internet-gateways \
  --filters Name=attachment.vpc-id,Values=vpc-021345abcdef6789 \
  --region region-id
{
  "InternetGateways": [
```

```

    "Attachments": [
      {
        "State": "available",
        "VpcId": "vpc-021345abcdef6789"
      }
    ],
    "InternetGatewayId": "igw-1234567890abcdef",
    "OwnerId": "123456789012",
    "Tags": []
  }
]
}

```

- g. 인터넷 게이트웨이를 분리합니다.

```

$ aws ec2 detach-internet-gateway \
  --internet-gateway-id igw-1234567890abcdef \
  --vpc-id vpc-021345abcdef6789 \
  --region region-id

```

- h. 인터넷 게이트웨이를 삭제합니다.

```

$ aws ec2 delete-internet-gateway \
  --internet-gateway-id igw-1234567890abcdef \
  --region region-id

```

- i. VPC를 삭제합니다.

```

$ aws ec2 delete-vpc \
  --vpc-id vpc-021345abcdef6789 \
  --region region-id

```

- j. ReadOnlyUser 암호가 포함된 보안 암호를 삭제합니다.

```

$ aws secretsmanager delete-secret \
  --secret-id arn:aws:secretsmanager:region-  
id:123456789012:secret:ADSecretPassword-1234 \
  --region region-id

```

# AWS KMS 키를 사용한 공유 스토리지 암호화 구성

고객 관리 AWS KMS 키를 설정하여 구성된 클러스터 파일 스토리지 시스템에서 데이터를 암호화하고 보호하는 방법을 알아보십시오. [AWS ParallelCluster](#)

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

AWS ParallelCluster 다음과 같은 공유 스토리지 구성 옵션을 지원합니다.

- [SharedStorage](#) / [EbsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [EfsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [FsxLustreSettings](#) / [KmsKeyId](#)

이러한 옵션을 사용하여 Amazon EBS, Amazon EFS 및 FSx for Lustre 공유 스토리지 시스템 암호화를 위한 고객 관리 AWS KMS 키를 제공할 수 있습니다. 이를 사용하려면 다음 항목에 대한 IAM 정책을 만들고 구성해야 합니다.

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

## 사전 조건

- AWS ParallelCluster [설치되었습니다](#).
- AWS CLI [가 설치 및 구성되었습니다](#).
- [Amazon EC2 키](#) 페어가 있습니다.
- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.

## 주제

- [정책을 생성합니다](#).
- [클러스터 구성 및 생성](#)

## 정책을 생성합니다.

정책을 생성합니다.

1. IAM 콘솔 <https://console.aws.amazon.com/iam/home>으로 이동하세요.
2. 정책을 선택하세요.
3. 정책 생성을 선택합니다.
4. JSON 탭을 선택하고 다음 정책을 붙여넣습니다. 모든 항목을 사용자 AWS 계정 ID와 키 Amazon 리소스 이름 (ARN) *123456789012* 으로 바꾸고 자신의 것으로 바꿔야 합니다. AWS 리전

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:region-id:123456789012:key/abcd1234-ef56-gh78-ij90-abcd1234efgh5678"
      ]
    }
  ]
}
```

5. 이 튜토리얼에서는 정책 이름을 ParallelClusterKmsPolicy로 입력한 후 정책 생성을 선택하세요.
6. 정책 ARN을 기록해 둡니다. 클러스터를 구성하는 데 필요합니다.

## 클러스터 구성 및 생성

다음은 암호화가 적용된 Amazon Elastic Block Store 공유 파일 시스템을 포함하는 예제 클러스터 구성입니다.

```

Region: eu-west-1
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh:
    KeyName: my-ssh-key
  Iam:
    AdditionalIamPolicies:
      - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: q1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 0
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
SharedStorage:
  - MountDir: /shared/ebs1
    Name: shared-ebs1
    StorageType: Ebs
    EbsSettings:
      Encrypted: True
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678

```

빨간색 텍스트의 아이템을 원하는 값으로 교체합니다. 그런 다음 AWS KMS 키를 사용하여 Amazon EBS에서 데이터를 암호화하는 클러스터를 생성합니다.

구성은 Amazon EFS 및 FSx for Lustre 파일 시스템과 유사합니다.

Amazon EFS SharedStorage 구성은 다음과 같습니다.

```

...
SharedStorage:
  - MountDir: /shared/efs1
    Name: shared-efs1
    StorageType: Efs
    EfsSettings:
      Encrypted: True
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678

```

FSx for Lustre SharedStorage 구성은 다음과 같습니다.

```

...
SharedStorage:
  - MountDir: /shared/fsx1
    Name: shared-fsx1
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      DeploymentType: PERSISTENT_1
      PerUnitStorageThroughput: 200
      KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678

```

## 다중 대기열 모드 클러스터에서 작업 실행

이 자습서에서는 [다중 대기열](#) 모드에서 첫 번째 Hello World "" 작업을 실행하는 AWS ParallelCluster 방법을 다룹니다.

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

### 사전 조건

- AWS ParallelCluster [설치되었습니다](#).
- AWS CLI [가 설치 및 구성되었습니다](#).
- [Amazon EC2 키 페어](#)가 있습니다.

- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.

## 클러스터 구성

먼저 다음 명령을 실행하여 제대로 AWS ParallelCluster 설치되었는지 확인합니다.

```
$ pcluster version
```

pcluster version에 대한 자세한 정보는 [pcluster version](#) 섹션을 참조하십시오.

이 명령은 실행 중인 버전을 반환합니다 AWS ParallelCluster.

다음으로 pcluster configure을 실행하여 기본 구성 파일을 생성합니다. 이 명령 다음에 나오는 모든 메시지를 따릅니다.

```
$ pcluster configure --config multi-queue-mode.yaml
```

pcluster configure 명령에 대한 자세한 내용은 [pcluster configure](#) 섹션을 참조하세요.

이 단계를 완료한 후에는 multi-queue-mode.yaml이라는 기본 구성 파일이 나타납니다. 이 파일에는 기본 클러스터 구성이 들어 있습니다.

다음 단계에서는 새 구성 파일을 수정하고 대기열이 여러 개 있는 클러스터를 시작합니다.

### Note

이 자습서에서 사용된 일부 인스턴스는 프리 티어에 사용할 수 없습니다.

이 자습서에서는 다음 구성과 일치하도록 구성 파일을 수정하세요. 빨간색으로 강조 표시된 항목은 구성 파일 값을 나타냅니다. 자신의 고유한 값을 유지하세요.

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: c5.xlarge
Networking:
  SubnetId: subnet-abcdef01234567890
Ssh:
```



```

  KeyName: yourkeypair
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - Name: spot
    ComputeResources:
    - Name: c5xlarge
      InstanceType: c5.xlarge
      MinCount: 1
      MaxCount: 10
    - Name: t2micro
      InstanceType: t2.micro
      MinCount: 1
      MaxCount: 10
  Networking:
    SubnetIds:
    - subnet-abcdef01234567890
  - Name: ondemand
    ComputeResources:
    - Name: c52xlarge
      InstanceType: c5.2xlarge
      MinCount: 0
      MaxCount: 10
  Networking:
    SubnetIds:
    - subnet-021345abcdef6789

```

## 클러스터 생성

구성 파일을 기반으로 multi-queue-cluster라는 이름이 지정된 클러스터를 만드세요.

```

$ pcluster create-cluster --cluster-name multi-queue-cluster --cluster-configuration
multi-queue-mode.yaml
{
  "cluster": {
    "clusterName": "multi-queue-cluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}

```

```
}

```

`pcluster create-cluster` 명령에 대한 자세한 내용은 [pcluster create-cluster](#) 섹션을 참조하세요.

다음 명령을 실행하여 클러스터 상태를 확인합니다.

```
$ pcluster list-clusters
{
  "cluster": {
    "clusterName": "multi-queue-cluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
```

클러스터가 생성되면 `clusterStatus` 필드가 `CREATE_COMPLETE`를 표시합니다.

## 헤드 노드에 로그인

프라이빗 SSH 키 파일을 사용하여 헤드 노드에 로그인합니다.

```
$ pcluster ssh --cluster-name multi-queue-cluster -i ~/path/to/yourkeyfile.pem
```

`pcluster ssh`에 대한 자세한 내용은 [pcluster ssh](#) 섹션을 참조하세요.

로그인되면 `sinfo` 명령을 실행하여 스케줄러 대기열이 설정 및 구성되어 있는지 확인합니다.

`sinfo`에 대한 자세한 내용은 Slurm 설명서에서 [sinfo](#) 섹션을 참조하세요.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
spot*      up    infinite   18  idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[1-9]
spot*      up    infinite    2  idle  spot-st-c5xlarge-1,spot-st-t2micro-1
ondemand   up    infinite   10  idle~ ondemand-dy-c52xlarge-[1-10]
```

출력에는 클러스터에서 사용할 수 있는 `idle` 상태에 `t2.micro` 하나 및 `c5.xlarge` 컴퓨팅 노드 하나가 있는 것으로 표시됩니다.

다른 노드는 모두 절전 상태이며, 노드 상태의 ~ 접미사로 표시되며, 이를 지원하는 Amazon EC2 인스턴스가 없습니다. 기본 대기열은 대기열 이름 뒤에 \* 접미사로 표시됩니다. spot은 기본 작업 대기열입니다.

## 다중 대기열 모드에서 작업 실행

그런 다음 작업을 실행하여 잠시 휴면 모드로 전환해 보세요. 작업은 나중에 자체 호스트 이름을 출력합니다. 현재 사용자가 이 스크립트를 실행할 수 있는지 확인하세요.

```
$ tee <<EOF hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from \$(hostname)"
EOF

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

sbatch 명령을 사용하여 작업을 제출합니다. -N 2 옵션으로 이 작업의 노드 두 개를 요청하고 작업이 성공적으로 제출되는지 확인합니다. sbatch에 대한 자세한 내용은 Slurm 설명서에서 [sbatch](#) 섹션을 참조하세요.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 1
```

squeue 명령으로 대기열을 보고 작업 상태를 확인할 수 있습니다. 단, 특정 대기열을 지정하지 않았으므로 기본 대기열(spot)이 사용됩니다. squeue에 대한 자세한 내용은 Slurm 설명서에서 [squeue](#) 섹션을 참조하세요.

```
$ squeue
JOBID PARTITION   NAME     USER  ST        TIME  NODES NODELIST(REASON)
   1      spot     wrap ec2-user  R           0:10     2 spot-st-c5xlarge-1,spot-st-t2micro-1
```

출력을 통해 작업이 현재 실행 중 상태인 것을 알 수 있습니다. 작업을 마칠 때까지 기다리세요. 이 작업에는 약 30초 정도 걸립니다. 그런 다음 squeue를 다시 실행하세요.

```
$ squeue
JOBID PARTITION   NAME     USER  ST        TIME  NODES NODELIST(REASON)
```

이제 대기열의 작업이 모두 완료되었으니 현재 디렉터리에서 `slurm-1.out`이라는 이름의 출력 파일을 찾아보세요.

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```

출력을 통해 `spot-st-t2micro-1` 및 `spot-st-c5xlarge-1` 노드에서 작업이 성공적으로 실행되었음을 알 수 있습니다.

이제 다음 명령으로 특정 인스턴스에 대한 제약 조건을 지정하여 동일한 작업을 제출하세요.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 2
```

이 파라미터를 `sbatch`에 사용했습니다.

- `-N 3`- 세 개의 노드를 요청합니다.
- `-p spot`- 작업을 `spot` 대기열에 제출합니다. `-p ondemand`를 지정하여 작업을 `ondemand` 대기열에 제출할 수도 있습니다.
- `-C "[c5.xlarge*1&t2.micro*2]"`- 이 작업에 대한 특정 노드 제약 조건을 지정합니다. 이것은 이 작업에 사용할 `c5.xlarge` 노드 1개와 `t2.micro` 노드 2개를 요청합니다.

`sinfo` 명령을 실행하여 노드와 대기열을 확인합니다. 에 있는 대기열을 파티션 인이라고 합니다.  
AWS ParallelCluster Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
spot*      up    infinite   1    alloc# spot-dy-t2micro-1
spot*      up    infinite  17    idle~  spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot*      up    infinite   1    mix    spot-st-c5xlarge-1
spot*      up    infinite   1    alloc  spot-st-t2micro-1
ondemand   up    infinite  10    idle~  ondemand-dy-c52xlarge-[1-10]
```

노드에 전원이 공급되고 있습니다. 이는 노드 상태에 `#` 접미사가 붙는 것으로 표시됩니다. `squeue` 명령을 실행하여 클러스터에서 작업에 대한 정보를 봅니다.

```
$ squeue
```

```

JOBID PARTITION    NAME      USER ST      TIME  NODES NODELIST(REASON)
  2     spot      wrap ec2-user CF      0:04    3 spot-dy-c5xlarge-1,spot-dy-
t2micro-1,spot-st-t2micro-1

```

작업은 CF(CONFIGURING) 상태이며, 인스턴스가 스케일 업되어 클러스터에 합류하기를 기다리고 있습니다.

약 3분 후에 노드를 사용할 수 있고 작업이 R(RUNNING) 상태로 전환됩니다.

```

$ squeue
JOBID PARTITION    NAME      USER ST      TIME  NODES NODELIST(REASON)
  2     spot      wrap ec2-user R      0:07    3 spot-dy-t2micro-1,spot-st-
c5xlarge-1,spot-st-t2micro-1

```

작업이 완료되어 세 노드 모두 idle 상태입니다.

```

$ squeue
JOBID PARTITION    NAME      USER ST      TIME  NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
spot*      up     infinite   17    idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot*      up     infinite    3    idle  spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand   up     infinite   10    idle~ ondemand-dy-c52xlarge-[1-10]

```

그런 다음 대기열에 작업이 남아 있지 않으면 로컬 디렉토리에서 `slurm-2.out`을 확인하세요.

```

$ cat slurm-2.out
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
Hello World from spot-st-c5xlarge-1

```

클러스터의 최종 상태입니다.

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
spot*      up     infinite   17    idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot*      up     infinite    3    idle  spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand   up     infinite   10    idle~ ondemand-dy-c52xlarge-[1-10]

```

클러스터를 로그오프한 후 `pcluster delete-cluster`를 실행하여 정리할 수 있습니다. 자세한 내용은 [pcluster list-clusters](#) 및 [pcluster delete-cluster](#) 단원을 참조하세요.

```
$ pcluster list-clusters
{
  "clusters": [
    {
      "clusterName": "multi-queue-cluster",
      "cloudformationStackStatus": "CREATE_COMPLETE",
      "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
      "region": "eu-west-1",
      "version": "3.1.4",
      "clusterStatus": "CREATE_COMPLETE"
    }
  ]
}
$ pcluster delete-cluster -n multi-queue-cluster
{
  "cluster": {
    "clusterName": "multi-queue-cluster",
    "cloudformationStackStatus": "DELETE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
    "region": "eu-west-1",
    "version": "3.1.4",
    "clusterStatus": "DELETE_IN_PROGRESS"
  }
}
```

## AWS ParallelCluster API 사용하기

이 자습서에서는 [Amazon API Gateway와 AWS ParallelCluster CloudFormation 템플릿을 사용하여 API를 구축하고 테스트합니다.](#) 그런 다음 에서 사용할 수 있는 예제 클라이언트를 GitHub 사용하여 API를 사용합니다. API 사용에 대한 자세한 내용은 [AWS ParallelCluster API](#)를 참조하세요.

이 튜토리얼은 [공공 부문 고객 워크숍용 HPC](#)에서 발췌했습니다.

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

## 사전 조건

- 컴퓨팅 환경에 [설치](#) 및 구성됩니다. AWS CLI
- AWS ParallelCluster 가상 환경에 설치됩니다. 자세한 내용은 [가상 환경에 설치를 AWS ParallelCluster](#) 참조하십시오.
- [Amazon EC2 키](#) 페어가 있습니다.
- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.

## 1단계: Amazon API Gateway를 사용하여 API 구축

홈 사용자 디렉터리에 머물면서 가상 환경을 활성화하세요.

1. 유용한 JSON 명령줄 프로세서를 설치하세요.

```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. 다음 명령을 실행하여 AWS ParallelCluster 버전을 가져오고 환경 변수에 할당합니다.

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" |tee -a ~/.bashrc
```

3. 환경 변수를 만들고 해당 변수에 리전 ID를 할당합니다.

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" |tee -a ~/.bashrc
```

4. 다음 명령을 실행하여 VPC를 배포합니다.

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" |tee -a ~/.bashrc
```

```
aws cloudformation create-stack \
  --region ${AWS_DEFAULT_REGION} \
  --stack-name ${API_STACK_NAME} \
```

```

--template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
parallelcluster-api.yaml \
--capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
--parameters ParameterKey=EnableIamAdminAccess,ParameterValue=true

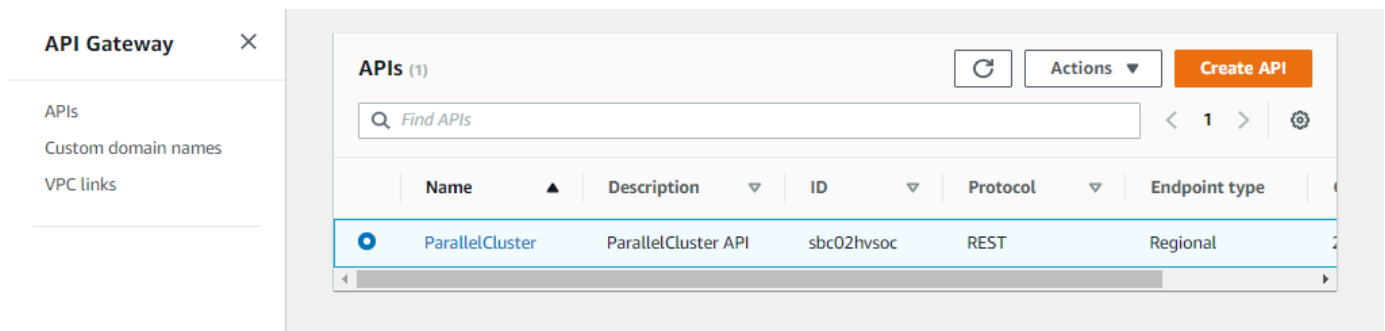
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-api-
stack/abcd1234-ef56-gh78-ei90-1234abcd5678"
}

```

프로세스가 완료되면 다음 단계로 진행합니다.

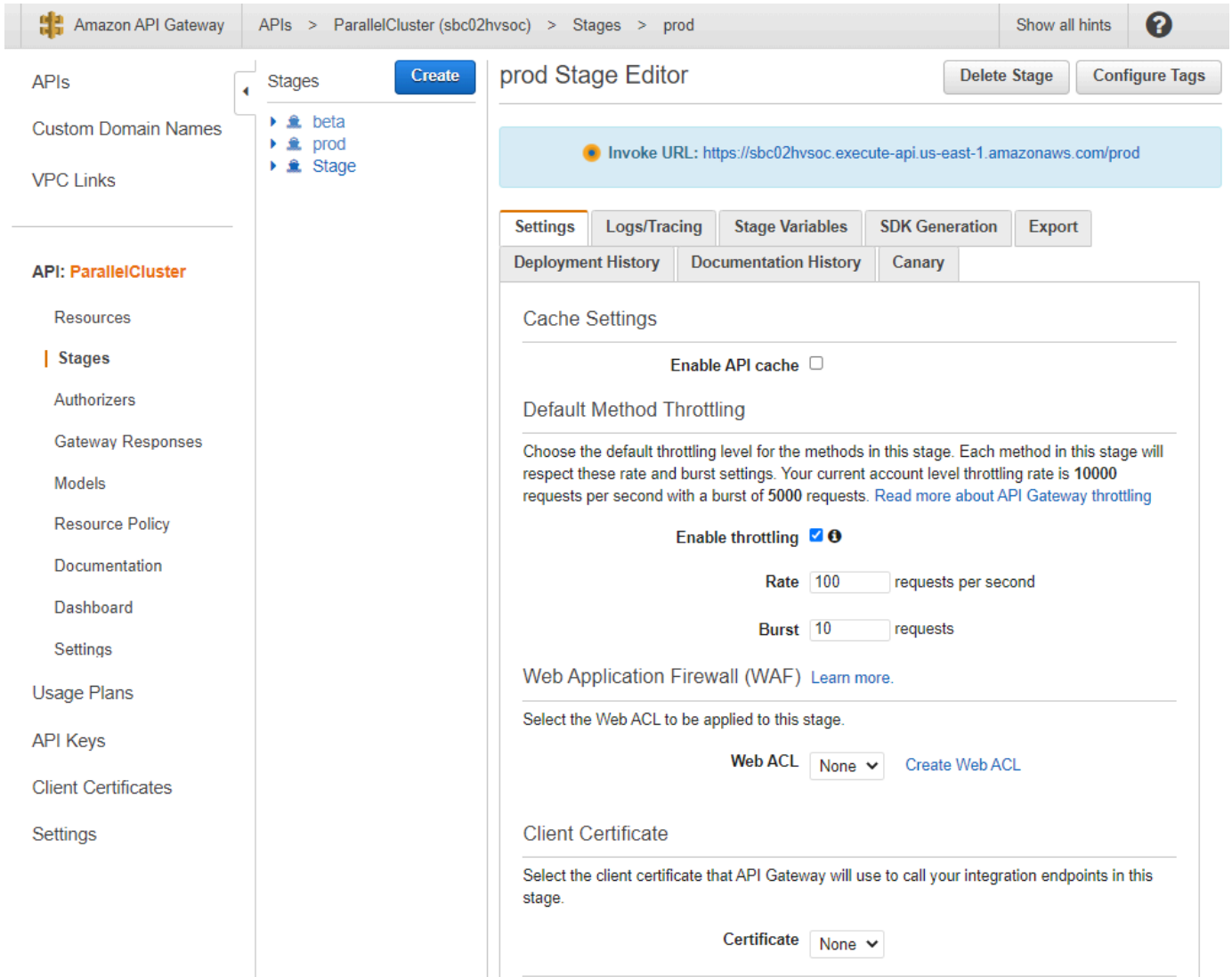
## 2단계: Amazon API Gateway 콘솔의 API를 테스트합니다.

1. AWS Management Console에 로그인합니다.
2. [Amazon API Gateway 콘솔](#)로 이동합니다.
3. API 배포를 선택합니다.



4. 스테이지를 선택하고 스테이지를 선택합니다.





5. API 게이트웨이에서 API에 액세스하거나 API를 간접 호출하기 위해 제공하는 URL을 기록해 둡니다. 이것은 파란색으로 강조 표시되어 있습니다.
6. 리소스를 선택하고 **/clusters** 아래의 **GET**을 선택합니다.
7. 테스트 아이콘을 선택한 다음 아래로 스크롤하여 테스트 아이콘을 선택합니다.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET

Resources Actions **/v3/clusters - GET - Method Execution**

- /
- /v3
  - /clusters
    - GET**
    - POST
    - /{clusterName}
      - DELETE
      - GET
      - PUT
    - /computefleet
      - GET
      - PATCH
    - /instances
      - DELETE
      - GET
    - /logstreams
      - GET
      - /{logStreamName}
        - GET
    - /stackevents
      - GET
  - /images
    - /custom

**Client** (TEST) →

**Method Request**

- Auth: AWS IAM
- ARN: arn:aws:execute-api:us-east-1:123456789012:sbc02hvsoc/\*/GET/v3/clusters
- Query Strings: region, nextToken, clusterStatus

← **Method Response**

Select an integration response.

/clusters GET에 대한 응답이 나타납니다.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET

Show all hints ?

Resources Actions

← Method Execution /v3/clusters - GET - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

**Path**

No path parameters exist for this resource. You can define path parameters by using the syntax `{myPathParam}` in a resource path.

**Request:** /v3/clusters

**Status:** 200

**Latency:** 3203 ms

**Response Body**

```
{
  "clusters": [
    {
      "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/test-cluster/4450d850-b684-11ec-84a7-0a047567c9f3",
      "cloudformationStackStatus": "CREATE_COMPLETE",
      "clusterName": "test-cluster",
      "clusterStatus": "CREATE_COMPLETE",
      "region": "us-east-1",
      "version": "3.1.2"
    }
  ]
}
```

**Query Strings**

{clusters}

param1=value1&param2=value2

**Headers**

{clusters}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

**Stage Variables**

No stage variables exist for this method.

**Client Certificate**

No client certificates have been generated.

**Response Headers**

```
{"Content-Length": "360", "X-Amzn-Trace-Id": "Root=1-62686455-c1cf243417b2721e33822ac5;Sampled=1", "Content-Type": "application/json"}
```

**Logs**

### 3단계: API 간접 호출을 위한 예제 클라이언트 준비 및 테스트

AWS ParallelCluster 소스 코드를 `cd api` 디렉터리에 복제하고 Python 클라이언트 라이브러리를 설치합니다.

1. 

```
$ git clone -b v${PCLUSTER_VERSION} https://github.com/aws/aws-parallelcluster aws-parallelcluster-v${PCLUSTER_VERSION}
cd aws-parallelcluster-v${PCLUSTER_VERSION}/api
```

```
$ pip3 install client/src
```

2. 홈 사용자 디렉토리로 돌아갑니다.
3. 클라이언트가 실행 시 사용하는 API 게이트웨이 기본 URL을 내보냅니다.

```
$ export PCLUSTER_API_URL=$( aws cloudformation describe-stacks
  --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiInvokeUrl`.OutputValue' --output text )
echo "export PCLUSTER_API_URL=${PCLUSTER_API_URL}" |tee -a ~/.bashrc
```

4. 클라이언트가 클러스터를 생성하는 데 사용하는 클러스터 이름을 내보냅니다.

```
$ export CLUSTER_NAME="test-api-cluster"
echo "export CLUSTER_NAME=${CLUSTER_NAME}" |tee -a ~/.bashrc
```

5. 다음 명령을 실행하여 예제 클라이언트가 API에 액세스하는 데 사용하는 보안 인증을 저장합니다.

```
$ export PCLUSTER_API_USER_ROLE=$( aws cloudformation describe-
stacks --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiUserRole`.OutputValue' --output text )
echo "export PCLUSTER_API_USER_ROLE=${PCLUSTER_API_USER_ROLE}" |tee -a ~/.bashrc
```

#### 4단계: 클라이언트 코드 스크립트 복사 및 클러스터 테스트 실행

1. 다음 예제 클라이언트 코드를 홈 사용자 디렉토리의 `test_pcluster_client.py`에 복사합니다. 클라이언트 코드는 다음을 수행하도록 요청합니다.
  - 클러스터를 생성합니다.
  - 클러스터를 설명합니다.
  - 클러스터를 나열합니다.
  - 컴퓨팅 플릿을 설명합니다.
  - 클러스터 인스턴스를 설명합니다.

```
# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
this
# software and associated documentation files (the "Software"), to deal in the
Software
```

```
# without restriction, including without limitation the rights to use, copy,
# modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and
# to
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
# COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)
    fleet_ops = cluster_compute_fleet_api.ClusterComputeFleetApi(api_client)
    instance_ops = cluster_instances_api.ClusterInstancesApi(api_client)
```

```
# Create cluster
build_done = False
try:
    with open('cluster-config.yaml', encoding="utf-8") as f:
        body = CreateClusterRequestContent(cluster_name=cluster_name,
cluster_configuration=f.read())
        api_response = cluster_ops.create_cluster(body, region=region)
except pcluster_client.ApiException as e:
    print("Exception when calling create_cluster: %s\n" % e)
    build_done = True
time.sleep(60)

# Confirm cluster status with describe_cluster
while not build_done:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('CREATE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
        elif api_response.cluster_status == ClusterStatus('CREATE_COMPLETE'):
            print('READY!')
            build_done = True
        else:
            print('ERROR!!!!')
            build_done = True
    except pcluster_client.ApiException as e:
        print("Exception when calling describe_cluster: %s\n" % e)

# List clusters
try:
    api_response = cluster_ops.list_clusters(region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling list_clusters: %s\n" % e)

# DescribeComputeFleet
try:
    api_response = fleet_ops.describe_compute_fleet(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling compute fleet: %s\n" % e)
```

```
# DescribeClusterInstances
try:
    api_response = instance_ops.describe_cluster_instances(cluster_name,
region=region)
    pprint(api_response)
except pcluster_client.ApiException as e:
    print("Exception when calling describe_cluster_instances: %s\n" % e)
```

- 클러스터 구성을 생성합니다.

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

- API 클라이언트 라이브러리는 환경 변수(예: `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY` 또는 `AWS_SESSION_TOKEN`) 또는 `$HOME/.aws`에서 구성 세부 정보를 자동으로 검색합니다. 다음 명령은 현재 IAM 역할을 지정된 `ParallelClusterApiUserRole`로 전환합니다.

```
$ eval $(aws sts assume-role --role-arn ${PCLUSTER_API_USER_ROLE} --role-session-name ApiTestSession | jq -r '.Credentials | "export AWS_ACCESS_KEY_ID=\(.AccessKeyId)\nexport AWS_SECRET_ACCESS_KEY=\(.SecretAccessKey)\nexport AWS_SESSION_TOKEN=\(.SessionToken)\n"')
```

주의해야 할 오류:

다음과 비슷한 오류가 표시되면 이미 `ParallelClusterApiUserRole` 및 `AWS_SESSION_TOKEN`이 만료된 것으로 가정한 것입니다.

```
An error occurred (AccessDenied) when calling the AssumeRole operation:
User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/ApiTestSession
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::XXXXXXXXXXXX:role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

역할을 삭제한 다음 `aws sts assume-role` 명령을 다시 실행하여 `ParallelClusterApiUserRole` 을 사용하세요.

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

현재 사용자에게 API 액세스 권한을 제공하려면 [리소스 정책을 확장해야](#) 합니다.

4. 다음 명령을 실행하여 예제 클라이언트를 시작합니다.

```
$ python3 test_pcluster_client.py
{'cluster_configuration': 'Region: us-east-1\n'
                          'Image:\n'
                          '  Os: alinux2\n'
                          'HeadNode:\n'
                          '  InstanceType: t2.micro\n'
                          '  Networking . . . :\n'
                          '    SubnetId: subnet-1234567890abcdef0\n'
                          '  Ssh:\n'
                          '    KeyName: adpc\n'
                          'Scheduling:\n'
                          '  Scheduler: slurm\n'
                          '  SlurmQueues:\n'
                          '    - Name: queue1\n'
                          '      ComputeResources:\n'
                          '        - Name: t2micro\n'
                          '          InstanceType: t2.micro\n'
                          '          MinCount: 0\n'
                          '          MaxCount: 10\n'
                          '          Networking . . . :\n'
                          '            SubnetIds:\n'
                          '              - subnet-1234567890abcdef0\n',
  'cluster_name': 'test-api-cluster'}
{'cloud_formation_stack_status': 'CREATE_IN_PROGRESS',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
  'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-not-delete...'},
  'cluster_name': 'test-api-cluster',
  'cluster_status': 'CREATE_IN_PROGRESS',
  'compute_fleet_status': 'UNKNOWN',
  'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
  'region': 'us-east-1',
  'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
  'version': '3.1.3'}
```



```

.
.
. . . working . . . {'cloud_formation_stack_status': 'CREATE_COMPLETE',
  'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
  'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
  'cluster_name': 'test-api-cluster',
  'cluster_status': 'CREATE_COMPLETE',
  'compute_fleet_status': 'RUNNING',
  'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
  'head_node': {'instance_id': 'i-abcdef01234567890',
    'instance_type': 't2.micro',
    'launch_time': datetime.datetime(2022, 4, 28, 16, 21, 46,
tzinfo=tzlocal()),
    'private_ip_address': '172.31.27.153',
    'public_ip_address': '52.90.156.51',
    'state': 'running'},
  'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
  'region': 'us-east-1',
  'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
  'version': '3.1.3'}
READY!

```

## 5단계: 클라이언트 코드 스크립트 복사 및 클러스터 삭제

1. 다음 예제 클라이언트 코드를 `delete_cluster_client.py`에 복사합니다. 클라이언트 코드는 클러스터 삭제를 요청합니다.

```

# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
this
# software and associated documentation files (the "Software"), to deal in the
Software
# without restriction, including without limitation the rights to use, copy,
modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and
to

```

```
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
# COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
# Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
    cluster_compute_fleet_api,
    cluster_instances_api,
    cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
    CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

# Defining the host is optional and defaults to http://localhost
# See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
    host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

# Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
    cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)

    # Delete the cluster
    gone = False
    try:
        api_response = cluster_ops.delete_cluster(cluster_name, region=region)
    except pcluster_client.ApiException as e:
        print("Exception when calling delete_cluster: %s\n" % e)
```

```

time.sleep(60)

# Confirm cluster status with describe_cluster
while not gone:
    try:
        api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
        pprint(api_response)
        if api_response.cluster_status == ClusterStatus('DELETE_IN_PROGRESS'):
            print('. . . working . . .', end='', flush=True)
            time.sleep(60)
    except pcluster_client.ApiException as e:
        gone = True
        print("DELETE COMPLETE or Exception when calling describe_cluster: %s
\n" % e)

```

2. 다음 명령을 실행하여 클러스터를 삭제합니다.

```

$ python3 delete_cluster_client.py
{'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 53, 48,
tzinfo=tzlocal()),
'private_ip_address': '172.31.17.132',
'public_ip_address': '34.201.100.37',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
.
.
.

```

```
. . . working . . . {'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
. . . working . . . DELETE COMPLETE or Exception when calling describe_cluster:
(404)
Reason: Not Found
.
.
.
HTTP response body: {"message": "Cluster 'test-api-cluster' does not exist or
belongs to an incompatible ParallelCluster major version."}
```

### 3. 테스트를 마친 후 환경 변수를 설정 해제하세요.

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

## 6단계: 정리

AWS Management Console AWS CLI or를 사용하여 API를 삭제할 수 있습니다.

1. AWS CloudFormation 콘솔에서 API 스택을 선택한 다음 삭제를 선택합니다.
2. AWS CLI를 사용 중이라면 다음 명령을 실행합니다.

사용 AWS CloudFormation.

```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```

## Slurm 회계를 사용하여 클러스터 생성

Slurm 회계를 사용하여 클러스터를 구성하고 생성하는 방법을 알아보세요. 자세한 정보는 [Slurm 회계 담당자: AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 정보는 [AWS ParallelCluster UI 비용](#)을 참조하세요.

이 자습서에서는 [CloudFormation 빠른 생성 템플릿 \(us-east-1\)을 사용하여 MySQL 서버리스](#) 데이터베이스를 생성합니다. [Amazon Aurora](#) 템플릿은 클러스터와 CloudFormation 동일한 VPC에 Amazon Aurora 서버리스 데이터베이스를 배포하는 데 필요한 모든 구성 요소를 만들도록 지시합니다. 또한 템플릿은 클러스터와 데이터베이스 간 연결을 위한 기본 네트워킹 및 보안 구성을 생성합니다.

### Note

[버전 3.3.0부터 클러스터 구성 파라미터/Database를 사용한 Slurm 어카운팅을 AWS ParallelCluster](#) 지원합니다. [SlurmSettings](#)

### Note

빠른 생성 템플릿이 그 예시입니다. 이 템플릿은 Slurm 회계 데이터베이스 서버의 가능한 모든 사용 사례를 다루지는 않습니다. 프로덕션 워크로드에 적합한 구성과 용량을 갖춘 데이터베이스 서버를 만드는 것은 사용자의 책임입니다.

사전 조건:

- AWS ParallelCluster [설치되었습니다](#).
- AWS CLI [가 설치 및 구성되었습니다](#).
- [Amazon EC2 키](#) 페어가 있습니다.
- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할이 있습니다.
- 빠른 생성 템플릿을 배포하는 지역은 Amazon Aurora MySQL 서버리스 v2를 지원합니다. 자세한 내용은 [Aurora MySQL을 사용하는 Aurora Serverless v2](#)을 참조하세요.

## 1단계: VPC 및 서브넷 생성 AWS ParallelCluster

제공된 CloudFormation 템플릿을 Slurm 계정 데이터베이스에 사용하려면 클러스터용 VPC가 준비되어 있어야 합니다. 이 작업은 수동으로 또는 [AWS ParallelCluster 명령줄 인터페이스를 사용하여 클러스터를 구성하고 생성합니다](#). 절차의 일부로 수행할 수 있습니다. 이미 AWS ParallelCluster를 사용한 경우 클러스터 및 데이터베이스 서버를 배포할 수 있는 VPC가 준비되어 있을 수 있습니다.

## 2단계: 데이터베이스 스택 생성

[CloudFormation 빠른 생성 템플릿 \(us-east-1\)](#) 을 사용하여 어카운팅을 위한 데이터베이스 스택을 생성합니다. Slurm 템플릿에는 다음과 같은 입력이 필요합니다.

- 데이터베이스 서버 보안 인증, 특히 관리자 사용자 이름과 암호
- 서버리스 클러스터의 크기 조정. Amazon Aurora 이는 예상 클러스터 로드 에 따라 달라집니다.
- 네트워킹 파라미터, 특히 서브넷 생성을 위한 대상 VPC와 서브넷 또는 CIDR 블록

데이터베이스 서버에 적합한 보안 인증과 크기를 선택합니다. 네트워킹 옵션의 경우 AWS ParallelCluster 클러스터가 배포된 것과 동일한 VPC를 사용해야 합니다. 데이터베이스의 서브넷을 생성하여 템플릿의 입력으로 전달할 수 있습니다. 또는 두 서브넷에 대해 분리된 CIDR 블록 두 개를 제공하고 CloudFormation 템플릿에서 CIDR 블록을 위한 두 개의 서브넷을 생성하도록 하십시오. CIDR 블록이 기존 서브넷과 겹치지 않는지 확인하세요. CIDR 블록이 기존 서브넷과 겹치는 경우 스택이 생성되지 않습니다.

데이터베이스 서버를 생성하는 데 몇 분 정도 걸립니다.

## 3단계: Slurm 회계가 활성화된 클러스터 생성

제공된 CloudFormation 템플릿은 일부 정의된 출력이 포함된 CloudFormation 스택을 생성합니다. 에서 CloudFormation 스택 뷰의 출력 탭에서 출력을 볼 수 있습니다. AWS Management Console Slurm 회계를 활성화하려면 AWS ParallelCluster 클러스터 구성 파일에서 다음 출력 중 일부를 사용해야 합니다.

- DatabaseHost: [SlurmSettings/Database/Uri](#) 클러스터 구성 파라미터에 사용됩니다.
- DatabaseAdminUser: [SlurmSettings/Database/UserName](#) 클러스터 구성 파라미터 값에 사용됩니다.
- DatabaseSecretArn: [SlurmSettings/Database/PasswordSecretArn](#) 클러스터 구성 파라미터에 사용됩니다.

- DatabaseClientSecurityGroup: [HeadNode/Networking/SecurityGroups](#) 구성 파라미터에 정의된 클러스터의 헤드 노드에 연결된 보안 그룹입니다.

클러스터 구성 파일 Database 파라미터를 출력 값으로 업데이트하세요. [pcluster](#) CLI를 사용하여 클러스터를 생성합니다.

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

클러스터를 생성한 후 `sacctmgr` 또는 `sacct` 와 같은 Slurm 회계 명령을 사용할 수 있습니다.

## 외부 Slurmdbd 어카운팅을 사용하여 클러스터 생성

외부 Slurmdbd 계정을 사용하여 클러스터를 구성하고 생성하는 방법을 알아보십시오. 자세한 내용은 [Slurm어카운팅](#)을 참조하십시오. AWS ParallelCluster

AWS ParallelCluster 명령줄 인터페이스 (CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 내용은 [에서 사용하는AWSAWS ParallelCluster서비스](#)를 참조하십시오.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS Free Tier 카테고리 내에서 사용할 수 있습니다. 자세한 내용은 [AWS ParallelCluster UI 비용](#)을 참조하십시오.

이 자습서에서는 AWS CloudFormation 빠른 생성 템플릿을 사용하여 클러스터와 동일한 VPC에 Slurmdbd 인스턴스를 배포하는 데 필요한 구성 요소를 생성합니다. 템플릿은 클러스터와 데이터베이스 간 연결을 위한 기본 네트워킹 및 보안 구성을 생성합니다.

### Note

부터 시작하여 클러스터 구성 `version 3.10.0` 매개 변수를 사용하여 외부 Slurmdbd를 AWS ParallelCluster 지원합니다. `SlurmSettings / ExternalSlurmdbd`

### Note

빠른 생성 템플릿이 그 예시입니다. 이 템플릿은 가능한 모든 사용 사례를 다루지는 않습니다. 프로덕션 워크로드에 적합한 구성과 용량을 갖춘 외부 Slurmdbd를 만드는 것은 사용자의 책임입니다.

## 사전 조건:

- AWS ParallelCluster [설치되었습니다.](#)
- AWS CLI [가 설치 및 구성되었습니다.](#)
- [Amazon Elastic Compute Cloud 키 페어가](#) 있습니다.
- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 AWS Identity and Access Management 역할이 있습니다.
- Slurm계정 데이터베이스가 있습니다. Slurm계정 데이터베이스 생성 튜토리얼을 단계별로 진행하려면 [Slurm Accounting Database 스택 만들기의](#) 1단계와 2단계를 따르십시오.

## 1단계: Slurmdbd 스택 생성

이 자습서에서는 [CloudFormation 빠른 생성 템플릿 \(us-east-1\) 을 사용하여 Slurmdbd 스택을 생성합니다.](#) 템플릿에는 다음과 같은 입력이 필요합니다.

### 네트워킹

- vPCID: Slurmdbd 인스턴스를 시작하기 위한 VPC ID입니다.
- SubnetId: Slurmdbd 인스턴스를 시작하기 위한 서브넷 ID입니다.
- PrivatePrefix: VPC의 CIDR 접두사입니다.
- PrivateIp: Slurmdbd 인스턴스에 할당할 보조 프라이빗 IP.

### 데이터베이스 연결

- DBMsClientsG: Slurmdbd 인스턴스에 연결할 보안 그룹입니다. 이 보안 그룹은 데이터베이스 서버와 Slurmdbd 인스턴스 간의 연결을 허용해야 합니다.
- DBMS DatabaseName: 데이터베이스의 이름.
- DBMS 사용자 이름: 데이터베이스의 사용자 이름.
- DBMS PasswordSecretArn: 데이터베이스 비밀번호가 포함된 비밀번호입니다.
- DBMsuri: 데이터베이스 서버의 URI입니다.

### 인스턴스 설정

- InstanceType: slurmdbd 인스턴스에 사용할 인스턴스 유형입니다.
- KeyName: slurmdbd 인스턴스에 사용할 Amazon EC2 키 페어.



## Slurmdbd 설정

- AMID: Slurmdbd 인스턴스의 AMI입니다. AMI는 ParallelCluster AMI여야 합니다. ParallelCluster AMI 버전에 따라 Slurmdbd 버전이 결정됩니다.
- MungeKeySecretArn: Slurmdbd와 클러스터 간의 통신을 인증하는 데 사용할 Munge 키가 포함된 비밀입니다.
- SlurmdbdPort: slurmdbd가 사용하는 포트 번호입니다.
- EnableSlurmdbdSystemService: slurmdbd를 시스템 서비스로 활성화하고 인스턴스 시작 시 실행되도록 합니다.

### ⚠ Warning

다른 버전의 SlurmDB 에서 데이터베이스를 생성한 경우 시스템 Slurmdbd 서비스로 사용하지 마십시오.

데이터베이스에 많은 항목이 포함되어 있는 경우 데이터베이스를 업데이트하는 데 수십 분이 걸리고 이 기간 동안에는 응답하지 Slurm Database Daemon (SlurmDBD) 않을 수 있습니다. SlurmDB업그레이드하기 전에 데이터베이스를 백업하십시오. 자세한 내용은 [Slurm 설명서](#)를 참조하십시오.

## 2단계: 외부 Slurmdbd 기능이 활성화된 클러스터 생성

제공된 AWS CloudFormation 템플릿은 일부 정의된 AWS CloudFormation 출력이 포함된 스택을 생성합니다.

에서 AWS CloudFormation 스택의 AWS Management Console출력 탭을 확인하여 생성된 엔티티를 검토하십시오. Slurm어카운팅을 활성화하려면 AWS ParallelCluster 구성 파일에 다음과 같은 출력 중 일부를 사용해야 합니다.

- SlurmdbdPrivatelp: [SlurmSettings/ExternalSlurmdbd/호스트](#) 클러스터 구성 매개 변수에 사용됩니다.
- SlurmdbdPort: [SlurmSettings/ExternalSlurmdbd/Port](#) 클러스터 구성 매개 변수 값에 사용됩니다.
- AccountingClientSecurityGroup: [네트워킹 HeadNode/구성 매개 변수에 정의된 클러스터의 헤드 노드에 연결된 보안 그룹](#)입니다. [AdditionalSecurityGroups](#)

추가로, 스택 뷰의 파라미터 탭에서: AWS CloudFormation

- MungeKeySecretArn: [SlurmSettings/MungeKeySecretArn](#) 클러스터 구성 파라미터 값에 사용됩니다.

클러스터 구성 파일 데이터베이스 매개변수를 출력 값으로 업데이트하십시오. pcluster를 사용하여 클러스터를 생성합니다 AWS CLI .

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

클러스터를 생성한 후 sacctmgr 또는 sacct 와 같은 Slurm 회계 명령을 사용할 수 있습니다.

#### Warning

외부 ParallelCluster 트래픽과 외부 간의 SlurmDB 트래픽은 암호화되지 않습니다. 클러스터와 외부 네트워크는 신뢰할 수 SlurmDB 있는 네트워크에서 실행하는 것이 좋습니다.

## 이전 AWS Systems Manager 문서 버전으로 되돌리기

이전 AWS Systems Manager 문서 버전으로 되돌리는 방법을 알아봅니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 [AWS Systems Manager 문서](#)를 참조하세요.

AWS ParallelCluster 명령줄 인터페이스(CLI) 또는 API를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 내용은 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#) 항목을 참조하세요.

AWS ParallelCluster UI는 서버리스 아키텍처를 기반으로 하며 대부분의 경우 AWS 프리 티어 범주 내에서 사용할 수 있습니다. 자세한 내용은 [AWS ParallelCluster UI 비용](#) 항목을 참조하세요.

사전 조건:

- SSM 문서를 관리할 수 권한이 있는 AWS 계정이 있습니다.
- AWS CLI가 [설치 및 구성되어 있습니다](#).

## 이전 SSM 문서 버전으로 되돌리기

1. 터미널에서 다음 명령을 실행하여 소유하고 있는 기존 SSM 문서 목록을 가져옵니다.

```
$ aws ssm list-documents --document-filter "key=Owner,value=Self"
```

2. SSM 문서를 이전 버전으로 되돌립니다. 이 예에서는 SessionManagerRunShell 문서의 이전 버전으로 되돌립니다. SSM SessionManagerRunShell 문서를 사용하여 시작하는 모든 SSM 셸 세션을 사용자 지정할 수 있습니다.
  - a. 다음 명령어를 실행하여 SessionManagerRunShell을 위한 DocumentVersion 파라미터를 찾으세요.

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {
        "Name": "linuxcmd",
        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then
source /opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/
bash"
      }
    ],
    "PlatformTypes": [
      "Windows",
      "Linux",
      "MacOS"
    ],
    "DocumentType": "Session",
    "SchemaVersion": "1.0",
    "LatestVersion": "2",
    "DefaultVersion": "1",
    "DocumentFormat": "JSON",
    "Tags": []
  }
}
```

```
}
```

최신 버전은 2입니다.

- b. 다음 명령을 실행하여 이전 버전으로 되돌립니다.

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-version 2
```

3. describe-document 명령을 다시 실행하여 문서 버전이 되돌려졌는지 확인합니다.

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
  "Document": {
    "Hash": "...",
    "HashType": "Sha256",
    "Name": "SSM-SessionManagerRunShell",
    "Owner": "123456789012",
    "CreateDate": "2023-02-20T19:04:32.390000+00:00",
    "Status": "Active",
    "DocumentVersion": "1",
    "Parameters": [
      {
        "Name": "linuxcmd",
        "Type": "String",
        "Description": "The command to run on connection...",
        "DefaultValue": "if [ -d '/opt/parallelcluster' ]; then source /
opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/bash"
      }
    ],
    "PlatformTypes": [
      "Windows",
      "Linux",
      "MacOS"
    ],
    "DocumentType": "Session",
    "SchemaVersion": "1.0",
    "LatestVersion": "1",
    "DefaultVersion": "1",
    "DocumentFormat": "JSON",
    "Tags": []
  }
}
```

최신 버전은 1입니다.

## 를 사용하여 클러스터 생성 AWS CloudFormation

AWS ParallelCluster CloudFormation 사용자 지정 리소스로 클러스터를 만드는 방법을 알아보세요. 자세한 정보는 [AWS CloudFormation 사용자 지정 리소스](#)을 참조하세요.

사용 AWS ParallelCluster시 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [AWS 에서 사용하는 서비스 AWS ParallelCluster](#)을 참조하세요.

사전 조건:

- [설치 및 구성되었습니다. AWS CLI](#)
- [아마존 EC2 키 페어](#).
- [pcluster](#) CLI를 실행하는 데 필요한 [권한](#)을 가진 IAM 역할

## CloudFormation 빠른 생성 스택을 통한 클러스터 생성

이 자습서에서는 빠른 생성 스택을 사용하여 클러스터를 생성하는 CloudFormation 템플릿과 다음 리소스를 배포합니다. AWS

- CloudFormation 빠른 생성 CloudFormation 스택을 사용하여 만든 루트 스택입니다.
- 기본 정책, 기본 VPC 설정, 사용자 지정 리소스 공급자를 포함하는 중첩된 CloudFormation 스택입니다.
- 로그인하고 작업을 실행할 수 있는 AWS ParallelCluster 클러스터 스택과 클러스터의 예

를 사용하여 클러스터를 생성합니다. AWS CloudFormation

1. AWS Management Console에 로그인합니다.
2. CloudFormation [빠른 생성 링크를 열어 콘솔에서](#) 다음 CloudFormation 리소스를 생성합니다.
  - 클러스터 헤드 노드와 컴퓨팅 노드를 각각 실행하기 위한 퍼블릭 서브넷과 프라이빗 서브넷이 있는 VPC가 있는 중첩된 CloudFormation 스택입니다.
  - 클러스터 관리를 위한 AWS ParallelCluster 사용자 지정 리소스가 있는 중첩된 CloudFormation 스택입니다.

- 클러스터 관리를 위한 기본 정책이 포함된 중첩된 CloudFormation 스택입니다.
- 중첩된 CloudFormation 스택의 루트 스택입니다.
- Slurm스케줄러와 정의된 수의 컴퓨팅 노드가 있는 AWS ParallelCluster 클러스터.

CloudFormation > Stacks > Create stack

## Quick create stack

**Template**

Template URL  
[https://pcluster-cfn-us-east-2.s3.amazonaws.com/parallelcluster/3.5.0/templates/custom\\_resource/cluster-1-click.yaml](https://pcluster-cfn-us-east-2.s3.amazonaws.com/parallelcluster/3.5.0/templates/custom_resource/cluster-1-click.yaml)

Stack description  
 AWS ParallelCluster CloudFormation Cluster

**Stack name**

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**AvailabilityZone**  
 Availability zone where instances will be launched

**KeyName**  
 KeyPair to login to the head node

**Capabilities**

**i** **The following resource(s) require capabilities: [AWS::CloudFormation::Stack]**

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: {0}. Check the capabilities of these resources. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY\_AUTO\_EXPAND

Cancel

3. 빠른 스택 생성 파라미터 섹션에서 다음 파라미터의 값을 입력합니다.
  - a. 예 KeyNameAmazon EC2 키 페어의 이름을 입력합니다.
  - b. 의 AvailabilityZone경우 클러스터 노드의 AZ를 선택합니다 (예:). us-east-1a
4. 페이지 하단에서 각 액세스 기능이 필요함을 확인하는 확인란을 선택합니다.
5. 스택 생성을 선택합니다.
6. CloudFormation 스택이 CREATE\_COMPLETE 상태에 도달할 때까지 기다리십시오.

## AWS CloudFormation 명령줄 인터페이스 (CLI) 를 사용한 클러스터 생성

이 자습서에서는 CLI ( AWS 명령줄 인터페이스) CloudFormation 를 사용하여 클러스터를 생성하는 CloudFormation 템플릿을 배포합니다.

다음 AWS 리소스를 생성하십시오.

- CloudFormation 빠른 생성 CloudFormation 스택을 사용하여 만든 루트 스택입니다.
- 기본 정책, 기본 VPC 설정, 사용자 지정 리소스 공급자를 포함하는 중첩된 CloudFormation 스택입니다.
- 로그인하고 작업을 실행할 수 있는 AWS ParallelCluster 클러스터 스택과 클러스터의 예

##### ## ### ##(예: ###)을 자체 값으로 바꿉니다.

를 사용하여 클러스터를 생성합니다. AWS CloudFormation

1. 다음 cluster\_template.yaml 내용으로 이름이 지정된 CloudFormation 템플릿을 생성하십시오.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
  KeyName:
    Description: KeyPair to login to the head node
    Type: AWS::EC2::KeyPair::KeyName

  AvailabilityZone:
    Description: Availability zone where instances will be launched
    Type: AWS::EC2::AvailabilityZone::Name
    Default: us-east-2a
```

```
Mappings:
  ParallelCluster:
    Constants:
      Version: 3.7.0

Resources:
  PclusterClusterProvider:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
        - { Version: !FindInMap [ParallelCluster, Constants, Version] }

  PclusterVpc:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        PublicCIDR: 10.0.0.0/24
        PrivateCIDR: 10.0.16.0/20
        AvailabilityZone: !Ref AvailabilityZone
      TemplateURL: !Sub
        - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
          ${AWS::URLSuffix}/parallelcluster/${Version}/templates/networking/public-private-
          ${Version}.cfm.json
        - { Version: !FindInMap [ParallelCluster, Constants, Version] }

  PclusterCluster:
    Type: Custom::PclusterCluster
    Properties:
      ServiceToken: !GetAtt [ PclusterClusterProvider , Outputs.ServiceToken ]
      ClusterName: !Sub 'c-${AWS::StackName}'
      ClusterConfiguration:
        Image:
          Os: alinux2
        HeadNode:
          InstanceType: t2.medium
          Networking:
            SubnetId: !GetAtt [ PclusterVpc , Outputs.PublicSubnetId ]
          Ssh:
            KeyName: !Ref KeyName
        Scheduling:
          Scheduler: slurm
```



```

SlurmQueues:
- Name: queue0
  ComputeResources:
  - Name: queue0-cr0
    InstanceType: t2.micro
  Networking:
  SubnetIds:
  - !GetAtt [ PclusterVpc , Outputs.PrivateSubnetId ]

Outputs:
  HeadNodeIp:
    Description: The Public IP address of the HeadNode
    Value: !GetAtt [ PclusterCluster, headNode.publicIpAddress ]
    
```

2. 다음 AWS CLI 명령을 실행하여 클러스터 생성 및 관리를 위한 CloudFormation 스택을 배포합니다.

```

$ aws cloudformation deploy --template-file ./cluster_template.yaml \
  --stack-name mycluster \
  --parameter-overrides KeyName=keypair \
    AvailabilityZone=us-east-2b \
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
    
```

## CloudFormation 클러스터 출력 보기

CloudFormation 클러스터 출력을 보고 유용한 클러스터 세부 정보를 얻을 수 있습니다. 추가된 ValidationMessages 속성은 클러스터 생성 및 업데이트 작업의 검증 메시지에 대한 액세스를 제공합니다.

1. [CloudFormation 콘솔로](#) 이동하여 사용자 AWS ParallelCluster 지정 리소스가 포함된 스택을 선택합니다.
2. 스택 세부 정보를 선택하고 출력 탭을 선택합니다.

| Key                | Value                                                                                                                                                                                                                 | Description                                            |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| HeadNodeIp         | 1.2.3.4                                                                                                                                                                                                               | The Public IP address of the HeadNode                  |
| ValidationMessages | [{"level": "WARNING", "type": "KeyPairValidator", "message": "If you do not specify a key pair, you can't connect to the instance unless you choose an AMI that is configured to allow users another way to log in"}] | Any warnings from cluster create or update operations. |

검증 메시지는 잘릴 수 있습니다. 로그 검색에 대한 자세한 내용은 [AWS ParallelCluster 문제 해결](#) 섹션을 참조하세요.

## 클러스터에 액세스

클러스터에 액세스하세요.

클러스터 헤드 노드에 **ssh**하세요.

1. CloudFormation 스택 배포가 완료되면 다음 명령을 사용하여 헤드 노드의 IP 주소를 가져옵니다.

```
$ HEAD_NODE_IP=$(aws cloudformation describe-stacks --stack-name=mycluster --query "Stacks|[0].Outputs[?OutputKey=='HeadNodeIp']|[0].OutputValue" --output=text)
```

CloudFormation 콘솔의 클러스터 스택 출력 탭에 있는 HeadNodeIp 파라미터에서 헤드 노드 IP 주소를 검색할 수도 있습니다.

헤드 노드 IP 주소는 클러스터 CloudFormation 템플릿의 Outputs 섹션, 특히 이 예제 클러스터의 경우 추가되었으므로 여기에서 찾을 수 있습니다.

2. 다음 명령을 실행하여 클러스터 헤드 노드에 연결합니다.

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

## 정리

클러스터를 삭제합니다.

1. 다음 AWS CLI 명령을 실행하여 CloudFormation 스택과 클러스터를 삭제합니다.

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. 다음 명령을 실행하여 스택 삭제 상태를 확인합니다.

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

## 테라폼으로 ParallelCluster API 배포하기

이 자습서에서는 API를 배포하기 위한 간단한 Terraform 프로젝트를 정의합니다. ParallelCluster

### 사전 조건

- 테라폼 v1.5.7+가 설치되었습니다.
- API를 배포할 권한이 있는 IAM 역할 ParallelCluster [the section called “필요한 권한”](#) 섹션을 참조하십시오.

### 테라폼 프로젝트 정의

1. 라는 디렉토리를 생성합니다. my-pcluster-api

생성하는 모든 파일은 이 디렉터리 내에 있습니다.

2. 파일을 provider.tf 생성하여 AWS 공급자를 구성합니다.

```
provider "aws" {
  region = var.region
  profile = var.profile
}
```

3. ParallelCluster모듈을 사용하여 리소스를 정의하는 파일을 main.tf 생성합니다.

```
module "parallelcluster_pcluster_api" {
  source = "aws-tf/parallelcluster/aws//modules/pcluster_api"
  version = "1.0.0"

  region          = var.region
  api_stack_name  = var.api_stack_name
  api_version     = var.api_version
  deploy_pcluster_api = true
  parameters = {
    EnableIamAdminAccess = "true"
  }
}
```

4. 파일을 variables.tf 생성하여 이 프로젝트에 삽입할 수 있는 변수를 정의합니다.

```
variable "region" {
```

```

description = "The region the ParallelCluster API is deployed in."
type        = string
default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

variable "api_stack_name" {
  type        = string
  description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
  default     = "ParallelCluster"
}

variable "api_version" {
  type        = string
  description = "The version of the ParallelCluster API."
}

```

5. terraform.tfvars 파일을 만들어 변수에 임의의 값을 설정합니다.

아래 파일은 스택 이름을 us-east-1 사용하여 ParallelCluster API 3.10.0을 배포합니다. MyParallelClusterAPI-310 스택 이름을 사용하여 이 ParallelCluster API 배포를 참조할 수 있습니다.

```

region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

```

6. 파일을 outputs.tf 생성하여 이 프로젝트에서 반환되는 출력을 정의합니다.

```

output "pcluster_api_stack_outputs" {
  value = module.parallelcluster_pcluster_api.stack_outputs
}

```

프로젝트 디렉토리는 다음과 같습니다.

```
my-pcluster-api
```

```

### main.tf - Terraform entrypoint to define the resources using the
ParallelCluster module.
### outputs.tf - Defines the outputs returned by Terraform.
### providers.tf - Configures the AWS provider.
### terraform.tfvars - Set the arbitrary values for the variables, i.e. region,
PCAPI version, PCAPI stack name
### variables.tf - Defines the variables, e.g. region, PCAPI version, PCAPI stack
name.

```

## API 배포

API를 배포하려면 표준 Terraform 명령을 순서대로 실행하십시오.

### 1. 프로젝트 빌드:

```
terraform init
```

### 2. 배포 계획을 정의하세요.

```
terraform plan -out tfplan
```

### 3. 계획 배포:

```
terraform apply tfplan
```

## 필요한 권한

Terraform으로 ParallelCluster API를 배포하려면 다음 권한이 필요합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:GetTemplate"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormationRead"
    }
  ]
}

```

```

    },
    {
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation>DeleteStack",
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/
MyParallelClusterAPI*",
      "Effect": "Allow",
      "Sid": "CloudFormationWrite"
    },
    {
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": [
        "arn:PARTITION:cloudformation:REGION:aws:transform/Include",
        "arn:PARTITION:cloudformation:REGION:aws:transform/
Serverless-2016-10-31"
      ],
      "Effect": "Allow",
      "Sid": "CloudFormationTransformWrite"
    },
    {
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:PARTITION:s3::*-aws-parallelcluster/parallelcluster/*/api/
ParallelCluster.openapi.yaml",
        "arn:PARTITION:s3::*-aws-parallelcluster/parallelcluster/*/layers/aws-
parallelcluster/lambda-layer.zip"
      ],
      "Effect": "Allow",
      "Sid": "S3ParallelClusterArtifacts"
    },
    {
      "Action": [
        "iam:CreateRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:CreatePolicy",
        "iam>DeletePolicy",

```

```

        "iam:GetPolicy",
        "iam:GetRolePolicy",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy",
        "iam:ListPolicyVersions"
    ],
    "Resource": [
        "arn:PARTITION:iam::ACCOUNT:role/*",
        "arn:PARTITION:iam::ACCOUNT:policy/*"
    ],
    "Effect": "Allow",
    "Sid": "IAM"
},
{
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:PARTITION:iam::ACCOUNT:role/ParallelClusterLambdaRole-*",
        "arn:PARTITION:iam::ACCOUNT:role/APIGatewayExecutionRole-*"
    ],
    "Effect": "Allow",
    "Sid": "IAMPassRole"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda>DeleteFunction",
        "lambda:GetFunction",
        "lambda:PublishLayerVersion",
        "lambda>DeleteLayerVersion",
        "lambda:GetLayerVersion",
        "lambda:TagResource",
        "lambda:UntagResource"
    ],
    "Resource": [
        "arn:PARTITION:lambda:REGION:ACCOUNT:layer:PCLayer-*",
        "arn:PARTITION:lambda:REGION:ACCOUNT:function:*-
ParallelClusterFunction-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
}

```

```

    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs>DeleteLogGroup",
        "logs:DescribeLogGroups",
        "logs:PutRetentionPolicy",
        "logs:TagLogGroup",
        "logs:UntagLogGroup"
      ],
      "Resource": [
        "arn:PARTITION:logs:REGION:ACCOUNT:log-group:/aws/lambda/*-
ParallelClusterFunction-*"
      ],
      "Effect": "Allow",
      "Sid": "Logs"
    },
    {
      "Action": [
        "apigateway:DELETE",
        "apigateway:GET",
        "apigateway:PATCH",
        "apigateway:POST",
        "apigateway:PUT",
        "apigateway:UpdateRestApiPolicy"
      ],
      "Resource": [
        "arn:PARTITION:apigateway:REGION::/restapis",
        "arn:PARTITION:apigateway:REGION::/restapis/*",
        "arn:PARTITION:apigateway:REGION::/tags/*"
      ],
      "Effect": "Allow",
      "Sid": "APIGateway"
    }
  ]
}

```

## 테라폼으로 클러스터 만들기

사용 AWS ParallelCluster시 AWS ParallelCluster 이미지와 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [the section called “AWS 에서 사용하는 서비스 AWS ParallelCluster”](#)을 참조하세요.



## 사전 조건

- 테라폼 v1.5.7+가 설치되어 있습니다.
- [the section called “AWS ParallelCluster API”v3.8.0+](#)가 계정에 배포되었습니다. [the section called “테라폼으로 ParallelCluster API 배포하기”](#) 섹션을 참조하십시오.
- API를 호출할 권한이 있는 IAM 역할. ParallelCluster [필수 권한] 을 참조하십시오.

## 테라폼 프로젝트 정의

이 튜토리얼에서는 클러스터를 배포하기 위한 간단한 Terraform 프로젝트를 정의합니다.

1. 라는 디렉터리를 생성합니다. my-clusters

생성하는 모든 파일은 이 디렉터리 내에 있습니다.

2. 파일을 terraform.tf 생성하여 ParallelCluster 제공자를 가져오십시오.

```
terraform {
  required_version = ">= 1.5.7"
  required_providers {
    aws-parallelcluster = {
      source = "aws-tf/aws-parallelcluster"
      version = "1.0.0"
    }
  }
}
```

3. 파일을 providers.tf 생성하여 ParallelCluster 및 AWS 제공자를 구성합니다.

```
provider "aws" {
  region = var.region
  profile = var.profile
}

provider "aws-parallelcluster" {
  region          = var.region
  profile         = var.profile
  api_stack_name = var.api_stack_name
  use_user_role   = true
}
```

4. ParallelCluster모듈을 사용하여 리소스를 정의하는 파일을 main.tf 생성합니다.

```

module "pcluster" {
  source = "aws-tf/parallelcluster/aws"
  version = "1.0.0"

  region          = var.region
  api_stack_name  = var.api_stack_name
  api_version     = var.api_version
  deploy_pcluster_api = false

  template_vars      = local.config_vars
  cluster_configs    = local.cluster_configs
  config_path        = "config/clusters.yaml"
}

```

5. 파일을 `clusters.tf` 생성하여 여러 클러스터를 Terraform 로컬 변수로 정의합니다.

#### Note

요소 내에 여러 클러스터를 정의할 수 있습니다. `cluster_config` 모든 군집에 대해 지역 변수 내에서 군집 속성을 명시적으로 정의 (참조 `DemoCluster01`) 하거나 외부 파일을 참조 (참조) 할 수 있습니다. `DemoCluster02`

구성 요소 내에서 설정할 수 있는 클러스터 속성을 검토하려면 을 참조하십시오. [the section called “클러스터 구성 파일”](#)

클러스터 생성을 위해 설정할 수 있는 옵션을 검토하려면 을 참조하십시오 [the section called “pcluster create-cluster”](#).

```

locals {
  cluster_configs = {
    DemoCluster01 : {
      region : local.config_vars.region
      rollbackOnFailure : false
      validationFailureLevel : "WARNING"
      suppressValidators : [
        "type:KeyPairValidator"
      ]
      configuration : {
        Region : local.config_vars.region
        Image : {

```

```
    Os : "alinux2"
  }
  HeadNode : {
    InstanceType : "t3.small"
    Networking : {
      SubnetId : local.config_vars.subnet
    }
    Iam : {
      AdditionalIamPolicies : [
        { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
      ]
    }
  }
  Scheduling : {
    Scheduler : "slurm"
    SlurmQueues : [{
      Name : "queue1"
      CapacityType : "ONDEMAND"
      Networking : {
        SubnetIds : [local.config_vars.subnet]
      }
      Iam : {
        AdditionalIamPolicies : [
          { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
        ]
      }
      ComputeResources : [{
        Name : "compute"
        InstanceType : "t3.small"
        MinCount : "1"
        MaxCount : "4"
      }]
    }]
    SlurmSettings : {
      QueueUpdateStrategy : "TERMINATE"
    }
  }
}
DemoCluster02 : {
  configuration : "config/cluster_config.yaml"
}
}
```

```
}

```

6. 파일을 `config/clusters.yaml` 생성하여 여러 클러스터를 YAML 구성으로 정의합니다.

```
DemoCluster03:
  region: ${region}
  rollbackOnFailure: true
  validationFailureLevel: WARNING
  suppressValidators:
    - type:KeyPairValidator
  configuration: config/cluster_config.yaml
DemoCluster04:
  region: ${region}
  rollbackOnFailure: false
  configuration: config/cluster_config.yaml

```

7. Terraform 변수를 삽입할 수 있는 표준 ParallelCluster 구성 파일인 파일을 생성합니다. `config/cluster_config.yaml`

구성 요소 내에서 설정할 수 있는 클러스터 속성을 검토하려면 [the section called “클러스터 구성 파일”](#) 을 참조하십시오.

```
Region: ${region}
Image:
  Os: alinux2
HeadNode:
  InstanceType: t3.small
Networking:
  SubnetId: ${subnet}
Iam:
  AdditionalIamPolicies:
    - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
Scheduling:
  Scheduler: slurm
SlurmQueues:
  - Name: queue1
    CapacityType: ONDEMAND
    Networking:
      SubnetIds:
        - ${subnet}
    Iam:
      AdditionalIamPolicies:
        - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore

```

```

ComputeResources:
  - Name: compute
    InstanceType: t3.small
    MinCount: 1
    MaxCount: 5
SlurmSettings:
  QueueUpdateStrategy: TERMINATE

```

8. 파일을 `clusters_vars.tf` 생성하여 클러스터 구성에 삽입할 수 있는 변수를 정의합니다.

이 파일을 사용하면 지역 및 서브넷과 같은 클러스터 구성에서 사용할 수 있는 동적 값을 정의할 수 있습니다.

이 예제는 프로젝트 변수에서 직접 값을 검색하지만 값을 결정하려면 사용자 지정 로직을 사용해야 할 수도 있습니다.

```

locals {
  config_vars = {
    subnet = var.subnet_id
    region = var.cluster_region
  }
}

```

9. 파일을 `variables.tf` 생성하여 이 프로젝트에 삽입할 수 있는 변수를 정의합니다.

```

variable "region" {
  description = "The region the ParallelCluster API is deployed in."
  type        = string
  default     = "us-east-1"
}

variable "cluster_region" {
  description = "The region the clusters will be deployed in."
  type        = string
  default     = "us-east-1"
}

variable "profile" {
  type        = string
  description = "The AWS profile used to deploy the clusters."
  default     = null
}

```

```

variable "subnet_id" {
  type      = string
  description = "The id of the subnet to be used for the ParallelCluster
instances."
}

variable "api_stack_name" {
  type      = string
  description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
  default   = "ParallelCluster"
}

variable "api_version" {
  type      = string
  description = "The version of the ParallelCluster API."
}

```

10. terraform.tfvars 파일을 만들어 변수에 임의의 값을 설정합니다.

아래 파일은 스택 이름과 함께 이미 배포된 기존 ParallelCluster API 3.10.0을 사용하여 서브넷 eu-west-1 subnet-123456789 내에 클러스터를 배포합니다. us-east-1 MyParallelClusterAPI-310

```

region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

cluster_region = "eu-west-1"
subnet_id = "subnet-123456789"

```

11. 파일을 outputs.tf 생성하여 이 프로젝트에서 반환되는 출력을 정의합니다.

```

output "clusters" {
  value = module.pcluster.clusters
}

```

프로젝트 디렉토리는 다음과 같습니다.

```

my-clusters
### config

```

```
#   ### cluster_config.yaml - Cluster configuration, where terraform variables can
#   be injected..
#   ### clusters.yaml - File listing all the clusters to deploy.
### clusters.tf - Clusters defined as Terraform local variables.
### clusters_vars.tf - Variables that can be injected into cluster configurations.
### main.tf - Terraform entrypoint where the ParallelCluster module is configured.
### outputs.tf - Defines the cluster as a Terraform output.
### providers.tf - Configures the providers: ParallelCluster and AWS.
### terraform.tf - Import the ParallelCluster provider.
### terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
### variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

## 클러스터 배포

클러스터를 배포하려면 표준 Terraform 명령을 순서대로 실행합니다.

### Note

이 예시에서는 계정에 ParallelCluster API를 이미 배포했다고 가정합니다.

#### 1. 프로젝트 빌드:

```
terraform init
```

#### 2. 배포 계획 정의:

```
terraform plan -out tfplan
```

#### 3. 계획 배포:

```
terraform apply tfplan
```

## 클러스터와 함께 ParallelCluster API 배포

ParallelCluster API를 배포하지 않았는데 클러스터와 함께 배포하려는 경우 다음 파일을 변경하십시오.

- main.tf

```

module "pcluster" {
  source = "aws-tf/aws/parallelcluster"
  version = "1.0.0"

  region          = var.region
  api_stack_name  = var.api_stack_name
  api_version     = var.api_version
  deploy_pcluster_api = true

  template_vars      = local.config_vars
  cluster_configs    = local.cluster_configs
  config_path        = "config/clusters.yaml"
}

```

#### • providers.tf

```

provider "aws-parallelcluster" {
  region = var.region
  profile = var.profile
  endpoint = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiInvokeUrl
  role_arn = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiUserRole
}

```

## 필요한 권한

Terraform으로 클러스터를 배포하려면 다음 권한이 필요합니다.

- API와의 상호 작용을 담당하는 ParallelCluster API 역할을 맡으십시오. ParallelCluster
- ParallelCluster API AWS CloudFormation 스택을 설명하여 존재하는지 확인하고 해당 매개변수와 출력을 검색합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:PARTITION:iam::ACCOUNT:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    }
  ]
}

```



```

    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}

```

## 테라폼으로 사용자 지정 AMI 생성

사용 AWS ParallelCluster시 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 정보는 [the section called “AWS 에서 사용하는 서비스 AWS ParallelCluster”](#)을 참조하세요.

### 사전 조건

- 테라폼 v1.5.7+가 설치되어 있습니다.
- [the section called “AWS ParallelCluster API”](#)v3.8.0+가 계정에 배포되었습니다. [the section called “테라폼으로 클러스터 만들기”](#) 섹션을 참조하십시오.
- API를 호출할 권한이 있는 IAM 역할 ParallelCluster [the section called “필요한 권한”](#) 섹션을 참조하십시오.

## 테라폼 프로젝트 정의

이 자습서에서는 간단한 Terraform 프로젝트를 정의하여 사용자 지정 ParallelCluster AMI를 배포합니다.

1. 라는 디렉터리를 생성합니다. my-amis

생성하는 모든 파일은 이 디렉터리 내에 있습니다.

2. 파일을 terraform.tf 생성하여 ParallelCluster 제공자를 가져오십시오.

```

terraform {
  required_version = ">= 1.5.7"
  required_providers {

```

```

aws-parallelcluster = {
  source = "aws-tf/aws-parallelcluster"
  version = "1.0.0"
}
}
}

```

3. 파일을 `providers.tf` 생성하여 ParallelCluster 및 AWS 제공자를 구성합니다.

```

provider "aws" {
  region = var.region
  profile = var.profile
}

provider "aws-parallelcluster" {
  region          = var.region
  profile         = var.profile
  api_stack_name = var.api_stack_name
  use_user_role  = true
}

```

4. ParallelCluster모듈을 사용하여 리소스를 정의하는 파일을 `main.tf` 생성합니다.

`image_configuration`요소 내에서 설정할 수 있는 이미지 속성을 검토하려면 [the section called “빌드 이미지 구성 파일”](#)을 참조하십시오.

이미지 생성을 위해 설정할 수 있는 옵션 (예: `image_id` 및) 을 `rollback_on_failure` 검토하려면 [the section called “pcluster build-image”](#)을 참조하십시오.

```

data "aws-parallelcluster_list_official_images" "parent_image" {
  region = var.region
  os     = var.os
  architecture = var.architecture
}

resource "aws-parallelcluster_image" "demo01" {
  image_id = "demo01"
  image_configuration = yamlencode({
    "Build":{
      "InstanceType": "c5.2xlarge",
      "ParentImage": data.aws-parallelcluster_list_official_images.parent_image.official_images[0].amiId,
      "UpdateOsPackages": {"Enabled": false}
    }
  })
}

```

```
    }  
  })  
  rollback_on_failure = false  
}
```

5. 파일을 `variables.tf` 만들어 이 프로젝트에 삽입할 수 있는 변수를 정의하십시오.

```
variable "region" {  
  description = "The region the ParallelCluster API is deployed in."  
  type        = string  
  default     = "us-east-1"  
}  
  
variable "profile" {  
  type        = string  
  description = "The AWS profile used to deploy the clusters."  
  default     = null  
}  
  
variable "api_stack_name" {  
  type        = string  
  description = "The name of the CloudFormation stack used to deploy the  
ParallelCluster API."  
  default     = "ParallelCluster"  
}  
  
variable "api_version" {  
  type        = string  
  description = "The version of the ParallelCluster API."  
}  
  
variable "os" {  
  type        = string  
  description = "The OS of the ParallelCluster image."  
}  
  
variable "architecture" {  
  type        = string  
  description = "The architecture of the ParallelCluster image."  
}
```

6. `terraform.tfvars` 파일을 만들어 변수에 임의의 값을 설정합니다.

아래 파일을 사용하면 스택 이름과 함께 이미 배포된 기존 ParallelCluster API 3.10.0을 사용하여 x86\_64용 Amazon Linux 2 us-east-1 기반의 사용자 지정 AMI를 배포합니다. us-east-1 MyParallelClusterAPI-310

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

os = "alinux2"
architecture = "x86_64"
```

7. 파일을 생성하여 이 프로젝트에서 outputs.tf 반환되는 출력을 정의합니다.

```
output "parent_image" {
  value = data.aws-parallelcluster_list_official_images.parent_image.official_images[0]
}

output "custom_image" {
  value = aws-parallelcluster_image.demo01
}
```

프로젝트 디렉토리는 다음과 같습니다.

```
my-amis
### main.tf - Terraform entrypoint where the ParallelCluster module is configured.
### outputs.tf - Defines the cluster as a Terraform output.
### providers.tf - Configures the providers: ParallelCluster and AWS.
### terraform.tf - Import the ParallelCluster provider.
### terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
### variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

## AMI 배포하기

AMI를 배포하려면 표준 Terraform 명령을 순서대로 실행합니다.

1. 프로젝트 빌드:

```
terraform init
```

## 2. 배포 계획을 정의하세요.

```
terraform plan -out tfplan
```

## 3. 계획 배포:

```
terraform apply tfplan
```

## 필요한 권한

Terraform으로 사용자 지정 AMI를 배포하려면 다음 권한이 필요합니다.

- API와의 상호 작용을 담당하는 ParallelCluster API 역할을 맡으십시오. ParallelCluster
- ParallelCluster API AWS CloudFormation 스택을 설명하여 존재하는지 확인하고 해당 매개변수와 출력을 검색합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Resource": "arn:PARTITION:iam::ACCOUNT:role/PCAPIUserRole-*",
      "Effect": "Allow",
      "Sid": "AssumePCAPIUserRole"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks"
      ],
      "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    }
  ]
}
```

## AWS ParallelClusterID 센터와의 UI 통합

이 자습서의 목표는 클러스터와 공유할 수 있는 Active Directory의 사용자를 통합하는 싱글 사인온 솔루션을 위해 AWS ParallelCluster UI를 IAM Identity Center와 통합하는 방법을 시연하는 것입니다.

### AWS ParallelCluster

AWS ParallelCluster를 사용하는 경우 AWS ParallelCluster 이미지 및 클러스터를 생성하거나 업데이트할 때 생성된 AWS 리소스에 대한 비용만 지불하면 됩니다. 자세한 내용은 [AWS에서 사용하는 서비스 AWS ParallelCluster](#) 항목을 참조하세요.

### 사전 조건:

- [여기의 지침에 따라 설치할 수 있는 기존 AWS ParallelCluster UI입니다.](#)
- 기존의 관리형 Active Directory (가급적이면 [통합용으로도](#) 사용할 수 있는 디렉토리) AWS ParallelCluster

## IAM Identity Center 활성화

ID 센터가 이미 사용자 AWS Managed Microsoft AD (Active Directory)에 연결되어 있으면 사용할 수 있으며 IAM ID 센터에 애플리케이션 추가 섹션으로 건너뛰어도 됩니다.

ID 센터가 아직 연결되지 않은 경우 아래 단계에 따라 설정하십시오. AWS Managed Microsoft AD

### ID 센터 활성화

1. 콘솔에서 IAM ID 센터로 이동합니다. (현재 계정이 있는 지역에 있는지 확인하십시오AWS Managed Microsoft AD.)
2. 활성화 버튼을 클릭하면 조직을 활성화할지 여부를 묻는 메시지가 표시될 수 있습니다. 이는 필수 사항이므로 활성화하도록 선택할 수 있습니다. 참고: 이렇게 하면 계정 관리자에게 확인 이메일이 발송됩니다. 링크를 따라 확인하라는 확인 이메일이 발송됩니다.

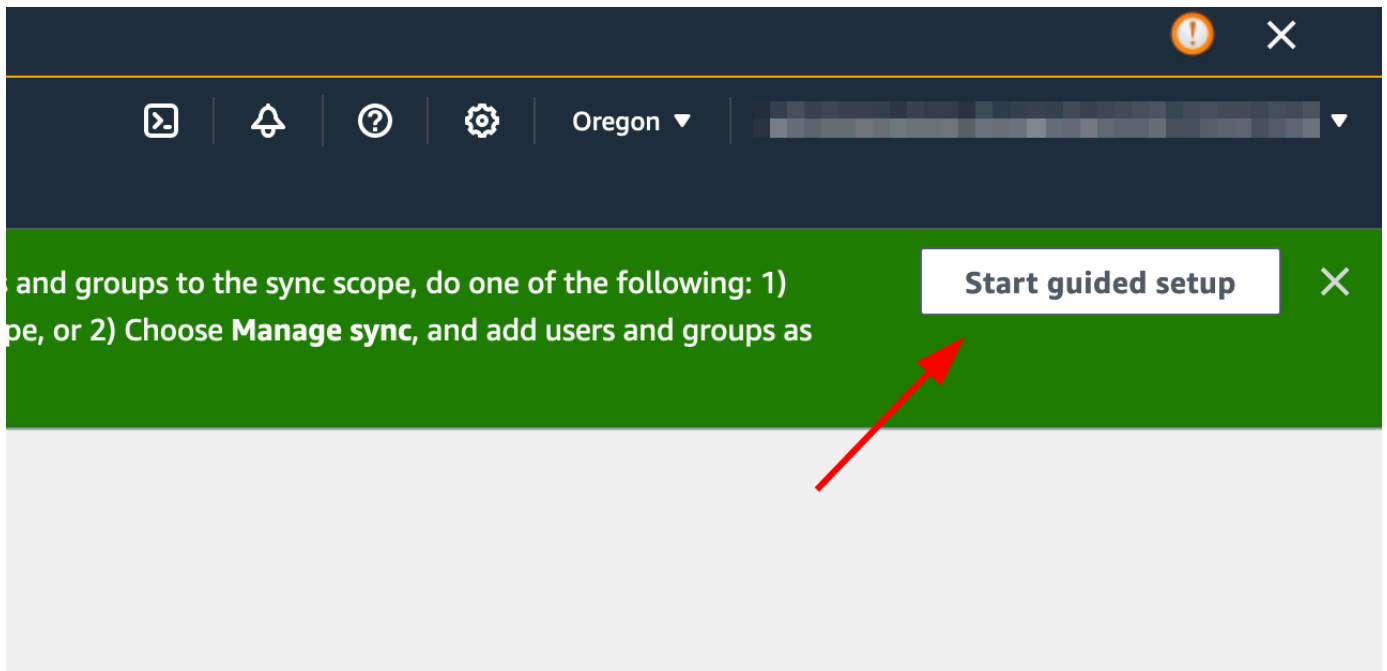
### ID 센터를 관리형 AD에 연결

1. ID 센터를 활성화한 후 다음 페이지에서 권장 설정 단계가 표시되고 1단계에서 ID 소스 선택을 선택합니다.
2. ID 소스 섹션에서 작업 드롭다운 메뉴 (오른쪽 상단)를 클릭한 다음 ID 소스 변경을 선택합니다.
3. 액티브 디렉터리를 선택합니다.

4. 기존 디렉터리에서 디렉터를 선택합니다.
5. 다음을 클릭합니다.
6. 변경 내용을 검토하고 하단으로 스크롤하여 입력란에 ACCEPT 를 입력하여 확인한 다음 ID 소스 변경을 클릭합니다.
7. 변경이 완료될 때까지 기다리면 상단에 녹색 배너가 보일 것입니다.

#### 사용자 및 그룹을 Identity Center에 동기화하는 중

1. 녹색 배너에서 가이드 설정 시작 (오른쪽 상단의 버튼) 을 클릭합니다.



2. 속성 매핑 구성에서 다음을 클릭합니다.
3. 동기화 범위 구성 섹션에서 ID 센터에 동기화할 사용자 이름을 입력한 다음 추가를 클릭합니다.
4. 사용자 및 그룹 추가를 완료한 후 다음을 클릭합니다.

**Users**

Groups

**User**

corp.pcluster.com
▼

Q

Add

**Added users and groups (4)** Remove

|                          | Username / Group name | Type | Domain            |
|--------------------------|-----------------------|------|-------------------|
| <input type="checkbox"/> | user1                 | User | corp.pcluster.com |
| <input type="checkbox"/> | user2                 | User | corp.pcluster.com |
| <input type="checkbox"/> | admin1                | User | corp.pcluster.com |
| <input type="checkbox"/> | admin2                | User | corp.pcluster.com |

Cancel

Previous

Next

5. 변경 사항을 검토한 다음 구성 저장을 클릭합니다.
6. 다음 화면에서 동기화되지 않는 사용자에게 경고가 표시되면 오른쪽 상단의 동기화 재개 버튼을 선택하세요.
7. 그런 다음 사용자를 활성화하려면 왼쪽의 사용자 탭에서 사용자를 선택한 다음 사용자 액세스 활성화 > 사용자 액세스 활성화를 클릭합니다.

참고: 상단에 경고 배너가 있는 경우 동기화 재개를 선택하고 사용자가 동기화할 때까지 기다려야 할 수 있습니다 (새로 고침 버튼을 눌러 아직 동기화되었는지 확인하세요).



The screenshot shows the IAM Identity Center 'Users' page. On the left is a navigation menu with options like 'Dashboard', 'Users', 'Groups', 'Settings', 'Multi-account permissions', and 'Application assignments'. The main content area displays a table of users:

| Username                 | Display name             | Status  | MFA device |
|--------------------------|--------------------------|---------|------------|
| user1@corp.pcluster.com  | user1@corp.pcluster.com  | Enabled | None       |
| admin1@corp.pcluster.com | admin1@corp.pcluster.com | Enabled | None       |
| user2@corp.pcluster.com  | user2@corp.pcluster.com  | Enabled | None       |
| admin2@corp.pcluster.com | admin2@corp.pcluster.com | Enabled | None       |

## IAM ID 센터에 애플리케이션 추가

사용자를 IAM ID 센터와 동기화한 후에는 새 애플리케이션을 추가해야 합니다. 이렇게 하면 IAM Identity Center 포털에서 사용할 수 있는 SSO 지원 애플리케이션이 구성됩니다. 이 경우 AWS ParallelCluster UI를 애플리케이션으로 추가하고 IAM Identity Center는 ID 제공자로 추가할 예정입니다.

다음 단계에서는 IAM ID AWS ParallelCluster 센터에 UI를 애플리케이션으로 추가합니다. AWS ParallelCluster UI는 사용자가 클러스터를 관리하는 데 도움이 되는 웹 포털입니다. 자세한 내용은 [AWS ParallelClusterUI](#)를 참조하십시오.

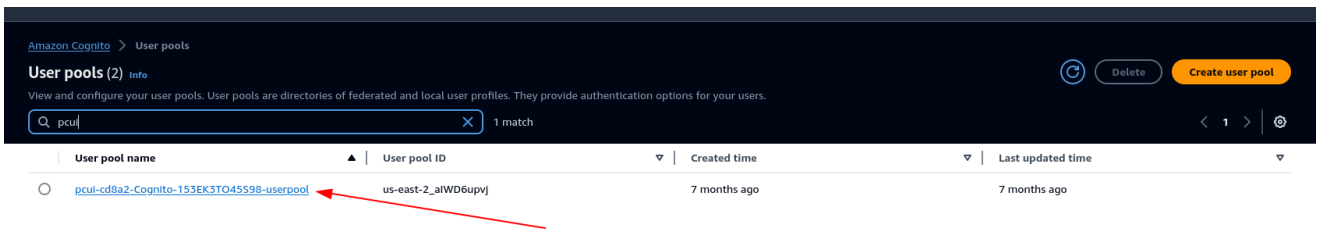
### ID 센터에서 애플리케이션 설정

1. IAM ID 센터 > 애플리케이션 (왼쪽 메뉴 표시줄에 있는 애플리케이션 클릭) 에서
2. 애플리케이션 추가를 클릭합니다.
3. 사용자 지정 SAML 2.0 애플리케이션 추가를 선택합니다.
4. 다음을 클릭합니다.
5. 사용하려는 디스플레이 이름과 설명을 선택합니다 (예: PCUI 및 AWS ParallelCluster UI).
6. IAM ID 센터 메타데이터에서 IAM ID 센터 SAML 메타데이터 파일의 링크를 복사하고 나중에 사용할 수 있도록 저장합니다. 이 링크는 웹 앱에서 SSO를 구성할 때 사용됩니다.

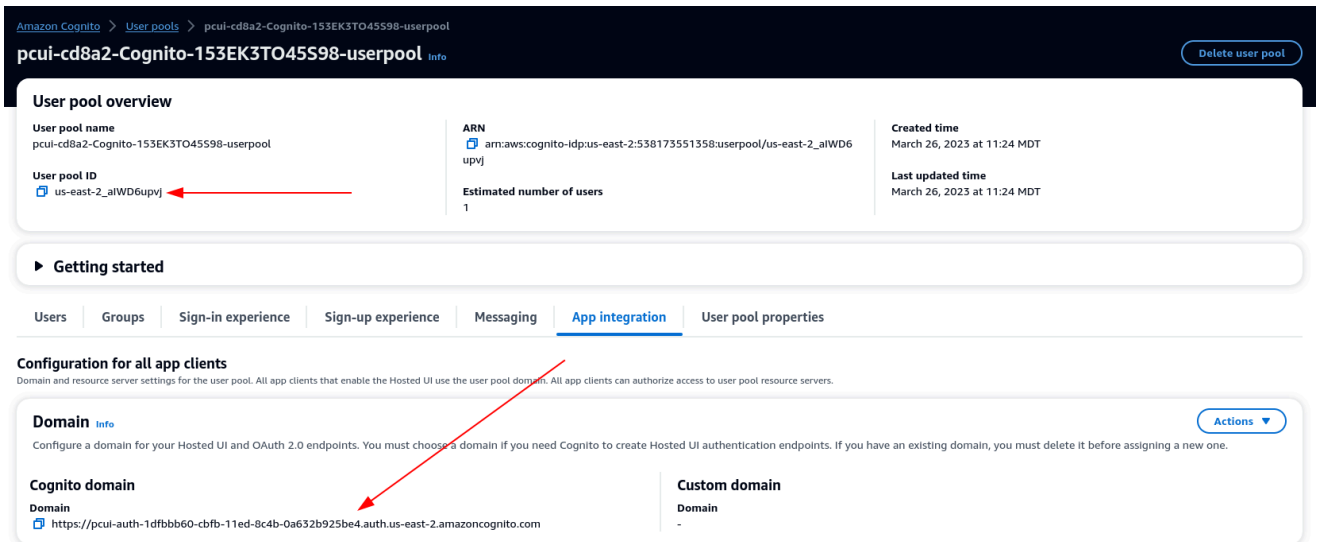
7. 애플리케이션 속성의 애플리케이션 시작 URL에 PCUI 주소를 입력합니다. CloudFormation 콘솔로 이동하여 PCUI에 해당하는 스택 (예: parallelcluster-ui) 을 선택한 다음 출력 탭으로 이동하여 UIURL을 찾으면 확인할 수 있습니다. ParallelCluster

예: <https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com>

8. 애플리케이션 메타데이터에서 메타데이터 값 수동 입력을 선택합니다. 그런 다음 다음 값을 입력합니다.
  - a. 중요: 도메인 접두사, 지역 및 userpool-id 값을 사용자 환경에 맞는 정보로 바꾸십시오.
  - b. Amazon Cognito > 사용자 풀 콘솔을 열어 도메인 접두사, 지역 및 userpool-id를 확인할 수 있습니다.



- c. PCUI에 해당하는 사용자 풀을 선택합니다 (PCUI-CD8A2-Cognito-153EK3to45S98-사용자 풀과 같은 사용자 풀 이름을 가짐).
- d. 앱 통합으로 이동합니다.



9. <domain-prefix>애플리케이션 어설션 소비자 서비스 (ACS) URL: <https://.<region>.amazoncognito.com/saml2/idrepsonse>

애플리케이션 SAML 잠재고객: `urn:amazon:cognito:sp: <userpool-id>`

10. 제출을 선택합니다. 그런 다음 추가한 애플리케이션의 세부 정보 페이지로 이동합니다.

11. 작업 드롭다운 목록을 선택하고 속성 매핑 편집을 선택합니다. 그런 다음 다음 속성을 제공합니다.
  - a. 애플리케이션의 사용자 속성: 제목 (참고: 제목은 미리 입력됨) → IAM ID 센터의 다음 문자열 값 또는 사용자 속성에 매핑: `${user:email}`, 형식: 이메일 주소
  - b. 애플리케이션의 사용자 속성: 이메일 → IAM ID 센터의 이 문자열 값 또는 사용자 속성에 매핑: `${user:email}`, 형식: 미지정

Attribute mappings for PCUI

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. [Learn more](#)

| User attribute in the application                        | Maps to this string value or user attribute in IAM Identity Center | Format       |                                       |
|----------------------------------------------------------|--------------------------------------------------------------------|--------------|---------------------------------------|
| Subject                                                  | <code>\${user:email}</code>                                        | emailAddress |                                       |
| email                                                    | <code>\${user:email}</code>                                        | unspecified  | <input type="button" value="Remove"/> |
| <input type="button" value="Add new attribute mapping"/> |                                                                    |              |                                       |


12. 변경 내용을 저장합니다.

13. 사용자 할당 버튼을 선택한 다음 사용자를 애플리케이션에 할당합니다. 이들은 Active Directory에서 PCUI 인터페이스에 액세스할 수 있는 사용자입니다.

IAM Identity Center > Applications > PCUI

### PCUI

Details Actions ▼

|                                                                                     |                      |                                       |
|-------------------------------------------------------------------------------------|----------------------|---------------------------------------|
|  | Display name<br>PCUI | Description<br>AWS ParallelCluster UI |
|-------------------------------------------------------------------------------------|----------------------|---------------------------------------|

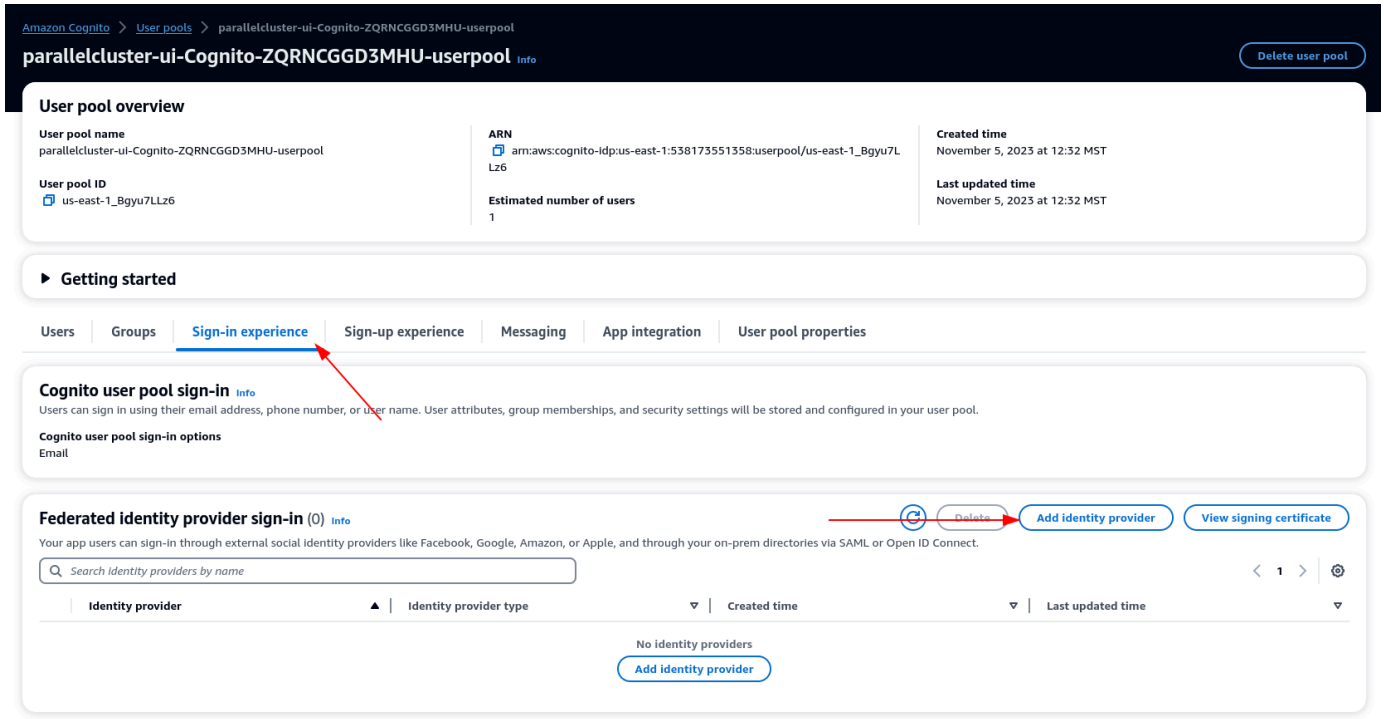
Assigned users (1)

The following users and groups from your connected directory can access this application. [Learn more](#)

| <input type="checkbox"/> | User/Group name                          | Type |
|--------------------------|------------------------------------------|------|
| <input type="checkbox"/> | <a href="#">admin1@corp.pcluster.com</a> | User |

## 사용자 풀에서 IAM ID 센터를 SAML IdP로 구성

1. 사용자 풀 설정에서 로그인 환경 > ID 공급자 추가를 선택합니다.



2. SAML IdP를 선택하세요
3. 제공자 이름을 입력하려면 다음을 입력하십시오. IdentityCenter
4. 메타데이터 문서 소스에서 메타데이터 문서 엔드포인트 URL 입력을 선택하고 Identity Center의 애플리케이션 설정 중에 복사한 URL을 입력합니다.
5. 속성에서 이메일의 경우 이메일을 선택합니다.

**SAML**  
Configure a SAML 2.0 identity provider for your user pool.

**Register your app with your SAML provider**  
To connect a SAML provider to Cognito, add your user pool as a relying party or application with your SAML 2.0 identity provider, and upload a metadata document to Cognito.

**Set up SAML federation with this user pool**

**Provider name** [Info](#)  
Enter a friendly name for your SAML 2.0 identity provider.  
IdentityCenter

**Identifiers - optional** [Info](#)  
Enter identifiers for this provider. Identifiers can be used to redirect users to the correct IdP in multitenant apps.  
Enter identifiers

Separate each identifier by a comma

**Sign-out flow** [Info](#)  
 Add sign-out flow  
Enable simultaneous sign-out from the SAML provider and Cognito.

**Metadata document source** [Info](#)  
Provide a SAML metadata document. This document is issued by your SAML provider. It includes the issuer's name, expiration information, and keys that can be used to validate the response from the identity provider.  
 Upload metadata document  
 Enter metadata document endpoint URL

**Enter metadata document endpoint URL** [Info](#)  
https://portal.sso.us-east-1.amazonaws.com/saml/metadata/NTM4MTczNTUxMzU4X2lucy0zNmZkNmQ3Y2NmMjU5ODM3

**Map attributes between your SAML provider and your user pool** [Info](#)  
Your required attributes are mapped to the equivalent SAML attributes. Each attribute you add must be mapped to a SAML attribute.

| User pool attribute | SAML attribute |
|---------------------|----------------|
| email               | email          |

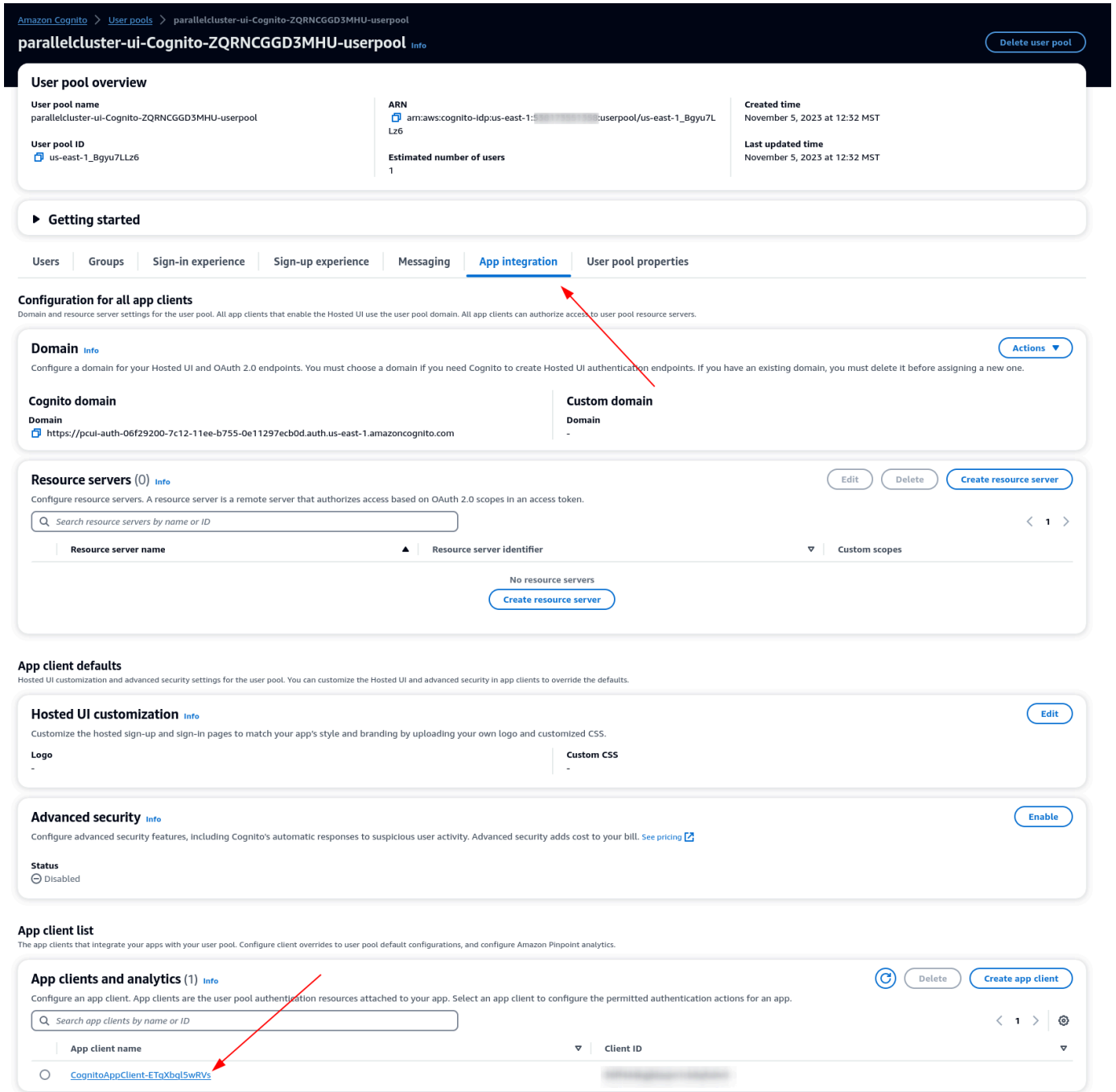
[Add another attribute](#)

[Cancel](#) [Add identity provider](#)

6. ID 제공업체 추가를 선택합니다.

IdP를 사용자 풀 앱 클라이언트와 통합

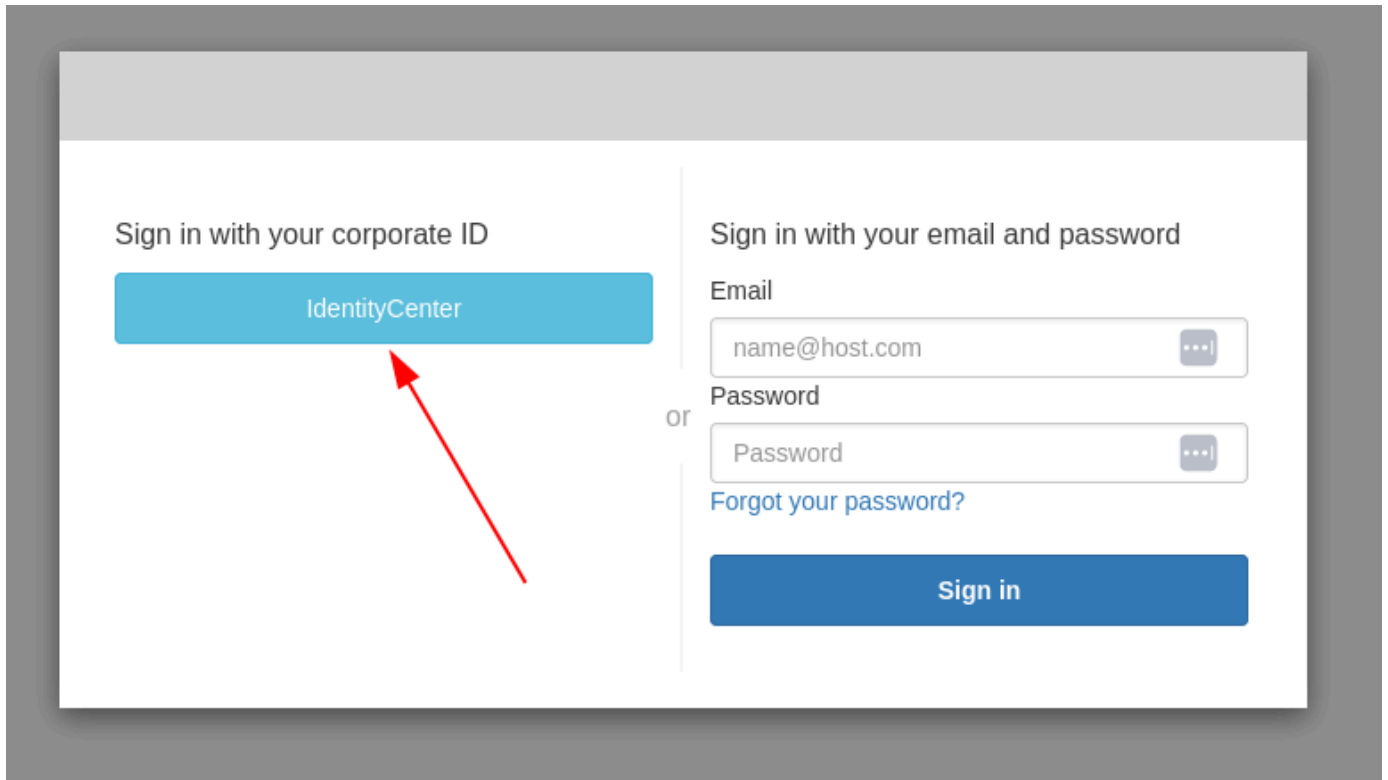
1. 그런 다음 사용자 풀의 앱 통합 섹션에서 앱 클라이언트 목록에 나열된 클라이언트를 선택합니다.



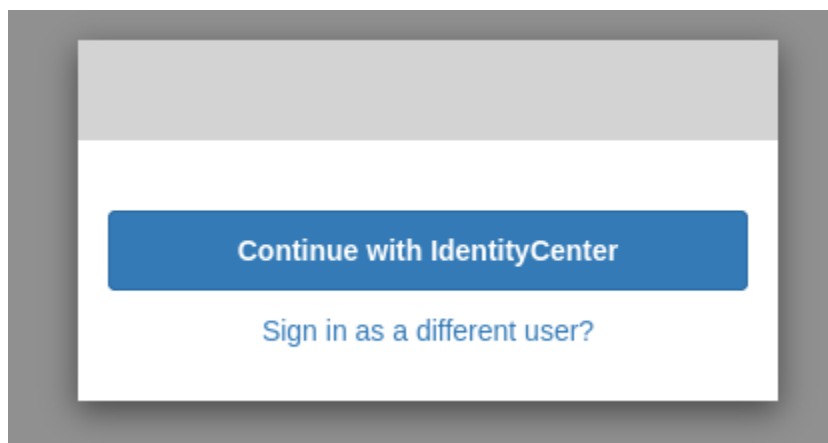
2. 호스팅된 UI에서 편집을 선택합니다.
3. ID IdentityCenter공급자에서도 선택합니다.
4. 변경 사항 저장(Save changes)을 선택합니다

설정을 검증하세요.

1. 다음으로 PCUI에 로그인하여 방금 만든 설정을 검증해 보겠습니다. PCUI 포털에 로그인하면 이제 회사 ID로 로그인할 수 있는 옵션이 표시됩니다.

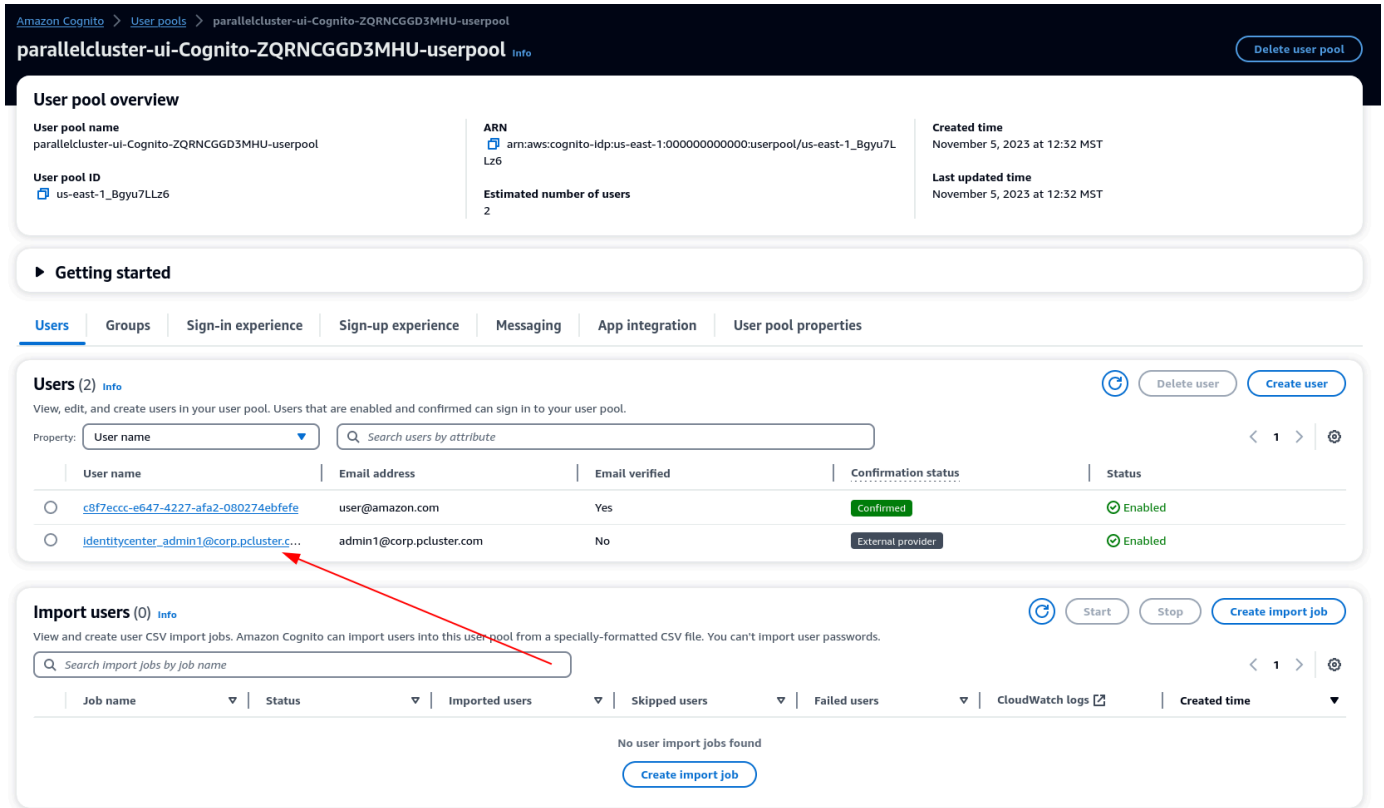


2. IdentityCenter버튼을 클릭하면 IAM Identity Center IdP 로그인으로 이동한 다음 PCUI가 포함된 애플리케이션이 있는 페이지가 표시되고 해당 애플리케이션이 열립니다.
3. 다음 화면이 표시되면 사용자가 Cognito 사용자 풀에 추가된 것입니다.



## 사용자를 관리자로 설정

1. 이제 Amazon Cognito > 사용자 풀 콘솔로 이동하여 새로 생성한 사용자 (접두사가 ID 센터여야 함) 를 선택합니다.



2. 그룹 멤버십에서 그룹에 사용자 추가를 선택하고 관리자를 선택한 다음 추가를 클릭합니다.
3. 이제 계속을 IdentityCenter 클릭하면 UI 페이지로 이동합니다. AWS ParallelCluster



# AWS ParallelCluster 문제 해결

AWS ParallelCluster 커뮤니티는 Wiki에 대한 다양한 문제 해결 팁을 제공하는 [AWS ParallelCluster GitHub Wiki](#) 페이지를 유지 관리합니다. 알려진 문제 목록을 알아보려면 [알려진 문제](#)를 참조하세요.

## 주제

- [클러스터를 생성하려는 경우](#)
- [작업을 실행하려는 경우](#)
- [클러스터를 업데이트하려는 경우](#)
- [스토리지에 액세스하려는 경우](#)
- [클러스터를 삭제하려는 경우](#)
- [AWS ParallelCluster API 스택 업그레이드 시도 중](#)
- [컴퓨팅 노드 초기화 오류가 표시되는 경우](#)
- [클러스터 상태 지표 문제 해결](#)
- [클러스터 배포 문제 해결](#)
- [Terraform을 사용한 클러스터 배포 문제 해결](#)
- [규모 조정 문제 해결](#)
- [배치 그룹 및 인스턴스 시작 문제](#)
- [교체할 수 없는 디렉터리](#)
- [NICE DCV 문제 해결](#)
- [AWS Batch 통합을 통해 클러스터의 문제 해결](#)
- [Active Directory와의 다중 사용자 통합 문제 해결](#)
- [사용자 지정 AMI 문제 해결](#)
- [cfn-hup이 실행 중이 아닐 때의 클러스터 업데이트 제한 시간 문제 해결](#)
- [네트워크 문제 해결](#)
- [onNodeUpdated 사용자 지정 작업에서 클러스터 업데이트가 실패한 경우](#)
- [사용자 지정 Slurm 구성에서 오류가 표시되는 경우](#)
- [클러스터 알람](#)
- [추가 지원](#)

## 클러스터를 생성하려는 경우

AWS ParallelCluster 버전 3.5.0 이상을 사용하여 클러스터를 생성할 때 로 `--rollback-on-failure` 설정된 상태에서 클러스터 생성에 실패한 경우 [pcluster describe-cluster](#) CLI 명령을 사용하여 상태 및 실패 정보를 가져옵니다. 이 경우 `pcluster describe-cluster` 출력의 예상 `clusterStatus`은 `CREATE_FAILED`입니다. `failureCode` 및 `failureReason`을 찾으려면 출력의 `failures` 섹션을 확인하세요. 그 후 다음 섹션에서 일치하는 `failureCode`를 찾아 추가 문제 해결 도움말을 찾아보세요. 자세한 내용은 [pcluster describe-cluster](#) 항목을 참조하세요.

다음 섹션에서는 헤드 노드의 로그(예: `/var/log/cfn-init.log` 및 `/var/log/chef-client.log` 파일)를 확인하는 것이 좋습니다. AWS ParallelCluster 로그 및 로그 확인 방법에 대한 자세한 내용은 [로그를 위한 키 로그 로그 검색 및 보존](#)을 참조하십시오.

없는 경우 AWS CloudFormation 콘솔로 이동하여 클러스터 스택을 확인하십시오. `failureCode` `HeadNodeWaitCondition` 또는 다른 리소스의 실패에 대해 알아보려면 `Status Reason`에서 추가 실패 상세 정보를 확인하세요. 자세한 내용은 [AWS CloudFormation 이벤트 보기 CREATE\\_FAILED](#) 항목을 참조하세요. 헤드 노드의 `/var/log/cfn-init.log` 및 `/var/log/chef-client.log` 파일을 확인합니다.

### **failureCode가 OnNodeConfiguredExecutionFailure일 시**

- 왜 실패했나요?

구성 내 헤드 노드 섹션의 `OnNodeConfigured`에 클러스터를 생성하기 위한 사용자 지정 스크립트를 제공했습니다. 하지만 사용자 지정 스크립트가 실행되지 않았습니다.

- 해결 방법은?

`/var/log/cfn-init.log` 파일을 확인하여 실패에 대해 자세히 알아보고 사용자 지정 스크립트에서 문제를 해결하는 방법을 알아보세요. 이 로그의 끝부분에서 `Running command runpostinstall` 메시지 뒤에 `OnNodeConfigured` 스크립트와 관련된 실행 정보가 표시될 수 있습니다.

### **failureCode가 OnNodeConfiguredDownloadFailure일 시**

- 왜 실패했나요?

구성 내 헤드 노드 섹션의 `OnNodeConfigured`에 클러스터를 생성하기 위한 사용자 지정 스크립트를 제공했습니다. 하지만 사용자 지정 스크립트가 다운로드되지 않았습니다.

- 해결 방법은?

URL이 유효하고 액세스가 올바르게 구성되어 있는지 확인하세요. 사용자 지정 부트스트랩 스크립트의 구성에 대한 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

`/var/log/cfn-init.log` 파일을 확인하세요. 이 로그의 끝부분에서 Running command `runpostinstall` 메시지 다음에 다운로드를 포함한 `OnNodeConfigured` 스크립트 처리와 관련된 실행 정보가 표시될 수 있습니다.

## failureCode가 OnNodeConfiguredFailure일 시

- 왜 실패했나요?

구성 내 헤드 노드 섹션의 `OnNodeConfigured`에 클러스터를 생성하기 위한 사용자 지정 스크립트를 제공했습니다. 하지만 클러스터 배포에서 사용자 지정 스크립트 사용이 실패했습니다. 즉각적인 원인을 확인할 수 없으며 추가 조사가 필요합니다.

- 해결 방법은?

`/var/log/cfn-init.log` 파일을 확인하세요. 이 로그의 끝부분에서 Running command `runpostinstall` 메시지 뒤에 `OnNodeConfigured` 스크립트 처리와 관련된 실행 정보가 표시될 수 있습니다.

## failureCode가 OnNodeStartExecutionFailure일 시

- 왜 실패했나요?

구성 내 헤드 노드 섹션의 `OnNodeStart`에 클러스터를 생성하기 위한 사용자 지정 스크립트를 제공했습니다. 하지만 사용자 지정 스크립트가 실행되지 않았습니다.

- 해결 방법은?

`/var/log/cfn-init.log` 파일을 확인하여 실패에 대해 자세히 알아보고 사용자 지정 스크립트에서 문제를 해결하는 방법을 알아보세요. 이 로그의 끝부분에서 Running command `runpreinstall` 메시지 뒤에 `OnNodeStart` 스크립트와 관련된 실행 정보가 표시될 수 있습니다.

## failureCode가 OnNodeStartDownloadFailure일 시

- 왜 실패했나요?

구성 내 헤드 노드 섹션의 OnNodeStart에 클러스터를 생성하기 위한 사용자 지정 스크립트를 제공했습니다. 하지만 사용자 지정 스크립트가 다운로드되지 않았습니다.

- 해결 방법은?

URL이 유효하고 액세스가 올바르게 구성되어 있는지 확인하세요. 사용자 지정 부트스트랩 스크립트의 구성에 대한 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

/var/log/cfn-init.log 파일을 확인하세요. 이 로그의 끝부분에서 Running command runpreinstall 메시지 다음에 다운로드를 포함한 OnNodeStart 스크립트 처리와 관련된 실행 정보가 표시될 수 있습니다.

## failureCode가 OnNodeStartFailure일 시

- 왜 실패했나요?

구성 내 헤드 노드 섹션의 OnNodeStart에 클러스터를 생성하기 위한 사용자 지정 스크립트를 제공했습니다. 하지만 클러스터 배포에서 사용자 지정 스크립트 사용이 실패했습니다. 즉각적인 원인을 확인할 수 없으며 추가 조사가 필요합니다.

- 해결 방법은?

/var/log/cfn-init.log 파일을 확인하세요. 이 로그의 끝부분에서 Running command runpreinstall 메시지 뒤에 OnNodeStart 스크립트 처리와 관련된 실행 정보가 표시될 수 있습니다.

## failureCode가 EbsMountFailure일 시

- 왜 실패했나요?

클러스터 구성에 정의된 EBS 볼륨이 탑재되지 못했습니다.

- 해결 방법은?

/var/log/chef-client.log 파일에서 실패 세부 정보를 확인하세요.

## failureCode가 EfsMountFailure일 시

- 왜 실패했나요?

클러스터 구성에 정의된 Amazon EFS 볼륨이 탑재되지 못했습니다.

- 해결 방법은?

기존 Amazon EFS 파일 시스템을 정의한 경우 클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인하세요. 자세한 내용은 [SharedStorage/EfsSettings/FileSystemId](#)를 참조하세요.

`/var/log/chef-client.log` 파일에서 실패 세부 정보를 확인하세요.

## failureCode가 FsxMountFailure일 시

- 왜 실패했나요?

클러스터 구성에 정의된 Amazon FSx 파일 시스템이 탑재되지 못했습니다.

- 해결 방법은?

기존 Amazon FSx 파일 시스템을 정의한 경우 클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인하세요. 자세한 내용은 [SharedStorage/FsxLustreSettings/FileSystemId](#)를 참조하세요.

`/var/log/chef-client.log` 파일에서 실패 세부 정보를 확인하세요.

## failureCode가 RaidMountFailure일 시

- 왜 실패했나요?

클러스터 구성에 정의된 RAID 볼륨이 탑재되지 못했습니다.

- 해결 방법은?

`/var/log/chef-client.log` 파일에서 실패 세부 정보를 확인하세요.

## failureCode가 AmiVersionMismatch일 시

- 왜 실패했나요?

사용자 지정 AMI를 생성하는 데 사용된 AWS ParallelCluster 버전은 클러스터를 구성하는 데 사용된 AWS ParallelCluster 버전과 다릅니다. CloudFormation 콘솔에서 클러스터 CloudFormation 스택 세부 정보를 확인하고 에서 AWS ParallelCluster 버전 및 AMI에 대한 추가 세부 정보를 확인하십시오.

오. Status Reason HeadNodeWaitCondition 자세한 정보는 [에서 AWS CloudFormation 이벤트 보기 CREATE\\_FAILED](#)을 참조하세요.

- 해결 방법은?

사용자 지정 AMI를 생성하는 데 사용된 AWS ParallelCluster 버전이 클러스터를 구성하는 데 사용된 AWS ParallelCluster 버전과 동일한지 확인하십시오. 사용자 지정 AMI 버전 또는 pcluster CLI 버전을 변경하여 동일하게 만들 수 있습니다.

## failureCode가 InvalidAmi일 시

- 왜 실패했나요?

사용자 지정 AMI는 를 사용하여 빌드되지 않았으므로 유효하지 않습니다 AWS ParallelCluster.

- 해결 방법은?

pcluster build-image 명령을 사용하여 AMI를 상위 이미지로 만들어 AMI를 생성합니다. 자세한 내용은 [pcluster build-image](#) 항목을 참조하세요.

## failureCode가 HeadNodeBootstrapFailure이며 failureReason이 헤드 노드 설정에 실패했습니다.

- 왜 실패했나요?

즉각적인 원인을 확인할 수 없으며 추가 조사가 필요합니다. 예를 들어 클러스터가 보호 상태일 수 있는데, 이는 정적 컴퓨팅 플릿을 프로비저닝하지 못했기 때문일 수 있습니다.

- 해결 방법은?

/var/log/chef-client.log. 파일에서 실패 세부 정보를 확인하세요.

### Note

RuntimeError 예외 Cluster state has been set to PROTECTED mode due to failures detected in static node provisioning가 표시되면 클러스터가 보호 상태인 것입니다. 자세한 내용은 [보호 모드를 디버깅하는 방법](#) 항목을 참조하세요.

## **failureCode**가 **HeadNodeBootstrapFailure**이며 **failureReason**이 클러스터 생성 시간이 초과되었습니다.

- 왜 실패했나요?

기본적으로 클러스터 생성을 완료하는 데 걸리는 시간 제한은 30분입니다. 클러스터 생성이 이 기간 내에 완료되지 않으면 시간 초과 오류와 함께 클러스터 생성이 실패합니다. 여러 가지 이유로 클러스터 생성 시간이 초과될 수 있습니다. 예를 들어 헤드 노드 생성 실패, 네트워크 문제, 헤드 노드에서 실행하는 데 너무 오래 걸리는 사용자 지정 스크립트, 컴퓨팅 노드에서 실행되는 사용자 지정 스크립트의 오류 또는 컴퓨팅 노드 프로비저닝의 긴 대기 시간으로 인해 시간 초과 실패가 발생할 수 있습니다. 즉각적인 원인을 확인할 수 없으며 추가 조사가 필요합니다.

- 해결 방법은?

`/var/log/cfn-init.log` 및 `/var/log/chef-client.log` 파일에서 실패 세부 정보를 확인하세요. AWS ParallelCluster 로그 및 로그 가져오기 방법에 대한 자세한 내용은 [디버깅을 위한 키 로그](#) 및 [로그 검색 및 보존](#)을 참조하세요.

이러한 로그에서 다음을 발견할 수 있습니다.

- **chef-client.log** 끝부분에 **Waiting for static fleet capacity provisioning**가 표시되는 경우

이는 정적 노드의 전원이 켜질 때까지 기다릴 때 클러스터 생성 시간이 초과되었음을 나타냅니다. 자세한 내용은 [컴퓨팅 노드 초기화 오류가 표시되는 경우](#) 항목을 참조하세요.

- **cfn-init.log** 끝부분에 **OnNodeConfigured** 또는 **OnNodeStart** 노드 스크립트가 종료되지 않았다고 표시되는 경우

이는 `OnNodeConfigured` 또는 `OnNodeStart` 사용자 지정 스크립트를 실행하는 데 시간이 오래 걸리고 시간 초과 오류가 발생했음을 나타냅니다. 사용자 지정 스크립트에서 오랜 시간 실행으로 이어질 수 있는 문제가 있는지 확인합니다. 사용자 지정 스크립트를 실행하는 데 시간이 오래 걸리는 경우 다음 예와 같이 클러스터 구성 파일에 `DevSettings` 섹션을 추가하여 제한 시간을 변경하는 것이 좋습니다.

```
DevSettings:
  Timeouts:
    HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- 로그를 찾을 수 없거나 헤드 노드가 성공적으로 생성되지 않았습니다.

헤드 노드가 성공적으로 생성되지 않아 로그를 찾을 수 없을 수 있습니다. CloudFormation 콘솔에서 클러스터 스택 세부 정보를 보고 추가 장애 세부 정보를 확인하십시오.

## **failureCode**가 **HeadNodeBootstrapFailure**이며 **failureReason**이 헤드 노드 부트스트랩에 실패했습니다.

- 왜 실패했나요?

즉각적인 원인을 확인할 수 없으며 추가 조사가 필요합니다.

- 해결 방법은?

`/var/log/cfn-init.log` 및 `/var/log/chef-client.log` 파일을 확인합니다.

## **failureCode**가 **ResourceCreationFailure**일 시

- 왜 실패했나요?

클러스터 생성 프로세스 중 일부 리소스 생성이 실패했습니다. 실패는 다양한 이유로 발생할 수 있습니다. 예를 들어 용량 문제나 잘못 구성된 IAM 정책으로 인해 리소스 생성 실패가 발생할 수 있습니다.

- 해결 방법은?

CloudFormation 콘솔에서 클러스터 스택을 보고 추가 리소스 생성 실패 세부 정보를 확인합니다.

## **failureCode**가 **ClusterCreationFailure**일 시

- 왜 실패했나요?

즉각적인 원인을 확인할 수 없으며 추가 조사가 필요합니다.

- 해결 방법은?

CloudFormation 콘솔에서 클러스터 스택을 보고 에서 추가 실패 세부 정보를 확인하십시오.

Status Reason HeadNodeWaitCondition

`/var/log/cfn-init.log` 및 `/var/log/chef-client.log` 파일을 확인합니다.



## **WaitCondition timed out...** CloudFormation 스택에서 보기

자세한 정보는 [failureCode가 HeadNodeBootstrapFailure](#)이며 [failureReason이 클러스터 생성 시간이 초과되었습니다.](#)을 참조하세요.

## **Resource creation cancelled** CloudFormation 스택에서 보기

자세한 정보는 [failureCode가 ResourceCreationFailure](#)일 시을 참조하세요.

## AWS CloudFormation 스택의 오류 확인 **Failed to run cfn-init...** 또는 기타 오류

추가 실패 세부 정보는 `/var/log/cfn-init.log` 및 `/var/log/chef-client.log`를 확인하세요.

## **INFO: Waiting for static fleet capacity provisioning**로 끝나는 `chef-client.log`이 표시되는 경우

이는 정적 노드의 전원이 켜질 때까지 기다릴 때 발생하는 클러스터 생성 타임아웃과 관련이 있습니다. 자세한 내용은 [컴퓨팅 노드 초기화 오류가 표시되는 경우](#) 항목을 참조하세요.

## **Failed to run preinstall or postinstall in cfn-init.log**가 표시되는 경우

클러스터 구성 HeadNode 섹션에 `OnNodeConfigured` 또는 `OnNodeStart` 스크립트가 있습니다. 스크립트가 제대로 작동하지 않습니다. `/var/log/cfn-init.log` 파일에서 사용자 지정 스크립트 오류 세부 정보를 확인하세요.

## **This AMI was created with xxx, but is trying to be used with xxx...** CloudFormation 스택에서 보기

자세한 정보는 [failureCode가 AmiVersionMismatch](#)일 시을 참조하세요.

## **This AMI was not baked by AWS ParallelCluster...** CloudFormation 스택에서 보기

자세한 정보는 [failureCode가 InvalidAmi](#)일 시을 참조하세요.

## pcluster create-cluster 명령이 로컬에서 실행되지 않는 경우

로컬 파일 시스템의 ~/.parallelcluster/pcluster-cli.log에서 오류 세부 정보를 확인하세요.

### 추가 지원

[클러스터 배포 문제 해결](#)의 문제 해결 지침을 따르세요.

현재 AWS ParallelCluster 사용 중인 시나리오가 [GitHub 알려진 문제에 포함되어](#) 있는지 확인하세요 GitHub.

추가 지원이 필요하면 [추가 지원](#)을 참조하세요.

### 작업을 실행하려는 경우

**srun 대화형 작업이 srun: error: fwd\_tree\_thread: can't find address for <host>, check slurm.conf** 오류가 발생하여 실패합니다.

- 왜 실패했나요?

srun 명령을 실행하여 작업을 제출한 다음 업데이트가 완료된 후 Slurm 대몬(daemon)을 다시 시작하지 않고 pcluster update-cluster 명령을 사용하여 대기열 크기를 늘렸습니다.

Slurm은 통신을 최적화하기 위해 Slurm 대몬(daemon)을 트리 계층으로 구성합니다. 이 계층 구조는 대몬(daemon)이 시작될 때만 업데이트됩니다.

srun를 사용하여 작업을 시작한 다음 pcluster update-cluster 명령을 실행하여 대기열 크기를 늘린다고 가정해 보겠습니다. 업데이트의 일부로 새 컴퓨팅 노드가 시작됩니다. 그런 다음 새 컴퓨팅 노드 중 하나에 작업을 Slurm 대기열에 추가합니다. 이 경우 Slurm 대몬(daemon)과 srun는 모두 새 컴퓨팅 노드를 감지하지 못합니다. srun은 새 노드를 감지하지 못하므로 오류가 반환됩니다.

- 해결 방법은?

모든 컴퓨팅 노드에서 Slurm 대몬(daemon)을 재시작한 다음 srun를 사용하여 작업을 제출하세요. 컴퓨팅 노드를 재시작하는 scontrol reboot 명령을 실행하여 Slurm 대몬(daemon) 재시작을 예약할 수 있습니다. 자세한 내용은 Slurm 설명서의 [scontrol reboot](#)를 참조하세요. 대응하는 systemd 서비스의 재시작을 요청하여 컴퓨팅 노드에서 Slurm 대몬(daemon)을 수동으로 재시작할 수도 있습니다.

## 작업이 `squeue` 명령을 실행한 CF 상태에서 멈췄습니다.

동적 노드의 전원을 켤 때 발생하는 문제일 수 있습니다. 자세한 내용은 [컴퓨팅 노드 초기화 오류가 표시되는 경우](#) 항목을 참조하세요.

## 대규모 작업을 실행한 후 `nfsd: too many open connections, consider increasing the number of threads in /var/log/messages`가 표시된 경우

네트워크로 연결된 파일 시스템에서 네트워크 제한에 도달하면 I/O 대기 시간도 늘어납니다. 네트워크가 네트워크 및 I/O 지표 모두에 대한 데이터를 쓰는 데 사용되므로 소프트웨어 락업이 발생할 수 있습니다.

5세대 인스턴스에서는 패킷 카운터를 노출하기 위해 ENA 드라이버가 사용됩니다. 이러한 카운터는 네트워크가 인스턴스 대역폭 한도에 도달한 시점을 기준으로 AWS 패킷 수를 계산합니다. 이러한 카운터가 0보다 큰지 확인할 수 있습니다. 0보다 크다면 대역폭 한도를 초과한 것입니다. `ethtool -S eth0 | grep exceeded`를 실행하여 이러한 카운터를 볼 수 있습니다.

네트워크 한도를 초과하는 것은 NFS 연결을 너무 많이 지원한 결과인 경우가 많습니다. 이는 네트워크 한도에 도달하거나 초과할 때 가장 먼저 확인해야 할 사항 중 하나입니다.

예를 들어 다음 출력은 누락된 패키지를 표시합니다.

```
$ ethtool -S eth0 | grep exceeded
bw_in_allowance_exceeded: 38750610
bw_out_allowance_exceeded: 1165693
pps_allowance_exceeded: 103
contrack_allowance_exceeded: 0
linklocal_allowance_exceeded: 0
```

이 메시지가 표시되지 않도록 헤드 노드 인스턴스 유형을 성능이 더 좋은 인스턴스 유형으로 변경하는 것이 좋습니다. 데이터 스토리지를 NFS 공유로 내보내지 않는 공유 스토리지 파일 시스템(예: Amazon EFS 또는 Amazon FSx)으로 옮기는 것을 고려해 보세요. 자세한 내용은 AWS ParallelCluster Wiki [공유 스토리지](#) on의 [모범 사례](#)를 참조하십시오. GitHub

## MPI 작업 실행

### 디버그 모드 활성화

OpenMPI 디버그 모드를 활성화하려면 [디버깅에 도움이 되는 Open MPI의 제어 기능](#)을 참조하세요.

IntelMPI 디버그 모드를 활성화하려면 [기타 환경 변수](#)를 참조하세요.

## 작업 출력에서 **MPI\_ERRORS\_ARE\_FATAL** 및 **OPAL ERROR**이 표시되는 경우

이러한 오류 코드는 애플리케이션의 MPI 계층에서 가져온 것입니다. 애플리케이션에서 MPI 디버그 로그를 가져오는 방법은 [디버그 모드 활성화](#) 항목을 참조하세요.

이 오류가 발생하는 원인은 응용 프로그램이 OpenMPI와 같은 특정 MPI 구현용으로 컴파일되었고 IntelMPI와 같은 다른 MPI 구현을 사용하여 응용 프로그램을 실행하려고 하기 때문일 수 있습니다. 동일한 MPI 구현으로 애플리케이션을 컴파일하고 실행하고 있는지 확인하세요.

## 관리형 DNS를 비활성화한 상태에서 **mpirun** 사용

/Dns [SlurmSettings/DisableManagedDns](#) 및 [UseEc2Hostnames](#)를 로 설정하여 true 만든 클러스터의 경우 [DNS에서](#) Slurm 노드 이름을 확인하지 않습니다. SlurmMPI 프로세스가 nodenames 활성화되지 않은 경우 및 컨텍스트에서 MPI 작업이 실행되는 경우 MPI 프로세스를 부트스트랩할 수 있습니다. Slurm [Slurm MPI 사용 설명서](#)의 지침에 따라 Slurm으로 MPI 작업을 실행하는 것이 좋습니다.

## 클러스터를 업데이트하려는 경우

**pcluster update-cluster** 명령이 로컬에서 실행되지 않습니다.

로컬 파일 시스템의 ~/.parallelcluster/pcluster-cli.log에서 오류 세부 정보를 확인하세요.

## **pcluster describe-cluster** 명령으로 **clusterStatus**가 **UPDATE\_FAILED**로 표시되는 경우

클러스터 스택 업데이트가 롤백된 경우 /var/log/chef-client.logs 파일에서 오류 세부 정보를 확인하세요.

문제가 [GitHub 알려진 문제에](#) 언급되어 있는지 확인하십시오. AWS ParallelCluster GitHub

클러스터 업데이트 제한 시간이 초과되었습니다.

cfn-hup가 실행되지 않는 것과 관련된 문제일 수 있습니다. cfn-hup 대몬(daemon)이 외부 원인으로 종료되면 자동으로 다시 시작되지 않습니다. 가 실행되고 cfn-hup 있지 않으면 클러스터 업데이트 중에 CloudFormation 스택이 예상대로 업데이트 프로세스를 시작하지만 헤드 노드에서 업데이트 절차

가 활성화되지 않아 결국 스택 배포 시간이 초과됩니다. [cfn-hup이 실행 중이 아닐 때의 클러스터 업데이트 제한 시간 문제 해결](#)에서 자세한 내용을 참조하여 문제를 해결하세요.

## 스토리지에 액세스하려는 경우

### 외부 Amazon FSx for Lustre 파일 시스템 사용

클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인합니다. 파일 시스템은 포트 988, 1021, 1022 및 1023을 통한 인바운드 및 아웃바운드 TCP 트래픽을 허용하는 보안 그룹에 연결되어 있어야 합니다. 보안 그룹을 설정하는 방법에 대한 자세한 내용은 [FileSystemsId](#)를 참조하십시오.

### 외부 Amazon Elastic File System 파일 시스템 사용

클러스터와 파일 시스템 간에 트래픽이 허용되는지 확인합니다. 파일 시스템은 포트 988, 1021, 1022 및 1023을 통한 인바운드 및 아웃바운드 TCP 트래픽을 허용하는 보안 그룹에 연결되어 있어야 합니다. 보안 그룹을 설정하는 방법에 대한 자세한 내용은 [FileSystemsId](#)를 참조하십시오.

## 클러스터를 삭제하려는 경우

**pcluster delete-cluster** 명령이 로컬에서 실행되지 않습니다.

로컬 ~/.parallelcluster/pcluster-cli.log 파일 시스템에서 파일을 확인합니다.

클러스터 스택 삭제에 실패했습니다.

클러스터 스택이 삭제되지 않는 경우 CloudFormation 스택 이벤트 메시지를 확인하십시오.

AWS ParallelCluster on의 [GitHub 알려진 문제에](#) 문제가 언급되어 있는지 확인하세요 GitHub.

## AWS ParallelCluster API 스택 업그레이드 시도 중

문제가 [GitHub 알려진 문제에](#) 언급되어 있는지 확인하십시오 GitHub. AWS ParallelCluster

## 컴퓨팅 노드 초기화 오류가 표시되는 경우

**clustermgtd.log**에서 **Node bootstrap error**가 표시되는 경우

이 문제는 컴퓨팅 노드의 부트스트랩 실패와 관련이 있습니다. 클러스터 보호 모드 문제를 디버깅하는 방법에 대한 자세한 내용은 [보호 모드를 디버깅하는 방법](#) 항목을 참조하십시오.

온디맨드 용량 예약(ODCR) 또는 영역별 예약 인스턴스를 구성했습니다.

P4d, P4de, Trn (Trn) 과 같이 여러 네트워크 인터페이스가 있는 인스턴스를 포함하는 ODCR AWS

클러스터 구성 파일에서 HeadNode가 퍼블릭 서브넷에 있고 컴퓨팅 노드가 프라이빗 서브넷에 있는지 확인합니다.

ODCR이 대상으로 지정된 ODCR인 경우

[온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작](#)에 나와 있는 지침을 따라 이미 `/opt/slurm/etc/pcluster/run_instances_overrides.json`를 설치했는데도 **Unable to read file '/opt/slurm/etc/pcluster/run\_instances\_overrides.json'**가 표시되는 경우

대상 ODCR과 함께 AWS ParallelCluster 버전 3.1.1 ~ 3.2.1을 사용하고 [실행 인스턴스 재정의 JSON 파일도 사용하는 경우 JSON 파일 형식이](#) 올바르지 않을 수 있습니다. `clustermgtd.log`에서 다음과 같은 오류가 발생할 수 있습니다.

```
Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'.
Using default: {} in /var/log/parallelcluster/clustermgtd.
```

다음을 실행하여 JSON 파일 형식이 올바른지 확인합니다.

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

클러스터 생성 실패 시 `clustermgtd.log`에서 또는 작업 실행 실패 시 `slurm_resume.log`에서 **Found RunInstances parameters override.**이 표시되는 경우

[실행 인스턴스 재정의 JSON 파일](#)을 사용하는 경우 `/opt/slurm/etc/pcluster/run_instances_overrides.json` 파일에서 대기열 이름과 컴퓨팅 리소스 이름을 올바르게 설정했는지 확인하세요.

작업 실행 실패 시 `slurm_resume.log`에서 또는 클러스터 실행 실패 시 `clustermgtd.log`에서 **An error occurred (InsufficientInstanceCapacity)**이 표시되는 경우

PG-ODCR(배치 그룹 ODCR) 사용

연결된 배치 그룹이 있는 ODCR을 만들 때는 구성 파일에 동일한 배치 그룹 이름을 사용해야 합니다. 클러스터 구성에서 대응하는 [배치 그룹 이름](#)을 설정합니다.

## 영역 예약 인스턴스 사용

클러스터 구성에서 PlacementGroup/Enabled를 true로와 함께 영역 예약 인스턴스를 사용하는 경우 다음과 같은 오류가 표시될 수 있습니다.

We currently do not have sufficient trn1.32xlarge capacity in the Availability Zone you requested (us-east-1d). Our system will be working on provisioning additional capacity.

You can currently get trn1.32xlarge capacity by not specifying an Availability Zone in your request or choosing us-east-1a, us-east-1b, us-east-1c, us-east-1e, us-east-1f.

영역 예약 인스턴스가 동일한 UC(또는 스파인)에 배치되지 않아 이러한 현상이 나타날 수 있으며, 배치 그룹을 사용할 때 용량 부족 오류(ICE)가 발생할 수 있습니다. 클러스터 구성에서 PlacementGroup 그룹 설정을 비활성화하여 클러스터가 인스턴스를 할당할 수 있는지 확인하면 이 경우를 확인할 수 있습니다.

## 작업 실행 실패 시 `slurm_resume.log`에서 또는 클러스터 실행 실패 시 `clustermgtd.log`에서 **An error occurred (VcpuLimitExceeded)**이 표시되는 경우

사용 중인 특정 Amazon EC2 인스턴스 유형에 대한 계정의 vCPU 한도를 확인하십시오. vCPU가 0개 또는 요청한 것보다 더 적으면 한도 증가를 요청하세요. 현재 한도를 확인하고 새 한도를 요청하는 방법에 대한 자세한 내용은 [Amazon EC2 사용 설명서의 Amazon EC2 서비스 할당량을](#) 참조하십시오.

## 작업 실행 실패 시 `slurm_resume.log`에서 또는 클러스터 실행 실패 시 `clustermgtd.log`에서 **An error occurred (InsufficientInstanceCapacity)**이 표시되는 경우

용량 부족 문제가 발생했습니다. <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>를 팔로우하여 문제를 해결하십시오.

## 노드가 **Reason (Code:InsufficientInstanceCapacity)...**으로 **DOWN** 상태로 표시되는 경우

용량 부족 문제가 발생했습니다. <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>를 팔로우하여 문제를 해결하세요. AWS ParallelCluster의 빠른 용량 부족 페일오버 모드에 대한 자세한 내용은 [Slurm 클러스터 빠른 용량 부족 장애 조치](#)를 참조하십시오.

## slurm\_resume.log에서 cannot change locale (en\_US.utf-8) because it has an invalid name가 표시되는 경우

yum 설치 프로세스에 실패하여 로케일 설정이 일관되지 않은 상태로 남아 있는 경우 이 문제가 발생할 수 있습니다. 예를 들어, 사용자가 설치 프로세스를 종료할 때 이러한 문제가 발생할 수 있습니다.

원인을 확인하려면 다음 작업을 수행합니다.

- `su - pcluster-admin`를 실행합니다.

셸에 `cannot change locale...no such file or directory`과 같은 오류가 표시됩니다.

- `localedef --list`를 실행합니다.

빈 목록을 반환하거나 기본 로케일을 포함하지 않습니다.

- `yum history` 및 `yum history info #ID`를 사용하여 마지막 yum 명령을 확인합니다. 마지막 ID에 `Return-Code: Success`가 있나요?

마지막 ID에 `Return-Code: Success`가 없으면 설치 후 스크립트가 성공적으로 실행되지 않았을 수 있습니다.

문제를 해결하려면 `yum reinstall glibc-all-langpacks`를 사용하여 로케일을 다시 빌드해 보세요. 다시 빌드한 후에 문제가 해결됐으면 `su - pcluster-admin`가 오류나 경고를 표시하지 않습니다.

이전 시나리오 중 어느 것도 제 상황에 적용되지 않습니다.

컴퓨팅 노드 초기화 문제를 해결하려면 [노드 초기화 문제 해결](#)을 참조하세요.

현재 AWS ParallelCluster 사용 중인 시나리오가 [GitHub 알려진 문제에](#) 포함되어 있는지 확인해 보십시오. GitHub

추가 지원이 필요하면 [추가 지원](#)을 참조하세요.

## 클러스터 상태 지표 문제 해결

클러스터 상태 지표는 AWS ParallelCluster 버전 3.6.0부터 AWS ParallelCluster Amazon CloudWatch 대시보드에 추가됩니다. 다음 섹션에서는 대시보드 상태 지표와 문제 해결을 위해 취할 수 있는 조치에 대해 알아볼 수 있습니다.



## 주제

- [인스턴스 프로비저닝 오류 그래프 참조](#)
- [비정상 인스턴스 오류 그래프 보기](#)
- [컴퓨팅 플릿 유휴 시간 그래프 보기](#)

## 인스턴스 프로비저닝 오류 그래프 참조

Instance Provisioning Errors 그래프에 0이 아닌 값이 표시되면 슬럼 노드를 지원하는 Amazon EC2 인스턴스가 또는 API에서 시작되지 못했다는 의미입니다. CreateFleet RunInstance

### IAMPolicyErrors 확인

- 어떻게 된 걸까요?

권한이 충분하지 않고 오류 코드 UnauthorizedOperation이 발생하여 여러 인스턴스가 시작되지 못했습니다.

- 해결 방법은?

사용자 지정 [InstanceRole](#) 또는 [InstanceProfile](#)을 구성한 경우, IAM 정책을 확인하고 올바른 보안 인증 정보를 사용하고 있는지 확인하세요.

clustermgtd 파일에서 정적 노드 오류 세부 정보를 확인하세요. slurm\_resume.log 파일에서 동적 노드 오류 세부 정보를 확인하세요. 세부 정보를 사용하여 추가해야 하는 누락된 권한에 대해 자세히 알아보세요.

### VcpuLimitErrors 확인

- 어떻게 된 걸까요?

AWS ParallelCluster 클러스터 컴퓨팅 AWS 계정 노드용으로 구성된 특정 Amazon EC2 인스턴스 유형에 대한 vCPU 한도에 도달하여 인스턴스를 시작하지 못했습니다.

- 해결 방법은?

정적 노드의 경우 clustermgtd 파일에서 VcpuLimitExceeded 오류를 확인하고, 추가 세부 정보를 보려면 동적 노드용 slurm\_resume.log 파일을 확인하세요. 이 문제를 해결하려면 vCPU 한도 증가를 요청할 수 있습니다. 현재 한도를 확인하고 새 한도를 요청하는 방법에 대한 자세한 내용

은 Linux 인스턴스용 Amazon Elastic Compute 클라우드 사용 설명서에서 Amazon Elastic Compute Cloud [서비스 할당량](#)을 참조하십시오.

## VolumeLimitErrors 확인

- 어떻게 된 걸까요?

Amazon EBS 볼륨 제한에 도달했는데 오류 코드 `InsufficientVolumeCapacity` 또는 `VolumeLimitExceeded` 오류가 있는 인스턴스를 시작할 수 없습니다. AWS 계정 AWS ParallelCluster

- 해결 방법은?

`clustermgtd` 파일에 정적 노드가 있는지 확인하고 `slurm_resume.log` 파일에 동적 노드가 있는지 확인하여 추가 볼륨 제한 세부 정보를 확인하세요. 이 문제를 해결하려면 다른 AWS 리전볼륨을 사용하거나 기존 볼륨을 정리하거나 AWS 지원 센터에 문의하여 Amazon EBS 볼륨 제한 증가 요청을 제출할 수 있습니다.

## InsufficientCapacityErrors 확인

- 어떻게 된 걸까요?

AWS ParallelCluster 노드를 지원하기 위해 Amazon EC2 인스턴스를 시작하기에 충분한 용량이 없습니다.

- 해결 방법은?

`clustermgtd` 파일에서 정적 노드가 있는지 확인하고, `slurm_resume.log` 파일에 동적 노드가 있는지 확인하여 용량 부족 오류 세부 정보를 확인하세요. 문제를 해결하려면 <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors> /의 지침을 따르십시오.

## OtherInstanceLaunchFailures

- 어떻게 된 걸까요?

컴퓨팅 노드를 지원하는 Amazon EC2 인스턴스가 또는 API를 사용하여 시작하지 못했습니다. `CreateFleet` `RunInstance`

- 해결 방법은?

clustermgtd 파일에서 정적 노드가 있는지 확인하고, slurm\_resume.log 파일에 동적 노드가 있는지 확인하여 오류 세부 정보를 확인하세요.

## 비정상 인스턴스 오류 그래프 보기

- 어떻게 된 걸까요?

여러 컴퓨팅 인스턴스가 시작되었지만 나중에 비정상적으로 종료되었습니다.

- 해결 방법은?

비정상 노드 문제 해결에 대한 자세한 내용은 [예상치 못한 노드 교체 및 종료 문제 해결](#) 섹션을 참조하세요.

## InstanceBootstrapTimeoutError 확인

- 어떻게 된 걸까요?

인스턴스는 resume\_timeout(동적 노드의 경우) 또는 node\_replacement\_timeout(정적 노드의 경우) 내에서 클러스터에 조인할 수 없습니다. 이는 네트워크가 컴퓨팅 노드에 맞게 올바르게 구성되지 않은 경우 발생할 수 있으며, 컴퓨팅 노드에서 실행되는 사용자 지정 스크립트를 완료하는 데 시간이 너무 오래 걸리는 경우 발생할 수 있습니다.

- 해결 방법은?

동적 노드의 경우 clustermgtd 로그(/var/log/parallelcluster/clustermgtd)에서 컴퓨팅 노드 IP 주소 및 다음과 같은 오류를 확인합니다.

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

정적 노드의 경우 clustermgtd 로그(/var/log/parallelcluster/clustermgtd)에서 컴퓨팅 노드 IP 주소 및 다음과 같은 오류를 확인합니다.

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

더 자세한 내용은 /var/log/cloud-init-output.log 파일에서 오류를 확인하세요.

clustermgtd 및 slurm\_resume 로그 파일에서 문제가 있는 컴퓨팅 노드 IP 주소를 검색할 수 있습니다.

## EC2HealthCheckErrors 확인

- 어떻게 된 걸까요?

인스턴스가 Amazon EC2 상태 확인에 실패했습니다.

- 해결 방법은?

이 문제를 해결하는 방법에 대한 자세한 내용은 [상태 확인에 실패한 인스턴스 문제 해결](#)을 참조하세요.

## ScheduledEventHealthCheckErrors 확인

- 어떻게 된 걸까요?

인스턴스가 Amazon EC2 예약 이벤트 상태 점검에 실패하여 비정상입니다.

- 해결 방법은?

이 문제를 해결하는 방법에 대한 자세한 내용은 [인스턴스의 예약 이벤트를](#) 참조하세요.

## NoCorrespondingInstanceErrors 확인

- 어떻게 된 걸까요?

AWS ParallelCluster 노드를 지원하는 인스턴스를 찾을 수 없습니다. 부트스트랩 작업 중에 노드가 자체 종료되었을 수 있습니다.

[SlurmQueues/CustomActions/OnNodeStart|OnNodeConfigured](#) 스크립트 또는 네트워크 오류가 NoCorrespondingInstanceErrors를 발생시킬 수 있습니다.

- 해결 방법은?

자세한 내용은 컴퓨팅 노드의 `/var/log/cloud-init-output.log`을 확인하세요.

## 컴퓨팅 플릿 유휴 시간 그래프 보기

**MaxDynamicNodeIdleTime**이 유휴 시간 스케일다운 임계값보다 훨씬 긴 것으로 확인됨

- 어떻게 된 걸까요?

인스턴스가 제대로 종료되지 않습니다. MaxDynamicNodeIdleTimeAmazon EC2 인스턴스가 지원하는 동적 노드가 유휴 상태인 최대 시간 (초) 을 보여 줍니다. 유휴 시간 스케일다운 임계값은 클러스터 구성 [ScaledownIdletime](#) 파라미터에서 파생됩니다. 컴퓨팅 노드가 유휴 시간 (Idle Time Scaledown) 초 이상 유휴 상태가 지속되면 노드의 Slurm 전원을 끄고 지원 인스턴스를 종료합니다. AWS ParallelCluster 이 경우, 무언가 인스턴스 종료를 방해하고 있습니다.

- 해결 방법은?

이 문제에 대한 자세한 내용은 [규모 조정 문제 해결](#)에서 [문제가 있는 인스턴스 및 노드 교체, 종료 또는 전원 끄기](#)를 참조하세요.

## 클러스터 배포 문제 해결

클러스터 생성에 실패하고 스택 생성을 롤백하는 경우 로그 파일을 살펴보고 문제를 진단할 수 있습니다. 실패 메시지는 다음 출력과 같을 수 있습니다.

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
--cluster-configuration cluster-config.yaml
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}

$ pcluster describe-cluster --cluster-name mycluster --region eu-west-1
{
  "creationTime": "2021-09-06T11:03:47.696Z",
  ...
  "cloudFormationStackStatus": "ROLLBACK_IN_PROGRESS",
  "clusterName": "mycluster",
  "computeFleetStatus": "UNKNOWN",
  "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
  "lastUpdatedTime": "2021-09-06T11:03:47.696Z",
  "region": "eu-west-1",
```

```
"clusterStatus": "CREATE_FAILED"
}
```

## 주제

- [에서 AWS CloudFormation 이벤트 보기 CREATE\\_FAILED](#)
- [CLI를 사용하여 로그 스트림을 볼 수 있습니다.](#)
- [rollback-on-failure을 사용하여 실패한 클러스터를 다시 생성합니다.](#)

## 에서 AWS CloudFormation 이벤트 보기 **CREATE\_FAILED**

콘솔 또는 AWS ParallelCluster CLI를 사용하여 CREATE\_FAILED 오류 CloudFormation 이벤트를 확인하여 근본 원인을 찾을 수 있습니다.

## 주제

- [콘솔에서 CloudFormation 이벤트 보기](#)
- [CLI를 사용하여 이벤트를 보고 CloudFormation 필터링할 수 있습니다. CREATE\\_FAILED](#)

## 콘솔에서 CloudFormation 이벤트 보기

"CREATE\_FAILED"상태 발생의 원인에 대한 자세한 내용을 보려면 CloudFormation 콘솔을 사용할 수 있습니다.

콘솔에서 CloudFormation 오류 메시지를 볼 수 있습니다.

1. [에 AWS Management Console 로그인하고 <https://console.aws.amazon.com/cloudformation> 으로 이동합니다.](#)
2. 이름이 *cluster\_name*인 스택을 선택합니다.
3. 이벤트 탭을 선택합니다.
4. 논리적 ID별로 리소스 이벤트 목록을 스크롤하여 생성에 실패한 리소스의 상태를 확인합니다. 하위 작업을 만들지 못한 경우 역방향으로 진행하여 실패한 리소스 이벤트를 찾아보세요.
5. 예를 들어, 다음 상태 메시지가 표시되면 현재 vCPU 한도를 초과하지 않는 인스턴스 유형을 사용하거나 vCPU 용량을 더 요청해야 합니다.

```
2022-02-04 16:09:44 UTC-0800 HeadNode CREATE_FAILED You have requested more vCPU
capacity than your current vCPU limit of 0 allows
```

for the instance bucket that the specified instance type belongs to. Please visit <http://aws.amazon.com/contact-us/ec2-request> to request an adjustment to this limit.

(Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request ID: a9876543-b321-c765-d432-dcba98766789; Proxy: null).

CLI를 사용하여 이벤트를 보고 CloudFormation 필터링할 수 있습니다.

## CREATE\_FAILED

클러스터 생성 문제를 진단하려면 CREATE\_FAILED 상태를 필터링하여 [pcluster get-cluster-stack-events](#) 명령을 사용할 수 있습니다. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI [출력 필터링](#)을 참조하십시오.

```
$ pcluster get-cluster-stack-events --cluster-name mycluster --region eu-west-1 \
  --query 'events[?resourceStatus==`CREATE_FAILED`]'
[
  {
    "eventId": "3ccdedd0-0f03-11ec-8c06-02c352fe2ef9",
    "physicalResourceId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "The following resource(s) failed to create: [HeadNode].",
  },
  {
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "mycluster",
    "resourceType": "AWS::CloudFormation::Stack",
    "timestamp": "2021-09-06T11:11:51.780Z"
  },
  {
    "eventId": "HeadNode-CREATE_FAILED-2021-09-06T11:11:50.127Z",
    "physicalResourceId": "i-04e91cc1f4ea796fe",
    "resourceStatus": "CREATE_FAILED",
    "resourceStatusReason": "Received FAILURE signal with UniqueId
i-04e91cc1f4ea796fe",
    "resourceProperties": "{\"LaunchTemplate\":{\"Version\":\"1\",\"LaunchTemplateId
\":\"lt-057d2b1e687f05a62\"}}",
    "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
    "stackName": "mycluster",
    "logicalResourceId": "HeadNode",
```

```

    "resourceType": "AWS::EC2::Instance",
    "timestamp": "2021-09-06T11:11:50.127Z"
  }
]

```

이전 예제에서는 헤드 노드 설정에 장애가 발생했습니다.

## CLI를 사용하여 로그 스트림을 볼 수 있습니다.

이러한 문제를 디버깅하려면 [pcluster list-cluster-log-streams](#)으로 node-type를 필터링한 다음 로그 스트림 콘텐츠를 분석하여 헤드 노드에서 이용 가능한 로그 스트림을 나열하는 방법이 있습니다.

```

$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
--filters 'Name=node-type,Values=HeadNode'
{
  "logStreams": [
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
      ...
    },
    {
      "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
      ...
    },
    ...
  ]
}

```

초기화 오류를 찾는 데 사용할 수 있는 두 가지 기본 로그 스트림은 다음과 같습니다.



- `cfn-init`은 `cfn-init` 스크립트의 로그입니다. 먼저 이 로그 스트림을 확인합니다. 이 로그에서 `Command chef failed` 오류를 확인할 수 있을 것입니다. 오류 메시지와 관련된 자세한 내용은 이 라인 바로 앞에 있는 라인을 참조하세요. 자세한 내용은 [cfn-init](#)을 참조하세요.
- `cloud-init`은 [cloud-init](#)에 대한 로그입니다. `cfn-init`에 아무것도 표시되지 않으면 다음으로 이 로그를 확인해 보세요.

[pcluster get-cluster-log-events](#)을 사용하여 로그 스트림의 콘텐츠를 검색할 수 있습니다(검색되는 이벤트 수를 제한하는 `--limit 5` 옵션 참고).

```
$ pcluster get-cluster-log-events --cluster-name mycluster \
  --region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init \
  --limit 5
{
  "nextToken": "f/36370880979637159565202782352491087067973952362220945409/s",
  "prevToken": "b/36370880752972385367337528725601470541902663176996585497/s",
  "events": [
    {
      "message": "2021-09-06 11:11:39,049 [ERROR] Unhandled exception during build:
Command runpostinstall failed",
      "timestamp": "2021-09-06T11:11:39.049Z"
    },
    {
      "message": "Traceback (most recent call last):\n File \"/opt/aws/bin/
cfn-init\", line 176, in <module>\n   worklog.build(metadata, configSets)\n
File \"/usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line
135, in build\n   Contractor(metadata).build(configSets, self)\n File \"/
usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 561, in
build\n   self.run_config(config, worklog)\n File \"/usr/lib/python3.7/
site-packages/cfnbootstrap/construction.py\", line 573, in run_config\n
CloudFormationCarpenter(config, self._auth_config).build(worklog)\n File \"/usr/
lib/python3.7/site-packages/cfnbootstrap/construction.py\", line 273, in build\n
self._config.commands)\n File \"/usr/lib/python3.7/site-packages/cfnbootstrap/
command_tool.py\", line 127, in apply\n   raise ToolError(u\"Command %s failed\" %
name)\",
      "timestamp": "2021-09-06T11:11:39.049Z"
    },
    {
      "message": "cfnbootstrap.construction_errors.ToolError: Command runpostinstall
failed",
      "timestamp": "2021-09-06T11:11:39.049Z"
    },
  ]
}
```

```

{
  "message": "2021-09-06 11:11:49,212 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.eu-west-1.amazonaws.com",
  "timestamp": "2021-09-06T11:11:49.212Z"
},
{
  "message": "2021-09-06 11:11:49,213 [DEBUG] Signaling resource HeadNode in stack
mycluster with unique ID i-04e91cc1f4ea796fe and status FAILURE",
  "timestamp": "2021-09-06T11:11:49.213Z"
}
]
}

```

이전 예제에서 실패는 `runpostinstall` 실패로 인해 발생했으므로 이 오류는 [CustomActions](#)의 `OnNodeConfigured` 구성 파라미터에 사용된 사용자 지정 부트스트랩 스크립트의 내용과 엄격하게 관련되어 있습니다.

**rollback-on-failure**를 사용하여 실패한 클러스터를 다시 생성합니다.

AWS ParallelCluster 로그 그룹에 클러스터 CloudWatch 로그 스트림을 생성합니다. CloudWatch 콘솔 사용자 지정 대시보드 또는 로그 그룹에서 이러한 로그를 볼 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs와 통합](#) 및 [아마존 CloudWatch 대시보드](#) 섹션을 참조하세요. 사용 가능한 로그 스트림이 없는 경우 [CustomActions](#) 사용자 지정 부트스트랩 스크립트 또는 AMI 관련 문제로 인해 오류가 발생할 수 있습니다. 이 경우 생성 문제를 진단하려면 `false`로 설정된 `--rollback-on-failure` 파라미터를 포함하여 [pcluster create-cluster](#)를 사용하여 클러스터를 다시 생성하세요. 그 후 다음과 같이 SSH를 사용하여 클러스터를 확인합니다.

```

$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
  --cluster-configuration cluster-config.yaml --rollback-on-failure false
{
  "cluster": {
    "clusterName": "mycluster",
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
    "region": "eu-west-1",
    "version": "3.7.0",
    "clusterStatus": "CREATE_IN_PROGRESS"
  }
}
$ pcluster ssh --cluster-name mycluster

```

헤드 노드에 로그인한 후에는 오류를 찾는 데 사용할 수 있는 세 개의 기본 로그 파일을 찾을 수 있습니다.

- `/var/log/cfn-init.log`은 `cfn-init` 스크립트의 로그입니다. 먼저 이 로그를 확인하세요. 이 로그에서 `Command chef failed` 같은 오류가 표시될 수 있습니다. 오류 메시지와 관련된 자세한 내용은 이 라인 바로 앞에 있는 라인을 참조하세요. 자세한 내용은 [cfn-init](#)을 참조하세요.
- `/var/log/cloud-init.log`은 [cloud-init](#)에 대한 로그입니다. `cfn-init.log`에 아무것도 표시되지 않으면 다음으로 이 로그를 확인해 보세요.
- `/var/log/cloud-init-output.log`은 [cloud-init](#)이 실행한 명령의 출력입니다. 여기에는 `cfn-init`의 출력이 포함됩니다. 대부분의 경우 이러한 유형의 문제를 해결하기 위해 이 로그를 볼 필요가 없습니다.

## Terraform을 사용한 클러스터 배포 문제 해결

이 섹션은 Terraform을 사용하여 배포된 클러스터와 관련이 있습니다.

### ParallelCluster API를 찾을 수 없습니다.

ParallelCluster API를 찾을 수 없어 계획이 실패할 수 있습니다. 이 경우 반환되는 오류는 다음과 같습니다.

```
Planning failed. Terraform encountered an error while generating this plan.

#
# Error: Unable to retrieve ParallelCluster API cloudformation stack.
#
#   with provider["registry.terraform.io/aws-tf/aws-parallelcluster"],
#   on providers.tf line 6, in provider "aws-parallelcluster":
#     6: provider "aws-parallelcluster" {
#
# operation error CloudFormation: DescribeStacks, https response error StatusCode: 400,
RequestID: REQUEST_ID, api error ValidationError: Stack with id PCAPI_STACK_NAME does
not exist
```

이 오류를 해결하려면 클러스터를 생성할 계정에 ParallelCluster API를 배포하세요. [the section called “테라폼으로 클러스터 만들기”](#) 섹션을 참조하십시오.

## 사용자는 ParallelCluster API를 호출할 권한이 없습니다.

Terraform 프로젝트를 배포하기로 가정한 IAM 역할/사용자에게 API와 상호 작용할 권한이 없기 때문에 계획이 실패할 수 있습니다. ParallelCluster 이 경우 반환되는 오류는 다음과 같습니다.

```
Planning failed. Terraform encountered an error while generating this plan.
```

```
# Error: 403 Forbidden
#
#   with
#     module.parallelcluster_clusters.module.clusters[0].pcluster_cluster.managed_configs["DemoCluster"],
#   on .terraform/modules/parallelcluster_clusters/modules/clusters/main.tf line 35, in
#   resource "pcluster_cluster" "managed_configs":
#     35: resource "pcluster_cluster" "managed_configs" {
#
#   [{"Message": "User: USER_ARN is not authorized to perform: execute-api:Invoke on
#     resource: PC_API_REST_RESOURCE with an explicit deny"}]
# }
```

이 오류를 해결하려면 ParallelCluster 제공자가 ParallelCluster API 역할을 사용하여 API와 상호 작용하도록 구성하십시오.

```
provider "aws-parallelcluster" {
  region      = var.region
  profile     = var.profile
  api_stack_name = var.api_stack_name
  **use_user_role** **= true**
}
```

## 규모 조정 문제 해결

이 섹션은 Slurm 작업 스케줄러와 함께 AWS ParallelCluster 버전 3.0.0 이상을 사용하여 설치된 클러스터와 관련이 있습니다. 다중 대기열 구성에 대한 자세한 내용은 [다중 대기열 구성](#) 섹션을 참조하십시오.

실행 중인 클러스터 중 하나에 문제가 있는 경우 문제 해결을 시작하기 전에 다음 명령을 실행하여 클러스터를 STOPPED 상태로 전환하십시오. 이렇게 하면 예상치 못한 비용이 발생하는 것을 방지할 수 있습니다.

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

```
--status STOP_REQUESTED
```

[pcluster list-cluster-log-streams](#) 명령을 사용하고 헤드 노드 또는 실패한 노드 중 하나의 `private-dns-name`로 필터링하여 클러스터 노드에서 이용 가능한 로그 스트림을 나열할 수 있습니다.

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
--filters 'Name=private-dns-name,Values=ip-10-0-0-101'
```

그다음 [pcluster get-cluster-log-events](#) 명령을 사용하고 다음 섹션에서 언급된 키 로그 중 하나에 대응하는 `--log-stream-name`를 전달하여 로그 스트림의 내용을 검색하고 분석할 수 있습니다.

```
$ pcluster get-cluster-log-events --cluster-name mycluster \
--region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init
```

AWS ParallelCluster 클러스터 CloudWatch 로그 스트림을 로그 그룹에 생성합니다. CloudWatch 콘솔 사용자 지정 대시보드 또는 로그 그룹에서 이러한 로그를 볼 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs와 통합](#) 및 [아마존 CloudWatch 대시보드](#) 섹션을 참조하세요.

## 주제

- [디버깅을 위한 키 로그](#)
- [작업 실행 실패 시 slurm\\_resume.log에서 또는 클러스터 실행 실패 시 clustermgtd.log에서 InsufficientInstanceCapacity 오류가 표시되는 경우](#)
- [노드 초기화 문제 해결](#)
- [예상치 못한 노드 교체 및 종료 문제 해결](#)
- [문제가 있는 인스턴스 및 노드 교체, 종료 또는 전원 끄기](#)
- [대기열\(파티션\) Inactive 상태](#)
- [기타 알려진 노드 및 작업 문제 해결](#)

## 디버깅을 위한 키 로그

다음 표에서는 헤드 노드의 키 로그 개요를 제공합니다.

- `/var/log/cfn-init.log`- 이것은 AWS CloudFormation init 로그입니다. 여기에는 인스턴스가 설정될 때 실행된 모든 명령이 들어 있습니다. 이를 사용하여 초기화 문제를 해결할 수 있습니다.

- `/var/log/chef-client.log` - Chef 클라이언트 로그입니다. 여기에는 Chef/CINC를 통해 실행된 모든 명령이 포함됩니다. 이를 사용하여 초기화 문제를 해결할 수 있습니다.
- `/var/log/parallelcluster/slurm_resume.log` - 이것은 ResumeProgram 로그입니다. 동적 노드의 인스턴스를 시작합니다. 이를 사용하여 동적 노드 시작 문제를 해결할 수 있습니다.
- `/var/log/parallelcluster/slurm_suspend.log` - 이것은 SuspendProgram 로그입니다. 동적 노드의 인스턴스가 종료될 때 호출됩니다. 이를 사용하여 동적 노드 종료 문제를 해결할 수 있습니다. 이 로그를 확인할 때는 `clustermgtd` 로그도 확인해야 합니다.
- `/var/log/parallelcluster/clustermgtd` - 이것은 `clustermgtd` 로그입니다. 이 데몬(daemon)은 대부분의 클러스터 작업 작업을 관리하는 중앙 대몬으로 실행됩니다. 이를 사용하여 시작, 종료 또는 클러스터 운영 문제를 해결할 수 있습니다.
- `/var/log/slurmctld.log` - Slurm 컨트롤 데몬 로그입니다. AWS ParallelCluster 규모 조정 결정을 내리지 않습니다. 오히려 Slurm 요구 사항을 충족하는 리소스를 시작하려고 시도할 뿐입니다. 규모 조정 및 할당 문제, 작업 관련 문제, 스케줄러 관련 시작 및 종료 문제에 유용합니다.
- `/var/log/parallelcluster/compute_console_output` - 이 로그는 예기치 않게 종료된 정적 컴퓨팅 노드 샘플 하위 집합의 콘솔 출력을 기록합니다. 정적 컴퓨팅 노드가 종료되고 컴퓨팅 노드 로그를 사용할 수 없는 경우 이 로그를 사용하십시오 CloudWatch. 수신하는 `compute_console_output` log 콘텐츠는 Amazon EC2 콘솔을 사용하거나 인스턴스 콘솔 출력을 검색할 때 동일합니다. AWS CLI

컴퓨팅 노드의 키 로그는 다음과 같습니다.

- `/var/log/cloud-init-output.log` - [cloud-init](#) 로그입니다. 여기에는 인스턴스가 설정될 때 실행된 모든 명령이 들어 있습니다. 이를 사용하여 초기화 문제를 해결할 수 있습니다.
- `/var/log/parallelcluster/computemgtd` - 이것은 `computemgtd` 로그입니다. 이것은 헤드 노드의 `clustermgtd` 데몬(daemon)이 오프라인 상태인 드문 상황에서 각 컴퓨팅 노드에서 실행되어 노드를 모니터링합니다. 이를 사용하여 예상치 못한 종료 문제를 해결할 수 있습니다.
- `/var/log/slurmd.log` - Slurm 컴퓨팅 데몬 로그입니다. 이를 사용하여 초기화 및 컴퓨팅 실패 문제를 해결할 수 있습니다.

## 작업 실행 실패 시 `slurm_resume.log`에서 또는 클러스터 실행 실패 시 `clustermgtd.log`에서 `InsufficientInstanceCapacity` 오류가 표시 되는 경우

클러스터에서 Slurm 스케줄러를 사용하는 경우 용량 부족 문제가 발생하는 것입니다. 인스턴스 시작이 요청되었을 때 사용 가능한 인스턴스가 부족하면 `InsufficientInstanceCapacity` 오류가 반환 됩니다.

정적 인스턴스 용량의 경우 `/var/log/parallelcluster/clustermgtd`의 `clustermgtd` 로그에서 오류를 찾을 수 있습니다.

동적 인스턴스 용량의 경우 `/var/log/parallelcluster/slurm_resume.log`의 `ResumeProgram` 로그에서 오류를 찾을 수 있습니다.

메시지는 다음 예제와 유사합니다.

```
An error occurred (InsufficientInstanceCapacity) when calling the RunInstances/
CreateFleet operation...
```

사용 사례에 따라 다음 방법 중 하나를 사용하여 이러한 유형의 오류 메시지가 나타나지 않도록 하세요.

- 배치 그룹이 활성화되어 있는 경우 해당 그룹을 비활성화하세요. 자세한 내용은 [배치 그룹 및 인스턴스 시작 문제](#) 항목을 참조하세요.
- 인스턴스의 용량을 예약하고 ODCR(온디맨드 용량 예약)로 인스턴스를 시작합니다. 자세한 내용은 [온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작](#) 항목을 참조하세요.
- 다양한 인스턴스 유형으로 여러 컴퓨팅 리소스를 구성합니다. 워크로드에 특정 인스턴스 유형이 필요하지 않은 경우 여러 컴퓨팅 리소스로 빠르게 부족한 용량 장애 조치를 활용할 수 있습니다. 자세한 내용은 [Slurm 클러스터 빠른 용량 부족 장애 조치](#) 항목을 참조하세요.
- 동일한 컴퓨팅 리소스에서 여러 인스턴스 유형을 구성하고 다중 인스턴스 유형 할당을 활용하세요. 다중 인스턴스 구성에 관한 자세한 내용은 [Slurm을 사용하여 여러 인스턴스 유형 할당 및 Scheduling/SlurmQueues/ComputeResources/Instances](#)를 참조하세요.
- 클러스터 구성 [Scheduling/SlurmQueues/Networking/SubnetIds](#) 안의 서브넷 ID를 변경하여 대기열을 다른 가용 영역으로 옮기세요.
- 워크로드가 밀접하게 연결되지 않은 경우 대기열을 여러 가용 영역에 분산시키세요. 다중 서브넷 구성에 관한 자세한 내용은 [Scheduling/SlurmQueues/Networking/SubnetIds](#)을 참조하세요.

## 노드 초기화 문제 해결

이 섹션에서는 노드 초기화 문제를 해결하는 방법을 다룹니다. 여기에는 노드가 시작, 전원 공급 또는 클러스터 조인에 실패하는 문제가 포함됩니다.

주제

- [헤드 노드](#)
- [컴퓨팅 노드](#)

### 헤드 노드

적용 가능한 로그:

- /var/log/cfn-init.log
- /var/log/chef-client.log
- /var/log/parallelcluster/clustermgtd
- /var/log/parallelcluster/slurm\_resume.log
- /var/log/slurmctld.log

/var/log/cfn-init.log 및 /var/log/chef-client.log 로그 또는 대응하는 로그 스트림을 확인합니다. 이 로그에는 헤드 노드가 설정되었을 때 실행된 모든 작업이 포함됩니다. 설정 중에 발생하는 대부분의 오류에는 /var/log/chef-client.log 로그에 오류 메시지가 있을 것입니다. OnNodeStart 또는 OnNodeConfigured 스크립트가 클러스터의 구성에 지정되어 있으면 로그 메시지를 통해 스크립트가 성공적으로 실행되는지 다시 확인하세요.

클러스터를 생성할 때 헤드 노드는 컴퓨팅 노드가 클러스터에 조인할 때까지 기다려야 클러스터에 조인할 수 있습니다. 따라서 컴퓨팅 노드가 클러스터에 조인하지 못하면 헤드 노드도 조인하지 못합니다. 사용하는 컴퓨팅 노드 유형에 따라 다음 일련의 절차 중 하나를 수행하여 이러한 유형의 문제를 해결할 수 있습니다.

### 컴퓨팅 노드

- 적용 가능한 로그:
  - /var/log/cloud-init-output.log
  - /var/log/slurmd.log



- 컴퓨팅 노드가 시작된 경우 먼저 `/var/log/cloud-init-output.log`를 확인하세요. 헤드 노드의 `/var/log/chef-client.log`과 비슷한 설정 로그가 들어 있을 것입니다. 설정 중에 발생하는 대부분의 오류에는 `/var/log/cloud-init-output.log` 로그에 오류 메시지가 있을 것입니다. 클러스터 구성에 사전 설치 또는 설치 후 스크립트가 지정된 경우 해당 스크립트가 성공적으로 실행되었는지 확인하세요.
- Slurm 구성을 수정하여 사용자 지정 AMI를 사용하는 경우 컴퓨팅 노드가 클러스터에 조인하지 못하게 하는 Slurm 관련 오류가 있을 수 있습니다. 스케줄러 관련 오류의 경우 `/var/log/slurmd.log` 로그를 확인하세요.

#### 동적 컴퓨팅 노드:

- ResumeProgram 로그(`/var/log/parallelcluster/slurm_resume.log`)에서 컴퓨팅 노드 이름을 검색하여 해당 노드와 함께 ResumeProgram이 직접 호출된 적이 있는지 확인합니다. (한 번도 호출되지 ResumeProgram 않았다면 `slurmctld log` (`/var/log/slurmctld.log`) 를 확인하여 해당 노드로 ResumeProgram 호출을 시도한 Slurm 적이 있는지 확인할 수 있습니다.)
- 권한이 올바르지 않으면 ResumeProgram가 ResumeProgram를 자동으로 실패하게 할 수 있습니다. ResumeProgram 설정을 수정하여 사용자 지정 AMI를 사용하는 경우 `slurm` 사용자가 ResumeProgram를 소유하고 있으며 `744(rwxr--r--)` 권한이 있는지 확인하세요.
- ResumeProgram가 호출되면 해당 노드에 대한 인스턴스가 시작되었는지 확인하세요. 시작된 인스턴스가 없는 경우 시작 실패를 설명하는 오류 메시지가 표시될 수 있습니다.
- 인스턴스가 시작된 경우 설정 프로세스 중에 문제가 있을 수 있습니다. ResumeProgram 로그에서 해당 프라이빗 IP 주소와 인스턴스 ID를 확인할 수 있습니다. 또한 특정 인스턴스의 대응하는 설정 로그를 볼 수 있습니다. 컴퓨팅 노드 설정 오류의 문제를 해결하는 방법에 대한 자세한 내용은 다음 섹션을 참조하세요.

#### 정적 컴퓨팅 노드:

- `clustermgtd`(`/var/log/parallelcluster/clustermgtd`) 로그를 확인하여 해당 노드의 인스턴스가 시작되었는지 확인합니다. 시작되지 않은 경우 시작 실패를 자세히 설명하는 명확한 오류 메시지가 표시될 것입니다.
- 인스턴스가 시작되면 설정 프로세스 중에 몇 가지 문제가 있습니다. ResumeProgram 로그에서 해당 프라이빗 IP 주소와 인스턴스 ID를 확인할 수 있습니다. 또한 특정 인스턴스의 대응하는 설정 로그를 볼 수 있습니다.

#### 스팟 인스턴스가 지원하는 컴퓨팅 노드:

- 스팟 인스턴스를 처음 사용하고 작업이 PD(보류 중) 상태인 경우 `/var/log/parallelcluster/slurm_resume.log` 파일을 다시 확인하세요. 아마도 다음과 같은 오류가 표시될 것입니다.

```
2022-05-20 13:06:24,796 - [slurm_plugin.common:add_instances_for_nodes] - ERROR -
Encountered exception when launching instances for nodes (x1) ['spot-dy-t2micro-2']:
An error occurred (AuthFailure.ServiceLinkedRoleCreationNotPermitted) when calling
the RunInstances operation: The provided credentials do not have permission to
create the service-linked role for Amazon EC2 Spot Instances.
```

스팟 인스턴스를 사용할 경우 계정에 `AWSServiceRoleForEC2Spot` 서비스 연결 역할이 있어야 합니다. 를 사용하여 계정에서 이 역할을 만들려면 다음 명령어를 실행합니다. AWS CLI

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

자세한 내용은 Amazon EC2 사용 설명서의 [스팟 인스턴스 요청에 대한 AWS ParallelCluster 사용 설명서 및 서비스 연결 역할을 참조하십시오](#) [스팟 인스턴스 작업](#).

## 예상치 못한 노드 교체 및 종료 문제 해결

이 섹션에서는 특히 노드가 예기치 않게 교체되거나 종료되는 경우 노드 관련 문제를 해결하는 방법을 계속 살펴봅니다.

- 적용 가능한 로그:
  - `/var/log/parallelcluster/clustermgtd` (헤드 노드)
  - `/var/log/slurmctld.log` (헤드 노드)
  - `/var/log/parallelcluster/computemgtd` (컴퓨팅 노드)

노드가 예기치 않게 교체되거나 종료됨

- `clustermgtd` 로그(`/var/log/parallelcluster/clustermgtd`)를 확인하여 `clustermgtd`가 노드를 교체 또는 종료했는지 확인합니다. `clustermgtd`가 모든 일반적인 노드 유지 관리 작업을 처리한다는 점에 유의하세요.
- `clustermgtd`가 노드를 교체하거나 종료한 경우 해당 노드를 그렇게 처리한 이유를 설명하는 메시지가 있을 것입니다. 이유가 스케줄러와 관련된 경우(예: 노드가 DOWN에 있기 때문) `slurmctld` 로그에서 자세한 내용을 확인하세요. 이유가 Amazon EC2와 관련된 것이라면 교체가 필요한 Amazon EC2 관련 문제를 자세히 설명하는 정보 메시지가 있을 것입니다.

- 노드를 종료하지 `clustermgtd` 않은 경우 먼저 Amazon EC2에 의한 예상 종료, 특히 스팟 종료 인지 확인하십시오. `computemgtd` 컴퓨팅 노드에서 실행 중일 때도 비정상적으로 `clustermgtd` 판단되면 노드를 종료할 수 있습니다. `computemgtd` 로그(`/var/log/parallelcluster/computemgtd`)를 확인하여 `computemgtd`이 노드를 종료했는지 확인하세요.

### 노드에 장애가 발생한 경우

- `slurmctld` 로그(`/var/log/slurmctld.log`)를 확인하여 작업이나 노드가 실패한 이유를 확인하세요. 단, 노드에 장애가 발생하면 작업이 자동으로 다시 대기열에 추가된다는 점에 유의하세요.
- `slurm_resume`이 해당 노드가 시작되었다고 보고하고 `clustermgtd`가 몇 분 후에 Amazon EC2에 해당 노드에 대응하는 인스턴스가 없다고 보고하면 설정 중에 노드가 실패할 수 있습니다. 컴퓨팅(`/var/log/cloud-init-output.log`)에서 로그를 검색하려면 다음 단계를 따르세요.
  - Slurm가 새 노드를 가동할 수 있도록 작업을 제출하세요.
  - 컴퓨팅 노드가 시작될 때까지 기다리십시오.
  - 장애가 발생한 컴퓨팅 노드가 종료되지 않고 중지되도록 인스턴스 시작 종료 동작을 수정하십시오.

```
$ aws ec2 modify-instance-attribute \
  --instance-id i-1234567890abcdef0 \
  --instance-initiated-shutdown-behavior "{\"Value\": \"stop\"}"
```

- 종료 방지 기능을 활성화합니다.

```
$ aws ec2 modify-instance-attribute \
  --instance-id i-1234567890abcdef0 \
  --disable-api-termination
```

- 쉽게 식별할 수 있도록 노드에 태그를 지정합니다.

```
$ aws ec2 create-tags \
  --resources i-1234567890abcdef0 \
  --tags Key=Name,Value=QUARANTINED-Compute
```

- 태그를 변경하여 클러스터에서 노드를 분리합니다. `parallelcluster:cluster-name`

```
$ aws ec2 create-tags \
  --resources i-1234567890abcdef0 \
  --tags Key=parallelcluster:clustername,Value=QUARANTINED-ClusterName
```

- 이 명령을 사용하여 노드에서 콘솔 출력을 검색합니다.

```
$ aws ec2 get-console-output --instance-id i-1234567890abcdef0 --output text
```

## 문제가 있는 인스턴스 및 노드 교체, 종료 또는 전원 끄기

- 적용 가능한 로그:
  - /var/log/parallelcluster/clustermgtd (헤드 노드)
  - /var/log/parallelcluster/slurm\_suspend.log (헤드 노드)
- 대부분의 경우 clustermgtd가 모든 예상 인스턴스 종료 작업을 처리합니다. clustermgtd 로그에서 노드 교체 또는 종료에 실패한 이유를 확인하세요.
- 동적 노드에 [SlurmSettings 속성](#) 장애가 발생한 경우 SuspendProgram 로그를 확인하여 특정 노드를 인수로 사용하여 SuspendProgram이 slurmctld에 의해 직접 호출되었는지 확인하세요. SuspendProgram는 실제로 어떤 작업도 수행하지 않습니다. 그보다는 직접 호출될 때만 로그를 기록합니다. 모든 인스턴스 종료 및 NodeAddr 재설정은 clustermgtd에 의해 수행됩니다. Slurm은 SuspendTimeout 이후에 노드를 자동으로 POWER\_SAVING 상태로 되돌립니다.
- 부트스트랩 장애로 인해 컴퓨팅 노드에 계속 장애가 발생하는 경우, [Slurm 클러스터 보호 모드](#)가 활성화된 상태로 시작되는지 확인하세요. 보호 모드가 활성화되지 않은 경우 보호 모드 설정을 수정하여 보호 모드를 활성화하세요. 부트스트랩 스크립트 문제 해결 및 수정

## 대기열(파티션) **Inactive** 상태

sinfo를 실행했을 때 출력에 AVAIL 상태가 `inact`인 대기열이 표시되면 클러스터에 [Slurm 클러스터 보호 모드](#)가 활성화되어 있고 대기열이 사전 정의된 기간 동안 INACTIVE 상태로 설정되어 있었을 수 있습니다.

## 기타 알려진 노드 및 작업 문제 해결

알려진 또 다른 유형의 문제는 작업을 할당하지 못하거나 규모 조정 결정을 내리지 AWS ParallelCluster 못할 수 있다는 것입니다. 이러한 유형의 문제에서는 AWS ParallelCluster 가 Slurm 지침에 따라서만 리소스를 시작, 종료 또는 유지 관리합니다. 이러한 문제의 경우 slurmctld 로그를 확인하여 문제를 해결하세요.

## 배치 그룹 및 인스턴스 시작 문제

노드 간 지연 시간을 최소화하려면 배치 그룹을 사용하세요. 배치 그룹은 인스턴스가 동일한 네트워크 백본에 위치하도록 보장합니다. 요청이 이루어질 때 사용 가능한 인스턴스가 충분하지 않으면 `InsufficientInstanceCapacity` 오류가 반환됩니다. 클러스터 배치 그룹을 사용할 때 오류가 발생할 가능성을 줄이려면 [SlurmQueues/Networking/PlacementGroup/Enabled](#) 파라미터를 `false`로 설정합니다.

용량 액세스를 추가로 제어하려면 [ODCR\(온디맨드 용량 예약\)](#)으로 인스턴스를 시작하는 것을 고려하세요.

자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 시작 문제 해결 및 배치 그룹 역할 및 제한](#)을 참조하세요.

## 교체할 수 없는 디렉터리

다음 디렉터리는 노드 간에 공유되므로 교체할 수 없습니다.

- `/home`- 여기에는 기본 사용자 홈 폴더 (`/home/ec2_user` 아마존 리눅스, `/home/centos` 우분투) 가 포함됩니다/`/home/ubuntu`.
- `/opt/intel` - 여기에는 Intel MPI, Intel Parallel Studio 및 관련 파일이 포함됩니다.
- `/opt/slurm`- 여기에는 Slurm 워크로드 매니저 및 관련 파일이 포함됩니다. (조건부, Scheduler: `slurm`의 경우에만 해당.)

## NICE DCV 문제 해결

주제

- [NICE DCV에 대한 로그](#)
- [Ubuntu NICE DCV 문제](#)

## NICE DCV에 대한 로그

NICE DCV에 대한 로그는 `/var/log/dcv/` 디렉터리의 파일에 기록됩니다. 이러한 로그를 검토하면 문제를 해결하는 데 도움이 될 수 있습니다.

NICE DCV를 실행하려면 인스턴스 유형에 1.7기비바이트(GiB) 이상의 RAM이 있어야 합니다. 나노 및 마이크로 인스턴스 유형에는 NICE DCV를 실행하기에 충분한 메모리가 없습니다.

AWS ParallelCluster 로그 그룹에 NICE DCV 로그 스트림을 생성합니다. CloudWatch 콘솔 사용자 지정 대시보드 또는 로그 그룹에서 이러한 로그를 볼 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs와 통합](#) 및 [아마존 CloudWatch 대시보드](#) 섹션을 참조하세요.

## Ubuntu NICE DCV 문제

Ubuntu의 NICE DCV 세션을 통해 Gnome 터미널을 실행할 때 AWS ParallelCluster 가 로그인 셸을 통해 사용할 수 있게 하는 사용자 환경에 자동으로 액세스하지 못할 수 있습니다. 사용자 환경은 openmpi 또는 intelmpi 같은 환경 모듈과 기타 사용자 설정을 제공합니다.

Gnome 터미널의 기본 설정으로 인해 셸이 로그인 셸로 시작되지 않습니다. 즉, 셸 프로파일은 자동으로 소싱되지 않으며 AWS ParallelCluster 사용자 환경이 로드되지 않습니다.

셸 프로ファイルを 제대로 소싱하고 AWS ParallelCluster 사용자 환경에 액세스하려면 다음 중 하나를 수행하십시오.

- 기본 터미널 설정 변경
  1. Gnome 터미널에서 편집 메뉴를 선택합니다.
  2. 환경설정을 선택한 다음 프로파일을 선택합니다.
  3. 명령을 선택하고 로그인 셸로 명령 실행을 선택합니다.
  4. 새 터미널을 엽니다.
- 명령줄을 사용하여 사용 가능한 프로파일을 가져올 수 있습니다.

```
$ source /etc/profile && source $HOME/.bashrc
```

## AWS Batch 통합을 통해 클러스터의 문제 해결

이 섹션은 AWS Batch 스케줄러 통합이 있는 클러스터와 관련이 있습니다.

주제

- [헤드 노드 문제](#)
- [컴퓨팅 문제](#)
- [작업 실패](#)
- [엔드포인트 URL의 연결 시간 초과 오류](#)

## 헤드 노드 문제

Slurm 클러스터와 동일한 방식으로 헤드 노드 설정 문제를 해결할 수 있습니다(Slurm 전용 로그 제외). 이러한 문제에 대한 자세한 내용은 [헤드 노드](#) 섹션을 참조하세요.

## 컴퓨팅 문제

AWS Batch 서비스의 규모 조정 및 컴퓨팅 측면을 관리합니다. 컴퓨팅 관련 문제가 발생하는 경우 문제 AWS Batch [해결](#) 설명서에서 도움을 받으십시오.

## 작업 실패

작업이 실패할 경우 [awsbout](#) 명령을 실행하여 작업 출력을 검색할 수 있습니다. [awsbstat](#) 명령을 실행하여 Amazon에서 저장한 작업 로그로 연결되는 링크를 얻을 수도 CloudWatch 있습니다.

## 엔드포인트 URL의 연결 시간 초과 오류

다중 노드 병렬 작업이 Connect timeout on endpoint URL 오류로 실패하는 경우

- awsbout 출력 로그에서 작업이 Detected 3/3 compute nodes. Waiting for all compute nodes to start. 출력의 다중 노드 병렬인지 확인합니다.
- 컴퓨팅 노드 서브넷이 퍼블릭인지 확인합니다.

다중 노드 병렬 작업은 in을 사용할 때 퍼블릭 서브넷 사용을 지원하지 않습니다. AWS Batch AWS ParallelCluster 컴퓨팅 노드와 작업에는 프라이빗 서브넷을 사용하세요. 자세한 내용을 알아보려면 AWS Batch 사용 설명서의 [컴퓨팅 환경 고려 사항](#)을 참조하세요. 컴퓨팅 노드의 프라이빗 서브넷을 구성하려면 [AWS ParallelCluster 스케줄러 포함 AWS Batch](#)을 참조하세요.

## Active Directory와의 다중 사용자 통합 문제 해결

이 섹션은 Active Directory와 통합된 클러스터와 관련이 있습니다.

Active Directory 통합 기능이 예상대로 작동하지 않는 경우 SSSD 로그가 유용한 진단 정보를 제공할 수 있습니다. 이러한 로그는 클러스터 노드의 /var/log/sss에 있습니다. 기본적으로 클러스터의 Amazon CloudWatch 로그 그룹에도 저장됩니다.

### 주제

- [Active Directory 관련 문제 해결](#)
- [디버그 모드 활성화](#)

- [LDAPS에서 LDAP로 이동하는 방법](#)
- [LDAPS 서버 인증서 검증을 비활성화하는 방법](#)
- [암호 대신 SSH 키를 사용하여 로그인하는 방법](#)
- [사용자 암호 및 만료된 암호를 재설정하는 방법](#)
- [조인한 도메인을 확인하는 방법](#)
- [인증서 문제를 해결하는 방법](#)
- [Active Directory와의 통합이 제대로 작동하는지 확인하는 방법](#)
- [컴퓨팅 노드 로그인 문제를 해결하는 방법](#)
- [다중 사용자 환경에서 SimCenter StarCCM+ 작업의 알려진 문제](#)
- [사용자 이름 확인과 관련된 알려진 문제](#)
- [홈 디렉터리 생성 문제를 해결하는 방법](#)

## Active Directory 관련 문제 해결

이 섹션은 Active Directory 유형별 문제 해결과 관련이 있습니다.

### Simple AD

- DomainReadOnlyUser 값은 사용자에게 대한 Simple AD 디렉터리 기본 검색과 일치해야 합니다.

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

Users에 cn을 참고하세요.

- 기본 관리자 사용자는 Administrator입니다.
- Ldapsearch에는 사용자 이름 앞에 NetBIOS 이름이 있어야 합니다.

Ldapsearch 구문은 다음과 같아야 합니다.

```
$ ldapsearch -x -D "corp\\Administrator" -w "Password" -H ldap://192.0.2.103 \
  -b "cn=Users,dc=corp,dc=example,dc=com"
```

### AWS Managed Microsoft AD

- DomainReadOnlyUser 값은 사용자의 AWS Managed Microsoft AD 디렉터리 기반 검색과 일치해야 합니다.



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

- 기본 관리자 사용자는 Admin입니다.
- Ldapsearch 구문은 다음과 같아야 합니다.

```
$ ldapsearch -x -D "Admin" -w "Password" -H ldap://192.0.2.103 \
  -b "ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

## 디버그 모드 활성화

SSSD의 디버그 로그는 문제를 해결하는 데 유용할 수 있습니다. 디버그 모드를 활성화하려면 클러스터 구성을 다음과 같이 변경하여 클러스터를 업데이트해야 합니다.

```
DirectoryService:
  AdditionalSssdConfigs:
    debug_level: "0x1fff"
```

## LDAPS에서 LDAP로 이동하는 방법

LDAPS(TLS/SSL을 사용하는 LDAP)에서 LDAP로 전환하는 것은 권장되지 않습니다. LDAP만으로는 암호화가 제공되지 않기 때문입니다. 하지만 테스트 목적과 문제 해결에는 유용할 수 있습니다.

클러스터를 이전 구성 정의로 업데이트하여 클러스터를 이전 구성으로 복원할 수 있습니다.

LDAPS에서 LDAP로 이동하려면 클러스터 구성을 다음과 같이 변경하여 클러스터를 업데이트해야 합니다.

```
DirectoryService:
  LdapTlsReqCert: never
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

## LDAPS 서버 인증서 검증을 비활성화하는 방법

테스트 또는 문제 해결을 위해 헤드 노드에서 LDAPS 서버 인증서 검증을 일시적으로 비활성화하는 것이 유용할 수 있습니다.

클러스터를 이전 구성 정의로 업데이트하여 클러스터를 이전 구성으로 복원할 수 있습니다.

LDAPS 서버 인증서 검증을 사용하지 않도록 설정하려면 클러스터 구성에서 다음과 같이 변경하여 클러스터를 업데이트해야 합니다.

```
DirectoryService:
  LdapTlsReqCert: never
```

## 암호 대신 SSH 키를 사용하여 로그인하는 방법

SSH 키는 처음으로 암호를 사용하여 로그인한 후 `/home/$user/.ssh/id_rsa`에 생성됩니다. SSH 키로 로그인하려면 암호로 로그인하고 SSH 키를 로컬로 복사한 다음 평소와 같이 이를 사용하여 암호 없이 SSH 작업을 수행해야 합니다.

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY $username@$head_node_ip
```

## 사용자 암호 및 만료된 암호를 재설정하는 방법

사용자가 클러스터에 액세스할 수 없는 경우 [AWS Managed Microsoft AD 암호가 만료되었을 수 있습니다](#).

암호를 재설정하려면 디렉터리에 대한 쓰기 권한이 있는 사용자 및 역할과 함께 다음 명령을 실행합니다.

```
$ aws ds reset-user-password \
  --directory-id "d-abcdef01234567890" \
  --user-name "USER_NAME" \
  --new-password "NEW_PASSWORD" \
  --region "region-id"
```

[DirectoryService/DomainReadOnlyUser](#)의 비밀번호를 재설정된 경우:

1. [DirectoryService/PasswordSecretArn](#) 암호를 새 암호로 업데이트해야 합니다.
2. 새 암호 값에 맞게 클러스터를 업데이트하세요.
  - a. `pcluster update-compute-fleet` 명령을 사용하여 컴퓨팅 플릿을 중지합니다.
  - b. 클러스터 헤드 노드 내에서 다음 명령을 실행합니다.

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

암호를 재설정하고 클러스터를 업데이트한 후에는 사용자의 클러스터 액세스를 복원해야 합니다.

자세한 내용은 AWS Directory Service 관리 안내서의 [사용자 암호 재설정](#)을 참조하세요.

## 조인한 도메인을 확인하는 방법

다음 명령은 헤드 노드가 아닌 도메인에 가입된 인스턴스에서 실행해야 합니다.

```
$ realm list corp.example.com \
  type: kerberos \
  realm-name: CORP.EXAMPLE.COM \
  domain-name: corp.example.com \
  configured: kerberos-member \
  server-software: active-directory \
  client-software: sssd \
  required-package: oddjob \
  required-package: oddjob-mkhomedir \
  required-package: sssd \
  required-package: adcli \
  required-package: samba-common-tools \
  login-formats: %U \
  login-policy: allow-realm-logins
```

## 인증서 문제를 해결하는 방법

LDAPS 통신이 작동하지 않는 경우 TLS 통신 오류 때문일 수 있으며, 이는 인증서 문제 때문일 수 있습니다.

인증서에 대한 참고 사항:

- 클러스터 구성 `LdapTlsCaCert`에 지정된 인증서는 도메인 컨트롤러용 인증서를 발급한 전체 CA(인증 기관) 체인의 인증서를 포함하는 PEM 인증서 번들이어야 합니다.
- PEM 인증서 번들은 PEM 인증서를 연결하여 만든 파일입니다.
- PEM 형식의 인증서(일반적으로 Linux에서 사용됨)는 base64 DER 형식의 인증서(일반적으로 Windows에서 내보내는 인증서)와 동일합니다.
- 도메인 컨트롤러용 인증서를 하위 CA에서 발급한 경우 인증서 번들에는 하위 CA와 루트 CA의 인증서가 모두 포함되어야 합니다.

문제 해결 확인 단계:

다음 확인 단계에서는 명령이 클러스터 헤드 노드 내에서 실행되고 도메인 컨트롤러가 **SERVER:PORT**에 연결할 수 있다고 가정합니다.

인증서와 관련된 문제를 해결하려면 다음 확인 단계를 따르세요.

확인 단계:

1. Active Directory 도메인 컨트롤러에 대한 연결을 확인하세요.

도메인 컨트롤러에 연결할 수 있는지 확인합니다. 이 단계가 성공하면 도메인 컨트롤러에 대한 SSL 연결이 성공하고 인증서가 확인됩니다. 사용자의 문제는 인증서와 관련이 없습니다.

이 단계가 실패하면 다음 확인을 진행하세요.

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

2. 인증서 검증 확인:

로컬 CA 인증서 번들이 도메인 컨트롤러에서 제공한 인증서를 검증할 수 있는지 확인하세요. 이 단계가 성공하면 문제는 인증서와 관련된 것이 아니라 다른 네트워킹 문제와 관련이 있는 것입니다.

이 단계가 실패하면 다음 확인을 진행하세요.

```
$ openssl verify -verbose -  
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

3. Active Directory 도메인 컨트롤러에서 제공된 인증서를 확인하세요.

도메인 컨트롤러가 제공한 인증서의 내용이 예상한 것과 같은지 확인하세요. 이 단계가 성공하면 컨트롤러 확인에 사용된 CA 인증서에 문제가 있을 수 있습니다. 다음 문제 해결 단계로 이동하세요.

이 단계가 실패할 경우 도메인 컨트롤러용으로 발급된 인증서를 수정하고 문제 해결 단계를 다시 실행해야 합니다.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

4. 인증서 내용 확인:

도메인 컨트롤러가 제공한 인증서의 내용이 예상한 것과 같은지 확인하세요. 이 단계가 성공하면 컨트롤러 확인에 사용된 CA 인증서에 문제가 있을 수 있습니다. 다음 문제 해결 단계로 이동하세요.

이 단계가 실패할 경우 도메인 컨트롤러용으로 발급된 인증서를 수정하고 문제 해결 단계를 다시 실행해야 합니다.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

## 5. 로컬 CA 인증서 번들의 내용을 확인하세요.

도메인 컨트롤러 인증서를 검증하는 데 사용되는 로컬 CA 인증서 번들의 내용이 예상과 같은지 확인하세요. 이 단계가 성공하면 도메인 컨트롤러에서 제공하는 인증서에 문제가 있을 수 있습니다.

이 단계가 실패할 경우 도메인 컨트롤러용으로 발급된 CA 인증서 번들을 수정하고 문제 해결 단계를 다시 실행해야 합니다.

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

## Active Directory와의 통합이 제대로 작동하는지 확인하는 방법

다음 두 가지 검사가 성공하면 Active Directory와의 통합이 작동하는 것입니다.

확인:

1. 디렉터리에 정의된 사용자를 검색할 수 있습니다.

클러스터 헤드 노드 내에서 ec2-user로서:

```
$ getent passwd $ANY_AD_USER
```

2. 사용자 비밀번호를 제공하여 헤드 노드에 SSH로 연결할 수 있습니다.

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

검사 1이 실패하면 검사 2도 실패할 것으로 예상됩니다.

추가 문제 해결 확인:

- 사용자가 디렉터리에 있는지 확인하세요.
- [디버그 로깅](#) 활성화
- LDAPS 문제를 배제하려면 [LDAPS에서 LDAP로 이동하여](#) 암호화를 일시적으로 비활성화하는 것이 좋습니다.

## 컴퓨팅 노드 로그인 문제를 해결하는 방법

이 섹션은 Active Directory와 통합된 클러스터의 컴퓨팅 노드에 로그인하는 것과 관련이 있습니다.

를 사용하면 AWS ParallelCluster 클러스터 컴퓨팅 노드에 대한 암호 로그인이 설계상 비활성화됩니다.

모든 사용자는 자신의 SSH 키를 사용하여 컴퓨팅 노드에 로그인해야 합니다.

클러스터 구성에서 [GenerateSshKeysForUsers](#)가 활성화된 경우 사용자는 첫 번째 인증(예: 로그인) 후 헤드 노드에서 SSH 키를 검색할 수 있습니다.

사용자가 헤드 노드에서 처음으로 인증하는 경우 디렉터리 사용자를 위해 자동으로 생성되는 SSH 키를 검색할 수 있습니다. 사용자의 홈 디렉터리도 생성됩니다. sudo-user가 헤드 노드의 사용자로 처음 전환할 때도 이런 일이 발생할 수 있습니다.

사용자가 헤드 노드에 로그인하지 않은 경우 SSH 키는 생성되지 않으며 사용자는 컴퓨팅 노드에 로그인할 수 없습니다.

## 다중 사용자 환경에서 SimCenter StarCCM+ 작업의 알려진 문제

이 섹션은 Siemens의 Simcenter StarCCM+ 전산 유체 역학 소프트웨어가 다중 사용자 환경에서 시작한 작업과 관련이 있습니다.

내장된 IntelMPI를 사용하도록 구성된 StarCCM+ v16 작업을 실행하는 경우 기본적으로 MPI 프로세스는 SSH를 사용하여 부트스트랩됩니다.

사용자 이름 확인이 잘못되는 알려진 [Slurm버그](#)로 인해 다음과 같은 오류가 발생하여 작업이 실패할 수 있습니다. error setting up the bootstrap proxies 이 버그는 AWS ParallelCluster 버전 3.1.1 및 3.1.2에만 영향을 미칩니다.

이러한 문제를 방지하려면 IntelMPI를 MPI 부트스트랩 방법으로 강제로 사용하십시오Slurm. [IntelMPI 공식 설명서](#)에 설명된 대로 StarCCM+를 시작하는 작업 스크립트로 환경 변수 I\_MPI\_HYDRA\_BOOTSTRAP=slurm를 내보냅니다.

## 사용자 이름 확인과 관련된 알려진 문제

이 섹션은 작업 내에서 사용자 이름을 검색하는 것과 관련이 있습니다.

[Slurm의 알려진 버그](#)로 인해 `srun` 없이 작업을 실행하면 작업 프로세스 내에서 검색된 사용자 이름은 `nobody`일 수 있습니다. 이 버그는 버전 3.1.1 및 3.1.2에만 영향을 미칩니다 AWS ParallelCluster .

예를 들어 디렉터리 사용자로 명령 `sbatch --wrap 'srun id'`을 실행하면 올바른 사용자 이름이 반환됩니다. 하지만 디렉터리 사용자로 `sbatch --wrap 'id'`를 실행하면 사용자 이름으로 `nobody`가 반환될 수 있습니다.

다음 해결 방법을 사용할 수 있습니다.

1. 가능하면 'srun' 대신 'sbatch'를 사용하여 작업을 시작하세요.
2. 클러스터 내 구성을 다음과 같이 [AdditionalSssdConfigs](#) 설정하여 SSSD 열거를 활성화합니다.

```
AdditionalSssdConfigs:
  enumerate: true
```

## 홈 디렉터리 생성 문제를 해결하는 방법

이 섹션은 홈 디렉터리 생성 문제와 관련이 있습니다.

다음 예와 같은 오류가 표시되면 헤드 노드에 처음 로그인했을 때 홈 디렉터리가 자동으로 생성되지 않은 것입니다. 또는 `sudoer`에서 헤드 노드의 Active Directory 사용자로 처음 전환했을 때 홈 디렉터리가 자동으로 생성되지 않은 경우도 있습니다.

```
$ ssh AD_USER@$HEAD_NODE_IP
/opt/parallelcluster/scripts/generate_ssh_key.sh failed: exit code 1

  _|  _|_ )
  _| (    /  Amazon Linux 2 AMI
  _|\_|_|_|

https://aws.amazon.com/amazon-linux-2/
Could not chdir to home directory /home/PclusterUser85: No such file or directory
```

홈 디렉터리 생성 실패는 클러스터 헤드 노드에 설치된 `odddjob` 및 `odddjob-mkhomedir` 패키지로 인해 발생할 수 있습니다.

홈 디렉터리와 SSH 키가 없으면 사용자는 클러스터 노드에 작업이나 SSH를 제출할 수 없습니다.

시스템에 oddjob 패키지가 필요한 경우 oddjobd 서비스가 실행 중인지 확인하고 PAM 구성 파일을 새로 고쳐 홈 디렉터리가 생성되었는지 확인하세요. 이렇게 하려면 다음 예와 같이 헤드 노드에서 명령을 실행하세요.

```
sudo systemctl start oddjobd
sudo authconfig --enablemkhomedir --updateall
```

시스템에 oddjob 패키지가 필요하지 않은 경우 패키지를 제거하고 PAM 구성 파일을 새로 고쳐 홈 디렉터리가 생성되었는지 확인하세요. 이렇게 하려면 다음 예와 같이 헤드 노드에서 명령을 실행하세요.

```
sudo yum remove -y oddjob oddjob-mkhomedir
sudo authconfig --enablemkhomedir --updateall
```

## 사용자 지정 AMI 문제 해결

사용자 지정 AMI를 사용할 때 다음 경고가 표시됩니다.

```
"validationMessages": [
  {
    "level": "WARNING",
    "type": "CustomAmiTagValidator",
    "message": "The custom AMI may not have been created by pcluster. You can ignore
this warning if the AMI is shared or copied from another pcluster AMI. If the
AMI is indeed not created by pcluster, cluster creation will fail. If the cluster
creation fails, please go to https://docs.aws.amazon.com/parallelcluster/latest/ug/
troubleshooting.html#troubleshooting-stack-creation-failures for troubleshooting."
  },
  {
    "level": "WARNING",
    "type": "AmiOsCompatibleValidator",
    "message": "Could not check node AMI ami-0000012345 OS and cluster OS alinux2
compatibility, please make sure they are compatible before cluster creation and update
operations."
  }
]
```

올바른 AMI가 사용되고 있다고 확신하는 경우 이러한 경고를 무시해도 됩니다.



앞으로 이러한 경고를 보지 않으려면 사용자 지정 AMI에 다음 태그를 지정하세요. 여기서 *my-os*은 *alinux2*, *ubuntu2204*, *ubuntu2004*, *centos7* 또는 *rhe18* 중 하나이며 *"3.7.0"*은 사용 중인 *pcluster* 버전입니다.

```
$ aws ec2 create-tags \
  --resources ami-yourcustomAmi \
  --tags Key="parallelcluster:version",Value="3.7.0"
Key="parallelcluster:os",Value="my-os"
```

## cf-n-hup이 실행 중이 아닐 때의 클러스터 업데이트 제한 시간 문제 해결

*cf-n-hup* 헬퍼는 리소스 메타데이터의 변경 사항을 감지하고 변경 사항이 감지되면 사용자 지정 작업을 실행하는 데몬(daemon)입니다. 이것이 UpdateStack API 작업을 통해 실행 중인 Amazon EC2 인스턴스에 대한 구성 업데이트를 수행하는 방법입니다.

현재 *cf-n-hup* 데몬은 *supervisord*에 의해 실행됩니다. 하지만 실행 후에는 *cf-n-hup* 프로세스가 *supervisord* 제어에서 분리됩니다. 외부 행위자가 *cf-n-hup* 데몬을 종료할 경우 데몬은 자동으로 다시 시작되지 않습니다. 가 *cf-n-hup* 실행되지 않는 경우 클러스터 업데이트 중에 CloudFormation 스택은 예상대로 업데이트 프로세스를 시작하지만 헤드 노드에서 업데이트 절차가 활성화되지 않아 결국 스택이 타임아웃됩니다. 클러스터 로그 */var/log/chef-client*에서 업데이트 레시피가 호출되지 않는 것을 확인할 수 있습니다.

장애 발생 시 **cf-n-hup**를 확인하고 다시 시작하세요.

1. 헤드 노드에서 *cf-n-hup*가 실행 중인지 확인합니다.

```
$ ps aux | grep cf-n-hup
```

2. 헤드 노드에서 *cf-n-hup* 로그 */var/log/cf-n-hup.log* 및 */var/log/supervisord.log*을 확인하세요.
3. *cf-n-hup*가 실행 중이 아니면 다음을 실행하여 다시 시작해 보세요.

```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl
start cf-n-hup
```

## 네트워크 문제 해결

### 단일 퍼블릭 서브넷 안의 클러스터 문제

컴퓨팅 노드 중 하나에서 `cloud-init-output.log`를 확인하세요. 노드가 Slurm 초기화 중에 중단되었음을 나타내는 다음과 같은 내용이 발견되면 DynamoDB VPC 엔드포인트가 누락되었기 때문일 가능성이 큼니다. DynamoDB 엔드포인트를 추가합니다. 자세한 내용은 [AWS ParallelCluster 인터넷에 액세스할 수 없는 단일 서브넷에서](#) 항목을 참조하세요.

```
ruby_block[retrieve compute node info] action run[2022-03-11T17:47:11+00:00] INFO:
  Processing ruby_block[retrieve compute node info] action run (aws-parallelcluster-
  slurm::init line 31)
```

### onNodeUpdated 사용자 지정 작업에서 클러스터 업데이트가 실패한 경우

[HeadNode/CustomActions/OnNodeUpdated](#) 스크립트가 실패하면 업데이트가 실패하고 롤백 시 스크립트가 실행되지 않습니다. 롤백이 완료된 후 필요한 정리를 수동으로 수행하는 것은 사용자의 책임입니다. 예를 들어 OnNodeUpdated 스크립트가 구성 파일의 필드 상태(예: true에서 false로)를 변경한 후 실패한 경우 해당 필드 값을 업데이트 전 상태(예: false에서 true로)로 수동으로 복원해야 합니다. 자세한 내용은 [사용자 지정 부트스트랩 작업](#) 항목을 참조하세요.

### 사용자 지정 Slurm 구성에서 오류가 표시되는 경우

AWS ParallelCluster 버전 3.6.0부터는 사용자 지정 구성에 단일 prolog 또는 epilog 스크립트를 포함하여 더 이상 대상으로 지정할 수 없습니다. Slurm AWS ParallelCluster 버전 3.6.0 이상 버전에서는 각 및 폴더에서 사용자 지정 prolog 및 epilog 스크립트를 찾아야 합니다. Prolog Epilog 이러한 폴더는 기본적으로 다음을 가리키도록 구성되어 있습니다.

- Prolog는 `/opt/slurm/etc/scripts/prolog.d/`를 가리킵니다.
- Epilog는 `/opt/slurm/etc/scripts/epilog.d/`를 가리킵니다.

`90_plcluster_health_check_manager` 프로로그 스크립트와 `90_pcluster_noop` 에필로그 스크립트는 제자리에 두는 것이 좋습니다.

Slurm는 스크립트를 알파벳 역순으로 실행합니다. Prolog 및 Epilog 폴더 모두에 적어도 한 개 이상의 파일이 포함되어 있어야 합니다. 자세한 내용은 [Slurmprolog 및 epilog](#) 및 [Slurm 구성 사용자 지정](#) 섹션을 참조하세요.

## 클러스터 알람

최적의 성능을 보장하려면 클러스터 상태 모니터링이 필수적입니다. AWS ParallelCluster 클러스터 헤드 노드에 대한 여러 CloudWatch 기반 경보를 모니터링할 수 있습니다.

이 섹션에서는 명명 규칙, 경보를 트리거하는 특정 조건, 제안된 문제 해결 단계를 포함하여 각 유형의 헤드 노드 클러스터 경보에 대한 세부 정보를 제공합니다.

클러스터 경보의 명명 규칙은 예를 들어 다음과 같습니다. CLUSTER\_NAME-COMPONENT-METRIC  
mycluster-HeadNode-Cpu

- CLUSTER\_NAME-HeadNode: 헤드 노드의 전체 상태를 나타냅니다. 아래 경보 중 하나 이상이 다음과 같으면 빨간색입니다.
- CLUSTER\_NAME-HeadNode-Health: Amazon EC2 상태 확인 실패가 하나 이상 있는 경우 빨간색입니다. 경보가 발생하는 경우 상태 확인에 [실패한 인스턴스 문제 해결](#)을 살펴보는 것이 좋습니다.
- CLUSTER\_NAME-HeadNode-Cpu: CPU 사용률이 90% 이상이면 빨간색입니다. 알람이 발생하는 경우 CPU를 가장 많이 소비하는 프로세스를 확인하세요. `ps -aux --sort=-%cpu | head -n 10`.
- CLUSTER\_NAME-HeadNode-Mem: 메모리 사용률이 90% 이상이면 빨간색입니다. 알람이 발생하는 경우 메모리를 가장 많이 소비하는 프로세스를 확인하세요. `ps -aux --sort=-%mem | head -n 10`.
- CLUSTER\_NAME-HeadNode-Disk: path /에서 디스크 점유 공간이 90% 를 초과하면 빨간색입니다. 알람이 발생하는 경우, 대부분의 공간을 차지하는 폴더를 확인하세요. `du -h --max-depth=2 / 2> /dev/null | sort -hr`

## 추가 지원

알려진 문제 목록은 기본 [GitHub Wiki](#) 페이지 또는 [문제](#) 페이지를 참조하십시오.

더 긴급한 문제의 경우 AWS Support 문의하거나 [새 GitHub 문제를](#) 여십시오.

# AWS ParallelCluster 지원 정책

AWS ParallelCluster 여러 릴리스를 동시에 지원합니다. 모든 AWS ParallelCluster 릴리스에는 예정된 지원 기간 종료 (EOSL) 날짜가 있습니다. EOSL 날짜 이후에는 해당 릴리스에 대한 추가 지원이나 유지 관리가 제공되지 않습니다.

AWS ParallelCluster `major.minor.patch` 버전 체계를 사용합니다. 최신 메이저 버전 릴리스의 새 마이너 버전 릴리스에는 새로운 기능, 성능 개선, 보안 업데이트 및 버그 수정이 포함됩니다. 마이너 버전은 메이저 버전 내에서 이전 버전과 호환됩니다. 심각한 문제의 경우 AWS는 패치 릴리스를 통해 수정 사항을 제공하지만 EOSL에 도달하지 않은 최신 마이너 버전의 릴리스에만 적용됩니다. 새 버전 릴리스의 업데이트를 사용하려면 새 마이너 버전 또는 패치 버전으로 업그레이드해야 합니다.

| AWS ParallelCluster 버전 | 지원 기간 종료(EOSL) 날짜 |
|------------------------|-------------------|
| 3.0. <i>x</i>          | 2023년 3월 31일      |
| 3.1. <i>x</i>          | 8/31/2023년        |
| 3.2. <i>x</i>          | 1/31/2024         |
| 3.3. <i>x</i>          | 5/31/2024         |
| 3.4. <i>x</i>          | 6/28/2024         |
| 3.5. <i>x</i>          | 8/31/2024         |
| 3.6. <i>x</i>          | 11/30/2024        |
| 3.7. <i>x</i>          | 2/28/2025         |
| 3.8. <i>x</i>          | 2025 6/30/30      |
| 3.9. <i>x</i>          | 09/05/2025        |
| 3.10. <i>x</i>         | 12/27/2025        |

# 보안 내부 AWS ParallelCluster

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

보안은 기업과 기업 간의 AWS 공동 책임입니다. [공동 책임 모델](#)에서는 이를 클라우드 자체의 보안과 클라우드 내부의 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 적용되는 규정 준수 프로그램에 대해 자세히 알아보려면 규정 준수 [프로그램별 범위 내 AWS 서비스 규정 준수](#) 참조하십시오. AWS ParallelCluster
- 클라우드에서의 보안 — 사용하는 특정 AWS 서비스나 서비스에 따라 책임이 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 여러 관련된 요소에 대해서도 책임이 있습니다.

이 문서에서는 공동 책임 모델을 사용할 때 공동 책임 모델을 적용하는 방법을 설명합니다 AWS ParallelCluster. 다음 항목에서는 보안 및 규정 준수 목표를 AWS ParallelCluster 충족하도록 구성하는 방법을 보여줍니다. 또한 AWS 리소스를 모니터링하고 보호하는 데 도움이 되는 AWS ParallelCluster 방식으로 사용하는 방법도 알아봅니다.

## 주제

- [에서 사용하는 서비스의 보안 정보 AWS ParallelCluster](#)
- [데이터 보호 기능 AWS ParallelCluster](#)
- [AWS ParallelCluster ID 및 액세스 관리](#)
- [AWS ParallelCluster의 규정 준수 확인](#)
- [TLS 1.2의 최소 버전 적용](#)

## 에서 사용하는 서비스의 보안 정보 AWS ParallelCluster

- [Amazon EC2의 보안](#)
- [Amazon API Gateway의 보안](#)
- [내부 보안 AWS Batch](#)
- [AWS CloudFormation의 보안](#)

- [아마존의 보안 CloudWatch](#)
- [AWS CodeBuild의 보안](#)
- [Amazon DynamoDB의 보안](#)
- [Amazon ECR의 보안](#)
- [Amazon ECS의 보안](#)
- [Amazon EFS의 보안](#)
- [FSx for Lustre의 보안](#)
- [보안 입력 AWS Identity and Access Management \(IAM\)](#)
- [EC2 Image Builder의 보안](#)
- [보안: 내부 AWS Lambda](#)
- [Amazon Route 53의 보안](#)
- [Amazon SNS의 보안](#)
- [Amazon SQS의 보안 \( AWS ParallelCluster 버전 2.x용\)](#)
- [Amazon S3의 보안](#)
- [Amazon VPC의 보안](#)

## 데이터 보호 기능 AWS ParallelCluster

AWS [공동 책임 모델](#) 의 데이터 보호에 적용됩니다 AWS ParallelCluster. 이 모델에 설명된 대로 AWS 는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 사용자는 인프라 에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시를 참조하십시오](#) FAQ. 유럽의 데이터 보호에 대한 자세한 내용은 [AWS 공동 책임 모델 및 AWS 보안 GDPR 블로그](#) 의 블로그 게시물을 참조하십시오.

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

- 각 계정마다 다단계 인증 (MFA) 을 사용하십시오.
- SSL/TLS/를 사용하여 AWS 리소스와 통신하세요. TLS1.2가 필요하고 TLS 1.3을 권장합니다.
- API를 사용하여 사용자 활동 로깅을 설정합니다 AWS CloudTrail.

- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API an을 AWS 통해 액세스할 때 FIPS 140-3개의 검증된 암호화 모듈이 필요한 경우 엔드포인트를 사용하십시오. FIPS 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [연방 정보 처리](#) 표준 ( ) 140-3을 참조하십시오. FIPS

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

## 데이터 암호화

보안 서비스의 주요 특징은 정보가 활발히 사용되지 않을 때 암호화된다는 것입니다.

### 저장 중 암호화

AWS ParallelCluster 는 사용자를 대신하여 AWS 서비스와 상호 작용하는 데 필요한 자격 증명 이외의 고객 데이터를 자체적으로 저장하지 않습니다.

클러스터의 노드에 있는 데이터의 경우 저장된 데이터가 암호화될 수 있습니다.

Amazon EBS 볼륨의 경우 [EbsSettings](#) 섹션

의 [EbsSettings/Encrypted](#) 및 [EbsSettings/KmsKeyId](#) 설정을 사용하여 암호화가 구성됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EBS 암호화](#)를 참조하십시오.

Amazon EFS 볼륨의 경우 [EfsSettings](#) 섹션

의 [EfsSettings/Encrypted](#) 및 [EfsSettings/KmsKeyId](#) 설정을 사용하여 암호화가 구성됩니다. 자세한 내용은 Amazon Elastic File System 사용 설명서의 [저장 중 암호화 작동 방식](#)을 참조하세요.

Lustre 파일 시스템의 경우 Amazon FSx 파일 시스템을 생성할 때 저장된 데이터의 암호화가 자동으로 활성화됩니다. FSx 자세한 내용은 Amazon FSx for Lustre 사용 [설명서의 저장 데이터 암호화](#)를 참조하십시오.

볼륨이 있는 인스턴스 유형의 경우 NVMe 인스턴스 스토어 NVMe 볼륨의 데이터는 인스턴스의 하드웨어 모듈에 구현된 XTS - AES -256 암호를 사용하여 암호화됩니다. 암호화 키는 하드웨어 모듈을 사용

하여 생성되며 각 NVMe 인스턴스 스토리지 디바이스마다 고유합니다. 인스턴스가 중지되거나 종료되면 모든 암호화 키가 손상되어 복구가 불가능해집니다. 이 암호화를 비활성화할 수 없으며, 사용자 자신의 암호화 키를 제공할 수 없습니다. 자세한 내용은 Amazon EC2 사용 설명서의 저장 중 [암호화](#)를 참조하십시오.

고객 데이터를 로컬 AWS ParallelCluster 컴퓨터로 전송하여 저장하는 AWS 서비스를 호출하는 경우 해당 데이터의 저장, 보호 및 암호화 방법에 대한 자세한 내용은 해당 서비스 사용 설명서의 보안 및 규정 준수 장을 참조하십시오.

## 전송 중 암호화

기본적으로 실행 중인 클라이언트 AWS ParallelCluster 컴퓨터와 AWS 서비스 엔드포인트에서 전송되는 모든 데이터는/연결을 통해 모든 데이터를 전송하여 암호화됩니다. HTTPS TLS 클러스터의 노드 간 트래픽은 선택한 인스턴스 유형에 따라 자동으로 암호화될 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [전송 중 암호화](#)를 참조하십시오.

다음 사항도 참조하십시오.

- [아마존에서의 데이터 보호 EC2](#)
- [EC2Image Builder에서의 데이터 보호](#)
- [의 데이터 보호 AWS CloudFormation](#)
- [아마존에서의 데이터 보호 EFS](#)
- [Amazon S3의 데이터 보호](#)
- [Lustre를 FSx 위한 데이터 보호](#)

## AWS ParallelCluster ID 및 액세스 관리

AWS ParallelCluster에서는 역할을 사용하여 AWS 리소스와 해당 서비스에 액세스합니다. AWS ParallelCluster가 권한을 부여하는 데 사용하는 인스턴스 및 사용자 정책은 [AWS Identity and Access Management 권한 입력 AWS ParallelCluster](#)에 설명되어 있습니다.

유일한 주요 차이점은 표준 사용자와 장기 보안 인증을 사용할 때 인증하는 방법입니다. 사용자가 AWS 서비스 콘솔에 액세스하려면 암호가 필요하지만 동일한 사용자가 AWS ParallelCluster를 사용하여 동일한 작업을 수행하려면 액세스 키 페어가 필요합니다. 다른 모든 단기 보안 인증은 콘솔에서 사용되는 것과 동일한 방식으로 사용됩니다.

AWS ParallelCluster에서 사용하는 보안 인증은 일반 텍스트 파일에 저장되며 암호화되지 않습니다.



- `$HOME/.aws/credentials` 파일에는 AWS 리소스에 액세스하는 데 필요한 장기 보안 인증이 저장됩니다. 여기에는 계정 액세스 키 ID와 비밀 액세스 키가 포함됩니다.
- 사용자가 수임하는 역할이나 AWS IAM Identity Center 서비스에 대한 보안 인증과 같은 단기 보안 인증도 각각 `$HOME/.aws/cli/cache` 및 `$HOME/.aws/sso/cache` 폴더에 저장됩니다.

### 위험 완화

- `$HOME/.aws` 폴더와 해당 하위 폴더 및 파일에 대한 파일 시스템 권한을 구성하여 권한 있는 사용자만 액세스할 수 있도록 제한하는 것이 좋습니다.
- 보안 인증이 손상된 경우, 손상 가능성을 줄이려면 가능한 한 임시 보안 인증이 있는 역할을 사용합니다. 단기 역할 보안 인증을 요청하고 새로 고치는 경우에만 장기 보안 인증을 사용합니다.

## AWS ParallelCluster의 규정 준수 확인

서드 파티 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 AWS 서비스의 보안 및 규정 준수를 평가합니다. AWS ParallelCluster를 사용하여 서비스에 액세스하는 경우에도 해당 서비스의 규정 준수는 변경되지 않습니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 [규정 준수 프로그램 제공 범위 내 AWS 서비스](#)를 참조하세요. 일반적인 내용은 [AWS 규정 준수 프로그램](#)을 참조하세요.

AWS Artifact를 사용하여 서드 파티 감사 보고서를 다운로드할 수 있습니다. 자세한 정보는 [AWS Artifact에서 보고서 다운로드](#)를 참조하세요.

AWS ParallelCluster 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- [보안 및 규정 준수 빠른 시작 안내서](#) - 이 배포 안내서에서는 아키텍처 고려 사항에 관해 설명하고 AWS에서 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- [Architecting for HIPAA security and compliance on Amazon Web Services AWS 백서](#) - 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 생성하는 방법을 설명합니다.
- [AWS 규정 준수 리소스](#) - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 모음입니다.
- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) - AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- [AWS Security Hub](#) - 이 AWS 서비스는 보안 산업 표준 및 모범 사례 규정 준수 여부를 확인하는 데 도움이 되도록 AWS 내 보안 상태를 종합적으로 보여줍니다.

## TLS 1.2의 최소 버전 적용

AWS 서비스와 통신할 때 보안을 강화하려면 TLS 1.2 이상을 사용하도록 AWS ParallelCluster를 구성해야 합니다. AWS ParallelCluster를 사용하면 TLS 버전을 설정하는 데 Python이 사용됩니다.

AWS ParallelCluster에서 TLS 1.2 이전의 TLS 버전을 사용하지 않도록 하려면 OpenSSL을 다시 컴파일하여 이 최소값을 적용하고 새로 빌드된 OpenSSL을 사용하도록 Python을 다시 컴파일해야 할 수 있습니다.

### 현재 지원되는 프로토콜 확인

먼저 OpenSSL을 사용하여 테스트 서버 및 Python SDK에 사용할 자체 서명된 인증서를 만듭니다.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

그런 다음 OpenSSL을 사용하여 테스트 서버를 가동합니다.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

새 터미널 창에서 가상 환경을 만들고 Python SDK를 설치합니다.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

SDK의 기본 HTTP 라이브러리를 사용하는 check.py라는 새로운 Python 스크립트를 만듭니다.

```
$ 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'))
```

새 스크립트를 실행합니다.

```
$ python check.py
```

그러면 연결에 대한 세부 정보가 표시됩니다. 출력에서 "프로토콜 : "을 검색합니다. 출력이 "TLSv1.2" 이상이면 SDK는 기본적으로 TLS v1.2 이상으로 설정됩니다. 이전 버전인 경우 OpenSSL을 다시 컴파일하고 Python을 다시 컴파일해야 합니다.

그러나 Python이 기본적으로 TLS v1.2 이상으로 설치되더라도 서버가 TLS v1.2 이상을 지원하지 않으면 Python이 TLS v1.2 이전 버전으로 다시 협상할 수 있습니다. Python이 이전 버전으로 자동으로 다시 협상하지 않는지 확인하려면 다음과 같이 테스트 서버를 다시 시작하세요.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

이전 버전의 OpenSSL을 사용하는 경우 `-no_tls1_3` 플래그를 사용할 수 없을 수 있습니다. 이 경우 사용 중인 OpenSSL 버전이 TLS v1.3을 지원하지 않으므로 플래그를 제거합니다. 그런 다음 Python 스크립트를 다시 실행합니다.

```
$ python check.py
```

Python 설치가 TLS 1.2 이전 버전에서 올바르게 다시 협상되지 않으면 SSL 오류가 발생합니다.

```
$ 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)')))
```

연결할 수 있는 경우 TLS v1.2 이전의 프로토콜 협상을 비활성화하기 위해 OpenSSL과 Python을 다시 컴파일해야 합니다.

## OpenSSL 및 Python 컴파일

AWS ParallelCluster에서 TLS 1.2 이전 버전을 협상하지 않도록 하려면 OpenSSL과 Python을 다시 컴파일해야 합니다. 이렇게 하려면 다음 내용을 복사하여 스크립트를 만들고 실행합니다.

```
#!/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

```

이것은 TLS 1.2 이전 버전을 자동으로 협상하지 않는 정적으로 연결된 OpenSSL을 가진 Python 버전을 컴파일합니다. 또한 /opt/openssl-with-min-tls1\_2 디렉터리에 OpenSSL을 설치하고 /opt/python-with-min-tls1\_2 디렉터리에 Python을 설치합니다. 이 스크립트를 실행한 후 새 버전의 Python 설치를 확인하세요.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

다음 사항이 인쇄되어야 합니다.

```
Python 3.8.1
```

이 새 버전의 Python이 TLS 1.2 이전 버전을 협상하지 않는지 확인하려면 새로 설치된 Python 버전(즉, [현재 지원되는 프로토콜 확인](#))을 사용하는 /opt/python-with-min-tls1\_2/bin/python3의 단계를 다시 실행합니다.

## 릴리스 정보 및 문서 기록

다음 표에서 AWS ParallelCluster 사용 설명서의 중요한 업데이트 및 새 기능이 나와 있습니다. 사용자로부터 받은 의견을 수렴하기 위해 설명서가 자주 업데이트됩니다.

| 변경 사항                                                         | 설명                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 날짜           |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <a href="#">AWS ParallelCluster UI 버전 2024.07.1이 출시되었습니다.</a> | <p>UI 버전 2024.07.1의 AWS ParallelCluster 출시를 발표하게 되어 기쁩니다.</p> <p>변경 사항:</p> <ul style="list-style-type: none"> <li>3.10.1에 대한 지원을 추가합니다. AWS ParallelCluster</li> </ul> <p>버그 수정:</p> <ul style="list-style-type: none"> <li>작업 계정 정보 렌더링이 중단되는 버그를 수정했습니다.</li> <li>PC 3.10.0 이상에서 모든 PC 3.2.0+ 기능을 비활성화하는 기능 플래그 지정 메커니즘의 버그를 수정했습니다.</li> </ul> <p>보안:</p> <ul style="list-style-type: none"> <li>암호화를 42.0.2에서 42.0.4로 업그레이드하고 취약점을 해결하기 위한 디코딩 알고리즘을 지정하십시오. <a href="https://cwe.mitre.org/data/definitions/476.html">https://cwe.mitre.org/data/definitions/476.html</a><a href="https://cwe.mitre.org/data/definitions/327.html">https://cwe.mitre.org/data/definitions/327.html</a></li> </ul> | 2024년 7월 24일 |

- 취약성을 해결하려면 Node.js 를 16.20.2에서 18.20.3으로, Next.js 를 13.5.1에서 14.1.1로 업그레이드하십시오. <https://cwe.mitre.org/data/definitions/918.html>
- 취약성을 해결하려면 ws 버전을 8.17.1 이상으로 제한하십시오. <https://cwe.mitre.org/data/definitions/476.html>

[전체 변경 로그를 참조하십시오.](#)

[AWS ParallelCluster 버전 3.10.1이 출시되었습니다.](#)

3.10.1의 AWS ParallelCluster 릴리스를 발표하게 되어 기쁩니다.

2024년 7월 8일

버그 수정:

- 중국 지역의 이미지 빌드 실패를 수정합니다.

[AWS ParallelCluster UI 버전 2024.07.0이 출시되었습니다](#)

UI 버전 2024.07.0의 AWS ParallelCluster 출시를 발표하게 되어 기쁩니다.

2024년 7월 2일

기능:

- 버전 3.10.0에 대한 지원이 추가되었습니다. AWS ParallelCluster

[AWS ParallelCluster 버전 3.10.0이 출시되었습니다.](#)

3.10.0의 릴리스를 발표하게 되어 기쁩니다. AWS ParallelCluster

업그레이드하려면 `aws-parallelcluster`를 입력하세요.  
`sudo pip install --upgrade aws-parallelcluster`

개선 사항:

- 클러스터를 외부 Slurmdbd에 Scheduling/SlurmSettings/ExternalSlurmdbd 연결하는 새 구성 섹션을 추가합니다.
- 격리된 네트워크에서 빌드 이미지를 실행할 수 있도록 허용하세요.
- 아마존 리눅스 2023에 대한 지원을 추가합니다.
- `price-capacity-optimized`에 대한 지원을 추가합니다.`AllocationStrategy`.
- 용량 블록이 있는 배치 그룹을 사용하지 못하도록 유효성 검사기를 추가하십시오.

변경 사항:

- CentOS 7은 더 이상 지원되지 않습니다.
- Cinc Client를 18.2.7에서 18.4.12 버전으로 업그레이드하십시오.

- 먼지를 버전 0.5.16으로 업그레이드하십시오 (0.5.15에서).
- Pmix를 4.2.9에서 5.0.2로 업그레이드하세요.
- 서드 파티 쿼백 종속성 업그레이드:
  - apt-7.5.22 (apt-7.5.14부터)
  - openssh-2.11.12 (openssh-2.11.3부터)
- 타사 쿼백 삭제: 셀리누스-6.1.12.
- EFA인스톨러를 로 업그레이드하십시오 1.32.0.
  - Efa-driver: efa-2.8.0-1
  - EFA 구성: efa-configuration-1.16-1
  - EFA 프로필: efa-profile-1.7-1
  - LibFabric-AWS: libfabric-aws-1.21.0-1
  - RDMA 코어: rdma-core-50.0-1
  - 오픈: 및 MPI openmpi40-aws-4.1.6-3  
openmpi50-aws-5.0.2-12
- NVIDIA드라이버를 버전 535.183.01 (535.154.05에서) 으로 업그레이드하십시오.



- Python을 3.9.19로 업그레이드합니다 (3.9.17에서).
- 인텔 MPI 라이브러리를 2021.12.1.8로 업그레이드하십시오 (2021.9.0.43482에서).

#### 버그 수정:

- 데이터 리포지토리 연결 구성을 선택 사항으로 만들도록 수정하십시오.  
AutoExportPolicy  
AutoImportPolicy
- 클러스터 삭제 중 인스턴스가 종료 또는 종료 상태일 때 컴퓨팅 플릿 정리가 완료되는 문제를 수정했습니다. 이는 종료 주기가 더 긴 인스턴스 유형의 클러스터 삭제 실패를 방지하기 위한 것입니다.
- 클러스터 구성 Monitoring 섹션에서 cloudwatch 대시보드를 활성화하고 경보를 비활성화할 수 있습니다.
- ParallelCluster 사용자 지정 리소스가 유효성 검사기의 사용을 차단하도록 허용합니다. PclusterCluster/ SuppressValidators
- 사용자가 로그인할 때마다 실행되지 않고 환경 변수에 cfn\_bootstrap\_virtualenv 추가되지 /

etc/profile.d/pcluster.sh 않도록 제거되었습니다. PATH

- failuresDescribeCluster 응답에서 필드를 failureReason 로 대체하여 ParallelCluster API 사양을 수정하세요.
- 누락된 CloudFormation 스택 상태 (IMPORT\_\*REVIEW\_IN\_PROGRESS , 및) 를 추가하여 ParallelCluster API 사양을 수정합니다. UPDATE\_FAILED
- 전송 중에 암호화가 적용된 EFS 파일 시스템이 클러스터 업데이트에 포함되지 않던 문제를 수정합니다.
- 공유 내부 데이터에 사용되는 경우 헤드 노드 재부팅 시 slurmctld 및 slurmdbd 서비스가 다시 시작되지 않던 문제를 수정했습니다. EFS
- Ubuntu 시스템에서는 Parallelcluster에서 가져온 구성과 충돌하는 cloud-init 로그 파일에 대한 기본 logrotate 구성을 제거하십시오.
- 8.10 RHEL 이상에서 이미지 빌드 실패를 수정합니다.

[1.0.0용 테라폼 프로바이더 출시 AWS ParallelCluster](#)

1.0.0용 Terraform Provider의 출시를 발표하게 되어 기쁩니다. AWS ParallelCluster

2024년 6월 26일

기능:

- [전체 변경 로그](#)

[1.0.0용 테라폼 모듈 출시 AWS ParallelCluster](#)

1.0.0용 테라폼 모듈의 출시를 발표하게 되어 기쁩니다. AWS ParallelCluster

2024년 6월 26일

기능:

- [전체 변경 로그](#)

## [AWS ParallelCluster 버전](#)

### [3.9.3 출시](#)

3.9.3의 릴리스를 발표하게 되어 기쁩니다. AWS ParallelCluster

2024년 6월 19일

업그레이드하려면 다음을 입력하십시오. `sudo pip install --upgrade aws-parallelcluster`

기능:

- 에 공유 스토리지 FSx Lustre 유형으로 대한 지원이 추가되었습니다 `us-iso-east-1` .

버그 수정:

- Slurm `Slurmctl` Parameters 팬아웃 문제를 방지하려면 Slurm `cloud_dns` 구성에서 제거하십시오.

인스턴스 시작 시 IP 주소를 설정하므로 필수 사항은 아닙니다.

[AWS ParallelCluster 버전 3.9.2가 출시되었습니다](#)

3.9.2의 릴리스를 발표하게 되어 기쁩니다. AWS ParallelCluster

2024년 5월 28일

기능:

- Slurm23.11.7로 업그레이드 하십시오 (23.11.4에서).
- 자세한 내용은 on을 참조 하십시오. [CHANGELOG 3.9.2](#) [GitHub](#)

[AWS ParallelCluster UI 버전 2024.05.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2024.05.0이 출시되었습니다.

2024년 5월 14일

버그 수정:

- 사용자가 Job Status 패널을 열 때 UI를 차단하는 프론트 엔드의 버그를 수정했습니다.
- [전체 변경 로그](#)

[AWS ParallelCluster UI 버전 2024.04.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2024.04.0이 출시되었습니다.

2024년 4월 17일

기능:

- 버전 3.9.1에 대한 지원이 추가되었습니다. AWS ParallelCluster
- [전체 변경 로그](#)

## [AWS ParallelCluster 버전 3.9.1 출시](#)

3.9.1의 릴리스를 발표하게 되어 기쁩니다. AWS ParallelCluster

2024년 4월 11일

업그레이드하려면 다음을 입력합니다. `sudo pip install --upgrade aws-parallelcluster`

버그 수정

- 업데이트 클러스터 작업의 일환으로 파일 시스템을 마운트 해제할 때 공유 스토리지 mountdir의 재귀적 삭제를 제거하십시오.

## [AWS ParallelCluster 버전 3.9.1이 출시되었습니다.](#)

3.9.1의 릴리스를 발표하게 되어 기쁩니다. AWS ParallelCluster

2024년 4월 11일

업그레이드하려면 다음을 입력합니다. `sudo pip install --upgrade aws-parallelcluster`

버그 수정

- 업데이트 클러스터 작업의 일환으로 파일 시스템을 마운트 해제할 때 공유 스토리지 mountdir의 재귀적 삭제를 제거하십시오.

[AWS ParallelCluster UI 버전 2024.03.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2024.03.0이 출시되었습니다.

2024년 3월 12일

기능:

- 버전 3.9.0에 대한 지원이 추가되었습니다. AWS ParallelCluster
- 우분투 22.04와 레드햇 엔터프라이즈 리눅스 9에 대한 지원이 추가되었습니다.
- 더 이상 사용되지 않는 우분투 18.04

버그 수정

- 클러스터를 여러 개 사용할 때 일부 클러스터가 나타나지 않는 문제가 해결되었습니다.

변경 사항에 대한 자세한 내용은 의 [aws-parallelcluster-ui](#) 패키지 CHANGELOG 파일을 참조하십시오 GitHub.

## [AWS ParallelCluster 버전 3.9.0이 출시되었습니다.](#)

2024년 3월 5일

3.9.0의 릴리스를 발표하게 되어 기쁩니다. AWS ParallelCluster

업그레이드하려면 다음을 입력합니다. `sudo pip install --upgrade aws-parallelcluster`

개선 사항:

- 사용자가 기본 사용자의 홈 디렉터리를 `/home (default)` `/local/home` 대신 이동할 수 있도록 구성 매개 변수를 `DeploymentSettings/DefaultUserHome` 추가합니다.
- 컴퓨팅 플릿을 중지할 필요 없이 `MinCountMaxCount`, `Queue` 및 `ComputeResource` 구성 파라미터를 업데이트할 수 있습니다. 이제 `Scheduling/SlurmSettings/QueueUpdateStrategy` 설정하여 업데이트할 수 `TERMINATE` 있습니다. AWS ParallelCluster 클러스터 업데이트를 통해 수행된 클러스터 용량 크기 조정 중에 제거된 노드만 종료합니다.
- 컴퓨팅 `FsxOpenZfs` 및 로그인 플릿을 `FileCache` 교체하지 않고 `Ef`, `FsxLustre` `FsxOntap`, 유형의 외부 공유



스토리지를 업데이트할 수 있습니다.

- 에 대한 지원을 추가합니다 RHEL9.
- build-image 프로세스를 통해 CustomAmi 생성된 Rocky Linux 9에 대한 지원을 추가합니다. 현재 공식 AWS ParallelCluster Rocky9 AMI Linux는 제공되지 않습니다.
- 사용자 지정 Slurm 설정 거부 CommunicationParameters 목록에서 제거합니다.
- supportd에서 기본 사용자의 sudo 액세스를 비활성화하는 DeploymentSettings/DisableSudoAccessForDefaultUser 매개변수를 추가합니다. OSes
- Lustre 파일 시스템 생성자 변경 ParallelCluster: Lustre 서버 버전을 다음으로 변경 FSx 2.15.
- 콕북 노드 속성을 통해 빌드 시 오픈 소스 Nvidia 드라이버와 클로즈드 소스 Nvidia 드라이버 중에서 선택할 수 AMI 있는 가능성을 추가합니다. ['cluster']['nvidia']['kernel\_open']
- \* clustermgtd 구성 옵션을 추가하여 최종 Amazon

Description ec2\_instance\_missing\_max\_count 인스턴스와 실행 인스턴스의 일관성에 대해 구성 가능한 재시도 횟수를 허용하십시오. EC2

## 변경

- 23.02.7에서 23.11.4로 업그레이드하십시오. Slurm
- 드라이버를 버전 535.154.05로 업그레이드합니다 NVIDIA.
- CLIpcluster에 Python 3.11, 3.12에 대한 지원을 추가하고. aws-parallelcluster-batch-cli
- 범위를 MaximumNetworkCards 반복하는 대신 Amazon EC2 DescribeInstances 응답 NetworkCardIndex 목록의 네트워크 카드 인덱스를 사용하여 네트워크 인터페이스를 구축하십시오.
- 인스턴스 유형 P3, G3, P2, G2를 사용하면 해당 GPU 아키텍처가 3.8.0 릴리스의 일부로 도입된 오픈 소스 Nvidia 드라이버 (OpenRM)와 호환되지 않으므로 클러스터 생성에 실패합니다.
- 타사 쿼백 종속성 업그레이드: nfs-5.1.2 (nfs-5.0.0에서)

- EFA설치 프로그램을 다음으로 업그레이드하십시오.  
1.30.0.
  - Efa-driver: efa-2.6.0-1
  - EFA 구성: efa-config-1.15-1
  - EFA 프로파일: efa-profile-1.6-1
  - LibFabric-AWS: libfabric-aws-1.19.0
  - RDMA 코어: rdma-core-46.0-1
  - 오픈: 및 MPI openmpi40-aws-4.1.6-2  
openmpi50-aws-5.0.0-11
- NICE DCV 버전으로 업그레이드 2023.1-16388.
  - server: 2023.1.16388-1
  - xdcv: 2023.1.565-1
  - gl: 2023.1.1047-1
  - web\_viewer: 2023.1.16388-1

### 버그 수정

- 로그인 노드에서 Active Directory 사용자로 제출했을 때 작업이 실패하는 문제를 수정했습니다. 이 문제는 헤드 노드의 외부 Active

Directory와의 통합 구성이  
불완전하여 발생했습니다.

- CloudFormation parallelcluster-policies.yaml 템플릿에 정의된 IAM 정책을 리팩터링하여 정책이 제한을 초과하여 발생하는 배포 실패를 방지합니다. ParallelCluster API IAM
- 헤드 노드가 키를 쓰는 데 예상보다 많은 시간이 걸리는 경우 로그인 노드가 부트스트랩에 실패하는 문제를 수정했습니다.

변경 사항에 대한 자세한 내용은 의 [aws-parallelcluster-ui](#) 패키지 CHANGELOG 파일을 참조하십시오. GitHub

[AWS ParallelCluster UI 버전 2024.02.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2024.02.0이 출시되었습니다.

2024년 2월 8일

변경 사항:

- Lambda 런타임 환경을 Python v3.9로 업데이트했습니다.

변경 사항에 대한 자세한 내용은 의 패키지 CHANGELOG 파일을 참조하십시오. [aws-parallelcluster-ui](#) GitHub

[AWS ParallelCluster UI 버전 2023.12.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2023.12.0이 출시되었습니다.

2023년 12월 21일

기능:

- 사설 네트워킹을 통한 PCUI 배포 지원이 추가되었습니다.
- PCUI 및 PCAPI 인프라에서 생성한 모든 IAM 역할에 권한 경계를 선택적으로 적용할 수 있는 기능이 추가되었습니다.
- 및 인프라에서 생성한 모든 IAM 역할과 정책에 접두사를 선택적으로 적용할 수 있는 PCUI 기능이 추가되었습니다. PCAPI
- 마법사의 기능 패리티 없이 ParallelCluster 버전 3.8.0에 대한 지원이 추가되었습니다.

변경 사항에 대한 자세한 내용은 의 패키지 CHANGELOG 파일을 참조하십시오. [aws-parallelcluster-ui](#) GitHub

[AWS ParallelCluster 버전 3.8.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.8.0이 출시되었습니다.

2023년 12월 19일

개선 사항:

- ML용 Amazon EC2 용량 블록에 대한 지원을 추가합니다.
- build-image 프로세스를 통해 CustomAmi 생성된 Rocky Linux 8에 대한 지원을 추가합니다. 현재 공식 AWS ParallelCluster Rocky8 AMI Linux는 제공되지 않습니다.
- Slurm컴퓨팅 노드용 Amazon EC2 인스턴스를 시작할 때 사용할 클러스터 조정 전략을 제어하는 Scheduling/Scaling Strategy 파라미터를 추가합니다. 가능한 값은all-or-nothing ,, greedy-all-or-nothing best-effort , all-or-nothing 이며 기본값은 다음과 같습니다.
- 클러스터 내 공유 파일 시스템 리소스 (Intel ParallelCluster, Slurm 및 data)에 대해 헤드 노드 루트 볼륨에서 NFS 내보내는 대신 EFS 스토리지를 사용하도록 HeadNode/SharedSto

rageType 매개 변수를 추가합니다. /home 이 개선 사항은 헤드 노드 네트워킹의 부하를 줄여줍니다.

- 구성 파일의 SharedStorage 섹션을 통해 FSx 외부 공유 스토리지 EFS 또는 외부 공유 /home 스토리지로 마운트할 수 있습니다.
- AWS Secrets Manager의 외부 사용자 정의 MUNGE 키를 사용할 수 있도록 새 매개 변수를 SlurmSettings/MungeKeySecretArn 추가합니다.
- 클러스터의 Amazon CloudWatch Alarms를 전환하기 위한 Monitoring/Alarms/Enabled 파라미터를 추가합니다.
- 헤드 노드 경보를 추가하여 Amazon EC2 상태 점검, CPU 사용률 및 헤드 노드의 전체 상태를 모니터링하고 클러스터로 만든 CloudWatch 대시보드에 추가합니다.
- 관리형 PERSISTENT\_2 DeploymentType FSx Lustre용으로 사용할 때 데이터 리포지토리 연결에 대한 지원을 추가하세요.
- 사용자가 데이터베이스 서버에서 계정에 사용할 데이터베이스의 사용자 지정 이름을 지정할 수 있도록

Scheduling/SlurmSettings/Database/DatabaseName 매개 변수를 추가합니다. Slurm

- 컴퓨팅 CapacityReservationTarget/CapacityReservationId 리소스에서 구성할 때 InstanceType 선택적 구성 매개 변수를 만드십시오.
- 에서 생성한 IAM AWS ParallelCluster API 역할 및 정책의 접두사를 지정할 수 있는 가능성을 추가합니다.
- 에서 생성한 IAM AWS ParallelCluster API 역할 및 정책에 적용할 권한 경계를 지정할 수 있는 가능성을 추가합니다.

## 변경

- 23.02.6에서 Slurm 23.02.7로 업그레이드하십시오.
- 드라이버를 버전 535.129.03으로 업그레이드합니다.  
NVIDIA
- CUDA툴킷을 버전 12.2.2로 업그레이드합니다.
- 폐쇄형 소스 모듈 대신 오픈 소스 NVIDIA GPU 드라이버 (OpenRM) 를 Linux용 NVIDIA 커널 모듈로 사용하십시오. NVIDIA



- 새 Scheduling/Scaling Strategy 클러스터 `all_or_nothing_batch` 구성을 위해 Slurm 재개 프로그램에서 구성 매개 변수 지원을 제거하세요.
- 클러스터 경보 명명 규칙을 '[클러스터-이름] - [구성 요소 이름] - [지표]'로 변경했습니다.
- 루트 볼륨과 추가 EBS 볼륨 모두에 대해 ADC 지역의 기본 볼륨 유형을 gp2에서 gp3으로 변경합니다.
- 에 대한 선택적 권한 AWS ParallelCluster API 경계는 이제 인프라에서 생성되는 모든 IAM 역할에 적용됩니다. API
  - EFA설치 프로그램을 로 1.29.1 업그레이드하십시오.
  - Efa-driver: efa-2.6.0-1
  - EFA 구성: efa-configuration-1.15-1
  - EFA 프로필: efa-profile-1.5-1
  - LibFabric-AWS: libfabric-aws-1.19.0-1
  - RDMA 코어: rdma-core-46.0-1

- 오픈MPI: openmpi40-aws-4.1.6-1
- 버전 2.3.10이 사용되는 Centos 7을 제외하고 지원되는 OSes 모든 버전에서 버전 2.4로 GDRCopy 업그레이드 하십시오.
- 버전 aws-cfn-bootstrap 2.0-28로 업그레이드 하세요.
- 파이썬 3.10인치에 대한 지원을 추가합니다. aws-parallelcluster-batch-cli

## 버그 수정

- 컴퓨팅 리소스에 선언된 인스턴스 유형 목록을 수정할 때 클러스터 업데이트 롤백 후 스케일링 구성이 일관되지 않는 문제를 수정합니다.
- 클러스터 구성 파일을 통해 외부 LDAP 서버와 통합된 클러스터에서 루트 권한이 없는 사용자를 전환할 때 발생하는 사용자 SSH 키 생성을 수정합니다.
- 설정 시 Slurm 절전 모드가 비활성화되는 문제를 수정했습니다. Scaledown Idletime = -1
- Accounting 스크립트에서 하드 코딩된 Slurm 설치 디렉토리 경로를 수정합니다. update\_slurm\_datab

ase\_password.sh

Slurm

[AWS ParallelCluster 버전  
3.7.2 출시](#)

AWS ParallelCluster 버전  
3.7.2가 출시되었습니다.

2023년 10월 25일

변경 사항:

- Slurm23.02.6으로 업그레이드하세요.

[AWS ParallelCluster UI 버전 2023.10.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2023.10.0이 출시되었습니다.

2023년 10월 20일

기능:

- ParallelCluster 3.7.2에 대한 지원이 추가되었으며 마법사의 기능 패리티는 FSx 파일 캐시 및 여러 인스턴스 유형과의 메모리 기반 스케줄링 호환성으로 제한되었습니다.

버그 수정:

- Cost Explorer와 상호 작용할 PCUI 권한이 없을 때 UI 오류가 발생하는 문제를 수정했습니다.

개선 사항

- 액세스 토큰을 TTL 10분에서 5분으로 줄여 보안을 개선했습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster-ui](#) 패키지의 CHANGELOG 파일을 참조하십시오 GitHub.

[AWS ParallelCluster 버전 3.7.1이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.7.1이 출시되었습니다.

2023년 9월 22일

변경 사항:

- Slurm23.02.4에서 23.02.5로 업그레이드하십시오.
  - Pmix를 4.2.6으로 업그레이드하세요 (3.2.3에서).
  - libjet를 1.12.0에서 1.15.3으로 업그레이드하세요.
- EFA설치 프로그램을 로 업그레이드하여 P5의 쓰기 데이터 문제를 수정합니다.
  - 1.26.1 RDMA
    - efa-2.5.0-1 EFA 드라이버:
    - EFA 구성: efa-config-1.15-1
    - EFA 프로필: efa-profile-1.5-1 .
    - LibFabric-AWS: libfabric-aws-1.18.2-1
    - ERdma-코어: rdma-core-46.0-1
    - 오픈MPI: openmpi40-aws-4.1.5-4

## [AWS ParallelCluster 버전](#) [3.7.0 출시](#)

AWS ParallelCluster 버전  
3.7.0이 출시되었습니다.

2023년 8월 30일

개선 사항:

- Support는 구성 YAML 파일을 사용하여 컴퓨팅 리소스의 정적 및 동적 노드 우선 순위 AWS ParallelCluster 구성을 지원합니다.
- Ubuntu 22에 대한 지원 추가  
RSA키는 기본적으로 지원되지 않습니다.
- 대기열 구성 설정  
JobExclusiveAllocation 을 추가하여 언제든지 파티션의 노드를 단일 작업에만 독점적으로 할당할 수 있습니다.
- 클러스터 생성 및 클러스터 업데이트 시 오버라이드  
aws-parallelcluster-node 패키지 허용 헤드 노드의 경우 클러스터 업데이트에도 적용됩니다. 개발 목적으로만 유용합니다.
- 컴퓨팅 노드에서 NFS 서버를 시작하지 마십시오.
- 로그인 노드에 대한 지원을 추가합니다.
- 컴퓨팅 리소스에 여러 인스턴스 유형이 지정된 경우 메모리 기반 스케줄링을 허용하십시오. Slurm

- 기존 Amazon File Cache를 공유 스토리지로 탑재하기 위한 지원을 추가합니다.

#### 변경 사항:

- 기본적으로 Slurm 동적 노드에 우선 순위(가중치) 1000을 할당합니다. 이렇게 하면 Slurm은 유휴 동적 노드보다 유휴 정적 노드를 높은 우선 순위로 지정할 수 있습니다.
- `aws-parallelcluster-node` 데몬이 관리형 파티션만 처리하도록 만드세요. AWS ParallelCluster Slurm
- EFS-utils 위치독 폴링 간격을 10초로 늘입니다. 이 변경 사항은 위치독이 실행 되도록 하는 유일한 조건인 `EncryptionInTransit`가 `true`로 설정된 경우에 적용됩니다.
- EFA인스톨러를 로 업그레이드하세요. 1.25.1
  - Efa-driver: efa-2.1.1g 에서 efa-2.5.0-1 로
  - Efa-config: efa-config-1.13-1 에서 efa-config-1.15-1 로
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws: libfabric-aws-1.17

- .1-1 에서 libfabric  
-aws-1.18.1-0 로
- Rdma-core: rdma-core  
-43.0-1 에서 rdma-core-46.0-1 로
- 열기MPI: openmpi40-aws-4.1.5-4 (from openmpi40-aws-4.1.5-1 )
- Slurm가 버전 23.02.4로 업그레이드되었습니다.
- Imds/의 디폴트 값을 ImdsSupport v1.0에서 v2.0으로 변경합니다.
- Ubuntu 18을 더 이상 사용하지 마세요.
- Centos 7의 제한을 고려하여 기본 루트 볼륨 크기를 40GB로 업데이트
- 루트 노드만 읽을 수 있도록 헤드 노드 내의 /tmp/wait\_condition\_handle.txt 파일에 대한 권한을 제한
- 노드 패키지 데몬이 PC에서 관리하는 Slurm 파티션과 노드 목록을 인식하는 데 사용할 파티션-노드 목록 매핑 JSON 파일을 만드십시오.  
Slurm
- NVIDIA드라이버를 버전 535.54.03으로 업그레이드하십시오.



- CUDA라이브러리를 버전 12.2.0으로 업그레이드합니다.
- 패브릭 매니저를 NVIDIA 엔비디아-패브릭매니저-535로 업그레이드하세요.
- 우분투 22.04의 경우에만 PL을 버전 23.04.1로 업그레이드하십시오ARM.
- NICE DCV 2023.0-15487 버전으로 업그레이드.
  - 서버: 2023.0.15487-1
  - xdcv: 2023.0.551-1
  - gl: 2023.0.1039-1
  - web\_viewer: 2023.0.15487-1

#### 버그 수정:

- 값이 -1보다 작게 설정되지 않도록 Scaledown Idletime 값에 유효성 검사를 추가합니다.
- 활성화된 AMI GPU 인스턴스에서 Ubuntu Deep Learning으로 인한 클러스터 생성 실패를 수정합니다. DCV
- 를 사용하여 ParallelCluster CloudFormation 사용자 지정 리소스 공급자를 생성할 때 탱글링 IAM 정책이 생성되는 문제를 수정했습니다. CustomLambdaRole

- 여러 네트워크 인터페이스가 있는 인스턴스에서 equals를 사용할 SlurmSettings/Dns/UseEc2Hostnames 때 컴퓨팅 노드 DNS 이름이 잘못 정렬되는 문제를 수정했습니다. True

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

### 설명서 전용 릴리스

AWS ParallelCluster 버전 3별 사용 설명서가 게시되었습니다.

2023년 7월 17일

### 설명서 전용 릴리스

- AWS ParallelCluster 버전 3에는 별도의 사용 설명서가 있습니다.

[AWS ParallelCluster 버전 3.6.1이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.6.1이 출시되었습니다.

2023년 7월 5일

변경 사항:

- 컴퓨팅 노드가 여러 Slurm 파티션에 추가되는 경우 `clustermgtd` 에 나타나는 노드 중복 방지

버그 수정:

- 루트 볼륨 장치 이름 (`/dev/sda1` 및 `/dev/xvda` ) 의 하드 코딩을 제거하고 사용 중에 사용된 장치에서 검색 하십시오AMI. `create-cluster`
- 로 ElasticIp 설정된 CloudFormation 사용자 지정 리소스를 사용할 때 발생하는 클러스터 생성 실패 문제를 True 수정했습니다.
- 대규모 구성 파일이 포함된 AWS CloudFormation 사용자 지정 리소스를 사용할 때 발생하는 클러스터 생성 및 업데이트 실패를 수정합니다.
- Ubuntu에서 `ptrace` 보호 기능이 비활성화되지 않고 `libfabric` 내에서 Cross Memory Attach (CMA) 를 허

용하지 않았던 문제를 수정했습니다.

- 여러 인스턴스 유형을 사용하고 인스턴스가 반환되지 않는 경우 용량이 빠르게 부족한 장애 조치 로직을 수정했습니다.

변경 사항에 대한 자세한 내용은 CHANGELOG [aws-parallelcluster](#)의 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster UI 버전 2023.06.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2023.06.0이 출시되었습니다.

2023년 6월 7일

변경 사항:

- 기본 AWS ParallelCluster API 버전을 3.6.0으로 업그레이드했습니다.

버그 수정:

- AWS GovCloud (미국 서부) 지역의 배포가 중단되는 문제를 수정했습니다.
- 이제 생성 시작 후 분할 패널이 클러스터 세부 정보를 올바르게 로드합니다.

참고:

- 에서는 비용 모니터링 기능을 사용할 수 없습니다. AWS GovCloud (US) Regions

변경 사항에 대한 자세한 내용은 에서 [aws-parallelcluster-ui](#) 패키지 CHANGELOG 파일을 참조하십시오 GitHub.

[AWS ParallelCluster 버전 3.6.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.6.0이 출시되었습니다.

2023년 5월 22일

설명서:

- [AWS ParallelCluster Python 라이브러리 API](#) 설명서가 추가되었습니다.

개선 사항:

- [에 대한 지원을 추가합니다.](#)  
RHEL8
- [를 사용하여 클러스터를 만들고 관리하기 위한 AWS CloudFormation 사용자 지정 리소스를 추가합니다](#)  
CloudFormation.
- 구성 YAML 파일에 [클러스터 Slurm 구성 사용자 지정](#)을 위한 지원을 추가합니다. AWS ParallelCluster
- [를 Slurm LUA 지원하여 빌드 하십시오.](#)
- 클러스터당 대기열의 최대 개수를 10개에서 50개로 늘립니다. 각 대기열에는 최대 50개의 컴퓨팅 리소스가 포함될 수 있습니다. 각 클러스터에는 최대 50개의 컴퓨팅 리소스가 포함될 수 있습니다.
- `OnNodeStart` , `OnNodeConfigured` , 및

OnNodeUpdated 파라미터로 구성된 이벤트에 대해 여러 [사용자 지정 작업 스크립트](#)의 시퀀스를 지정하는 지원 추가

- 작업을 실행하기 전에 컴퓨팅 노드에 GPU 상태 점검을 적용하기 위한 새 구성 섹션 HealthChecks Gpu /을 추가합니다.
- SlurmQueues 및 SlurmQueues /ComputeResources 구성에 대한 Tags 지원을 추가합니다.
- Monitoring 구성에 [DetailedMonitoring](#) 지원 추가
- AWS ParallelCluster [CloudWatch 대시보드에](#) 헤드 노드 메모리 및 루트 볼륨 디스크 사용률 추적에 대한 disk\_used\_percent 메트릭을 추가하고 mem\_used\_percent 이러한 메트릭을 모니터링하기 위한 경보를 설정합니다.
- AWS ParallelCluster 관리형 로그에 대한 [로그 순환](#) 지원을 추가합니다.
- [대시보드에서 일반적인 컴퓨팅 노드 오류와 동적 노드 최장 유휴 시간을 추적할 수 있습니다.](#) [CloudWatch](#)

- 소켓을 생성할 때 DCV 인증 서버가 최소한 TLS-1.2 프로토콜을 사용하도록 강제하십시오. SSL
- 및 를 제외한 지원되는 모든 운영 체제에 [NVIDIA 데이터 센터 GPU 관리자 \(DCGM\)](#) 패키지를 설치합니다. aarch64 centos7 alinux2
- 기본적으로 커널 모듈 [nvidia-vm](#)을 로드하여 통합 가상 메모리 (UVM) 기능을 드라이버에 제공하십시오. CUDA
- [NVIDIA퍼시스턴스 데몬을 시스템 서비스로 설치](#)합니다.

#### 변경 사항:

- Slurm을 버전 22.05.8에서 버전 23.02.2으로 업그레이드
- munge를 버전 0.5.15에서 버전 0.5.14으로 업그레이드
- Slurm TreeWidth 를 30으로 설정
- Slurm prolog 및 epilog 구성을 각각 대상 디렉토리 /opt/slurm/etc/scripts/prolog.d/ 및 /opt/slurm/etc/scr



ipts/epilog.d/ 에 설정합니다.

- 컴퓨팅 노드 등록 중에 Prolog 스크립트를 실행하기 위한 Slurm BatchStartTimeout 를 최대 3분으로 설정
- CloudWatch 로그의 RetentionInDays 기본값을 14일에서 180일로 늘리십시오.
- EFA설치 프로그램을 로 업 그레이드하십시오. 1.22.1
  - Dkms: 2.8.3-2
  - Efa-driver: efa-2.1.1g (변경 없음)
  - Efa-config: efa-config-1.13-1 (변경 없음)
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws: libfabric-aws-1.17.0-1 에서 libfabric-aws-1.17.1-1 로
  - Rdma-core: rdma-core-43.0-1 (변경 없음)
  - 열기MPI: openmpi40-aws-4.1.5-1 (변경 없음)
- Amazon Linux 2에서 Lustre 클라이언트 버전을 2.12로 업그레이드 Ubuntu 20.04, 18.04, CentOS >= 7.7에

Lustre 클라이언트 2.12가 설치되었습니다.

- CentOS 7.6에서 Lustre 클라이언트 버전을 2.10.8로 업그레이드
- NVIDIA드라이버를 버전 470.182.03 (버전에서 470.141.03 ) 으로 업그레이드합니다.
- NVIDIA패브릭 관리자를 버전 470.182.03 (버전에서 470.141.03 ) 으로 업그레이드합니다.
- NVIDIA CUDA 툴킷을 버전 11.8.0 (버전에서 11.7.1) 으로 업그레이드합니다.
- NVIDIA CUDA 샘플을 버전으로 11.8.0 업그레이드하십시오.
- 인텔 MPI 라이브러리를 버전 2021 업데이트 9로 업그레이드합니다 (버전 2021 업데이트 6에서). 자세한 내용은 [인텔® MPI 라이브러리 2021 업데이트 9](#)를 참조하십시오.
- NICE DCV 버전으로 업그레이드 2023.0-15022 (버전에서 2022.2-14521 ).
  - server: 버전 2022.2-14521-1 에서 2023.0.15022-1 로
  - xdcv: 버전 2022.2.519-1 에서 2023.0.547-1 로

- `gl`: 버전 2022.2.10  
12-1 에서 2023.0.10  
27-1 로
- `web_viewer`: 버전  
2022.2.14521-1 에서  
2023.0.15022-1 로
- `aws-cfn-bootstrap` 을  
버전 2.0-24로 업그레이드
- AWS Batch 클러스터용 컨  
테이너 이미지를 빌드할 때  
CodeBuild 환경에서 사용하  
는 이미지 업그레이드:
  - `aws/codebuild/amazonlinux2-x86_64-standard:3.0` 에서  
`aws/codebuild/amazonlinux2-x86_64-standard:4.0` 로
  - `aws/codebuild/amazonlinux2-aarch64-standard:1.0` 에서  
`aws/codebuild/amazonlinux2-aarch64-standard:2.0` 로

#### 버그 수정:

- 잘못된 오류가 보고되지  
않도록 Amazon EFS 및  
Amazon FSx 네트워크 보안  
그룹 검사기를 수정하십시  
오.
- `build-image` 작업 중에  
Image Builder에서 생성한 리

소스의 태깅이 누락되는 문제를 수정합니다.

- MaxCount 속성에 대해 수치 비교를 항상 수행하도록 MaxCount 에 대한 업데이트 정책 수정
- 여러 네트워크 카드가 있는 컴퓨팅 노드 인스턴스의 IP 정렬을 수정했습니다.
- 대기열 파라미터 업데이트가 수행되고 Slurm 회계 구성이 업데이트되지 않은 경우의 `slurm_parallelcluster_slurmdbd.conf` 에서 `StoragePass` 대체 문제 수정
- 기존 EFS 파일 시스템으로 클러스터를 생성할 때 매달린 보안 그룹이 생성되는 문제를 수정했습니다.
- `cfn-hup` 대몬(daemon)을 다시 시작할 때 실패하는 문제를 수정했습니다.
- `INVALID_REG` 플래그가 있는 동적 노드를 Slurm 보호 모드의 부트스트랩 실패로 간주 Slurm 등록에 실패한 정적 노드는 `node_replacement_timeout` 등록 이후에 이미 부트스트랩 실패로 취급됩니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지

를 참조하십시오. [aws-parallelcluster-cookbook](#)  
[aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster UI 버전 2023.05.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2023.05.0이 출시되었습니다.

2023년 5월 16일

#### 개선 사항:

- AWS ParallelCluster 버전 3.6.0부터 8에 대한 지원을 추가합니다. RHEL
- 클러스터 비용 모니터링 추가
- AWS ParallelCluster 버전 3.6.0부터 대기열 및 컴퓨팅 리소스 할당량을 늘리십시오.

#### 변경 사항:

- 클러스터 생성 마법사 사용자 인터페이스가 개선되었습니다.
- UI 배포 속도가 향상되었습니다. AWS ParallelCluster
- 새 사용자 추가를 위한 인터페이스가 개선되었습니다.
- 대기열은 기본적으로 헤드 노드 서브넷에 있습니다.

#### 버그 수정:

- 클러스터 생성이 완료된 후 올바른 지역으로 전환
- “클러스터 편집” 특성에서 로딩 표시기 표시 수정

- EBS SnapshotId 속성이 제거될 때 클러스터 생성을 수정합니다.

변경 사항에 대한 자세한 내용은 의 [aws-parallelcluster-ui](#) 패키지 CHANGELOG 파일을 참조하십시오 GitHub.

[AWS ParallelCluster UI 버전 2023.04.0이 출시되었습니다.](#)

AWS ParallelCluster UI 버전 2023.04.0이 출시되었습니다.

2023년 4월 17일

개선 사항:

- 클러스터 생성 마법사 재설계
- 클러스터 로그 페이지 재설계
- 공유 스토리지에 사용자 지정 이름 설정 추가
- 클러스터에 스토리지를 추가할 때 여러 스토리지 선택 추가
- EFSAmazon과 FSx Lustre에 대한 DeletionPolicy 지원을 추가합니다.
- 클러스터 구성에 ImdsSupport 설정 추가
- C7 인스턴스 유형에 대한 지원 추가
- [이전 AWS Systems Manager 문서 버전으로 되돌리기](#) 튜토리얼 추가

변경 사항:

- 클러스터 구성 크기는 YAML 최대 1MB입니다.
- Boto3 IAM 임시 자격 증명을 통한 권한 부여로 인해 사용자가 로그아웃하지 않았습니다.



- 인스턴스 선택 시 멀티스레드 옵션이 비활성화되었습니다. HPC
- 클러스터 생성 페이지에서 비활성화 롤백 제거
- 필요한 정보가 제공될 때까지 사용자는 AWS ParallelCluster UI를 사용할 수 없습니다.
- 대기열을 최대 10개까지 추가할 수 있습니다.
- AWS ParallelCluster UI 설치 중에 SSM-SessionManagerRunShell 문서가 덮어쓰이지 않습니다.

#### 버그 수정:

- 비밀번호 재설정의 깨진 링크 수정
- EcrPrivateRepository 가 비어 있지 않음으로 인한 delete stack 깨짐 문제 수정
- 다중 사용자 관리 속성 섹션의 SSH 키 생성 확인란의 초기화 문제가 해결되었습니다.
- 속성이 정의되지 않은 작업으로 인해 발생하는 충돌이 수정되었습니다.
- 설정이 수정되었습니다 SCRATCHFSx.

- 한 번 클릭해도 여전히 활성화되어 있는 인스턴스 시작 및 중지 버튼 수정

변경 사항에 대한 자세한 내용은 에서 [aws-parallelcluster-ui](#) 패키지 CHANGELOG 파일을 참조하십시오 GitHub.

[AWS ParallelCluster 버전 3.5.1이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.5.1이 출시되었습니다.

2023년 3월 29일

개선 사항:

- [독립형 설치 프로그램 실행 파일 pcluster CLI 추가](#)

변경 사항:

- 설치 프로그램을 로 업그레이드합니다. EFA 1.22.0
  - Efa-driver: efa-2.1.1-1 에서 efa-2.1.1g 로
  - Efa-config: efa-config-1.12-1에서 efa-config-1.13-1 로
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws: libfabric-aws-1.16.1amzn3.0-1 에서 libfabric-aws-1.17.0-1 로
  - Rdma-core: rdma-core-43.0-1 (변경 없음)
  - 열기MPI: openmpi40-aws-4.1.5-1 (from openmpi40-aws-4.1.4-3 )

NICEDCV버전으로 업그레이드2022.2-14521 .

- server: 2022.2.14  
521-1
- xdcv: 2022.2.519-1
- gl: 2022.2.1012-1
- web\_viewer: 2022.2.14  
521-1

#### 버그 수정:

- 클러스터 업데이트의 일환으로 공유 Amazon EBS 볼륨을 제거할 `/etc/exports` 때 패턴 매칭으로 인해 발생할 수 있는 노드 시작 실패를 수정합니다. `MountDir`
- `clustermgtd` 반복마다 `compute_console_output` 로그 파일이 잘리지 않도록 수정했습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 [aws-parallelcluster-cookbook](#) 패키지를 참조하십시오. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster 버전 3.5.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.5.0이 출시되었습니다.

2023년 2월 20일

개선 사항:

- [AWS ParallelCluster UI](#)를 사용하여 클러스터에 액세스하고 관리할 수 있습니다.
- 워크로드에서 참조할 수 있는 CloudFormation 템플릿에 버전이 지정된 AWS ParallelCluster 정책을 추가합니다.
- 자체 코드에 사용할 수 있는 AWS ParallelCluster Python 라이브러리를 추가하세요.
- 컴퓨팅 노드 부트스트랩 장애 CloudWatch 시 Amazon에 컴퓨팅 노드 콘솔 출력 로깅을 추가합니다.
- 클러스터 생성 실패 시 `describe-cluster` 출력에 실패 코드 및 이유가 포함된 실패 필드 추가
- 하위 프로세스 모듈을 호출하는 동안 악의적인 문자열 삽입을 방지하기 위해 유효성 검사기 추가
- 정적 노드를 프로비저닝하는 동안 클러스터 상태가 PROTECTED 로 변경되면 클러스터 생성이 실패합니다.

변경 사항:

- 버전 22.05.7에서 Slurm 버전 22.05.8으로 업그레이드
- EFA설치 프로그램을 로 업그레이드하십시오. 1.21.0
  - Efa-driver: efa-2.1에서 efa-2.1.1-1 로
  - Efa-config: efa-config-1.11-1에서 efa-config-1.12-1 로
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws: libfabric-aws-1.16.1 에서 libfabric-aws-1.16.1amzn3.0-1 로
  - Rdma-core: rdma-core-43.0-2 에서 rdma-core-43.0-1 로
  - 열기MPI: openmpi40-aws-4.1.4-3 (변경 없음)
- Slurm 컨트롤러 로그를 더 자세하게 만들고 Slurm 절전 플러그인에 대한 추가 로깅 활성화

#### 버그 수정:

- Slurm 회계가 활성화된 경우 클러스터 이름이 40자를 넘지 않는지 확인하여 클러스터 데이터베이스 생성 문제를 해결합니다.

- Amazon EC2 인스턴스 상태 `clustermgtd` 확인이 실패할 경우 재부팅된 컴퓨팅 노드가 교체되는 문제를 수정했습니다. Slurm
- 헤드 노드의 잘못된 IAM 정책으로 인해 다른 계정과 용량 예약을 공유하는 컴퓨팅 노드가 시작되지 않았던 문제를 수정했습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG 파일](#), [aws-parallelcluster-cookbook](#), [aws-parallelcluster-node](#) 및 패키지를 참조하십시오. [aws-parallelcluster-ui](#) GitHub

[AWS ParallelCluster 버전 3.4.1이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.4.1이 출시되었습니다.

2023년 1월 13일

버그 수정:

- 컴퓨팅 노드의 내부 레지스트리에 업데이트가 잘못 적용될 수 있는 Slurm 스케줄러 문제 수정 결과적으로 이 문제가 발생하면 EC2 인스턴스를 사용할 수 없게 되거나 잘못된 인스턴스 유형으로 백업될 수 있습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub



[AWS ParallelCluster 버전 3.4.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.4.0이 출시되었습니다.

2022년 12월 22일

개선 사항:

- 용량 가용성을 올리기 위해 여러 가용 영역에서 노드를 시작하는 기능 지원
- 용량 가용성을 올리기 위해 각 대기열에 여러 서브넷을 지정하는 기능 지원
- [IamResourcePrefix](#) /에 새 구성 매개변수를 추가하여 에서 생성한 IAM 리소스의 경로 및 이름에 대한 접두사를 지정합니다. AWS ParallelCluster
- Lambda AWS ParallelCluster 함수에서 사용하는 Vpc 구성을 지정하기 [LambdaFunctionsVpcConfig](#) 위한 새 구성 섹션 [DeploymentSettings](#) /을 추가합니다.
- 클러스터 업데이트 중에 헤드노드에서 실행할 사용자 지정 스크립트를 지정하는 기능 추가 스크립트는 Slurm을 스케줄러로 사용할 때 [HeadNode/CustomActions/OnNodeUpdated](#) 로 지정할 수 있습니다.

변경 사항:

- 기존 파일 시스템에 대한 Amazon EFS 탑재 대상 생성을 제거합니다.
- 를 사용하여 EFS 파일 시스템을 amazon-efs-utils 마운트합니다. EFS 전송 중 암호화와 IAM 인증된 사용자를 사용하여 파일 시스템을 마운트할 수 있습니다.
- OS7센트와 우분투에 stunnel 5.67을 설치하여 전송 중 암호화를 지원하세요.  
EFS
- EFA설치 프로그램을 (예서) 로 업그레이드하세요.  
1.20.0 1.18.0
  - Efa-driver: efa-1.16.0-1 에서 efa-2.1로
  - Efa-config: efa-config-1.11-1 (변경 없음)
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws:  
libfabric-aws-1.16.0~amzn4.0-1 에서 libfabric-aws-1.16.1 로
  - Rdma-core: rdma-core-41.0-2 에서 rdma-core-43.0-2 로
  - 열기MPI: openmpi40-aws-4.1.4-3 from ()  
openmpi40-aws-4.1.4-2

- Slurm을 버전 22.05.5에서 버전 22.05.7로 업그레이드
- Python을 3.9.15 및 3.7.13에서 3.9.16 및 3.7.16로 업그레이드
- Slurm 22.05.7를 사용하면 IDLE+CLOUD+COMPLETING+POWER\_DOWN+\_RESPONDING 상태의 동적 노드는 비정상적으로 간주되지 않습니다.

변경 사항에 대한 자세한 내용은 [AWS-parallelcluster의 CHANGelog](#) 파일 및 [aws-parallelcluster-cookbook](#) 패키지를 참조하십시오. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster 버전 3.3.1이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.3.1이 출시되었습니다.

2022년 12월 2일

변경 사항:

- AMIs Amazon이 EC2 지원 중단된 지 2년이 지난 후 공식 AWS ParallelCluster 제품을 이제 사용할 수 있습니다.
- 콜드 스타트 페널티를 줄이고 AWS ParallelCluster API 타임아웃을 방지하려면 Lambda의 메모리 크기를 2048로 늘리십시오.

버그 수정:

- Managed FSx for Lustre 파일 시스템의 교체 및 컴퓨팅 플릿 서브넷 ID 변경을 포함하는 클러스터 업데이트 시 데이터 손실을 방지하십시오.
- [SharedStorage](#) DeletionPolicy 가 클러스터 업데이트 작업에 적용됩니다.

변경 사항에 대한 자세한 내용은 의 CHANGELOG [aws-parallelcluster](#) 패키지 파일을 참조하십시오. GitHub

[AWS ParallelCluster 설명서는 hpc6id에만 해당됩니다. 참고:](#)

AWS ParallelCluster 설명서 전용 업데이트 2022년 12월 2일

- AWS ParallelCluster /설정의 hpc6id 인스턴스 유형을 지원하지 않습니다.  
[HeadNodeInstanceType](#)

[AWS ParallelCluster 버전 3.1.5가 출시되었습니다.](#)

AWS ParallelCluster 버전 3.1.5가 출시되었습니다.

2022년 11월 16일

개선 사항:

- 유휴 노드 종료를 방지하는 Slurm 문제를 수정했습니다.
- EFA설치 프로그램을 1.18.0으로 업그레이드
  - Efa-driver: efa-1.16.0-1
  - Efa-config: efa-config-1.9-1 에서 efa-config-1.11-1 로
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws: libfabric-1.13.2 에서 libfabric-aws-1.16.0~amzn4.0-1 로
  - Rdma-core: rdma-core-37.0 에서 rdma-core-41.0-2 로
  - 열기MPI: openmpi40-aws-4.1.4-2 (에서) openmpi40-aws-4.1.1-2

변경 사항:

- 클러스터 업데이트를 lambda:UntagResource 위해 AWS ParallelC

luster API 스택에서 ParallelClusterUserRole 사용하는 항목을 lambda:ListTags 추가 하고 추가합니다.

- 인텔 MPI 라이브러리를 버전 2021 업데이트 6으로 업그레이드하십시오 (버전 2021 업데이트 4에서). 자세한 내용은 [인텔® MPI 라이브러리 2021 업데이트 6을](#) 참조하십시오.
- NVIDIA드라이버를 버전 470.141.03으로 업그레이드 하십시오 (470.103.01에서).
- NVIDIA패브릭 매니저를 버전 470.141.03으로 업그레이드 하십시오 (470.103.01에서).

[변경 사항에 대한 자세한 내용은 aws-parallelcluster의 CHANGELOG 파일 및 패키지를 참조하십시오.](#) [aws-parallelcluster-cookbookaws-parallelcluster-node](#) [GitHub](#)

[AWS ParallelCluster 버전 3.3.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.3.0이 출시되었습니다.

2022년 11월 2일

개선 사항:

- Slurm을 스케줄러로 사용할 때 컴퓨팅 리소스에 대한 다중 인스턴스 할당 구성에 대한 지원 추가 자세한 내용은 [Slurm을 사용하여 여러 인스턴스 유형 할당](#) 항목을 참조하세요.
- 업데이트된 구성을 사용하여 클러스터 업데이트를 통한 [SharedStorage](#) 추가 및 제거 지원 자세한 내용은 [공유 스토리지](#) 항목을 참조하세요.
- 스토리지 보존을 지원하기 위해 [Efs](#) 및 [FsxLustre](#) \_공유 스토리지 설정을 위한 새 구성 파라미터 `DeletionPolicy` 추가
- 새로운 구성 파라미터 [Scheduling /SlurmSettings /Database](#)를 사용하는 Slurm 회계 지원 자세한 내용은 [Slurm회계 담당자: AWS ParallelCluster](#) 단원을 참조하십시오.
- 온디맨드 용량 예약 (ODCR) 및 용량 예약 리소스 그룹에 대한 지원을 추가합니다. 자세한 내용은 [온디맨드 용량](#)



## 예약 (ODCR) 으로 인스턴스

시작 단원을 참조하십시오.

- 새 구성 매개 변수를 추가하여 클러스터에서 지원할 IMDS 버전 또는 클러스터의 빌드 이미지 인프라, [Imds/](#), 및 빌드 [ImdsSupport ImdsSupport](#) , [Imds/](#) 구성을 지정합니다.
- [SlurmQueues /ComputeResources](#) 섹션에서 [Networking /PlacementGroup](#) 지원 추가
- ENI디바이스당 하나로 제한되는 여러 네트워크 인터페이스가 있는 인스턴스에 대한 지원을 추가하세요.
- 연결된 보안 그룹에서 CIDR 블록을 확인하여 외부 Amazon EFS 파일 시스템의 네트워킹 검증을 개선하십시오.
- 구성된 인스턴스 유형이 배치 그룹을 지원하는지 확인하는 검사기 추가
- 안정성과 성능을 높이려면 NFS 스레드를 최소 (256,  $\max(8, \text{num\_cores} * 4)$ ) 로 구성하십시오.
- 빌드 시점에 NFS 설치를 옮겨 구성 시간을 줄이세요.
- 배포 시 AWS ParallelCluster API 생성되고 docker 이미지 빌드 이벤트를 알리는 데

사용되는 EcrImageBuilder SNS 주제에 대해 서버 측 압축을 활성화합니다.

변경 사항:

- [SlurmQueues /Networking /PlacementGroup /Enabled](#)의 동작 변경 이제 모든 컴퓨팅 리소스에 대한 단일 관리형 배치 그룹 대신 각 컴퓨팅 리소스에 대해 고유한 관리형 배치 그룹을 생성합니다.
- [SlurmQueues /Networking /PlacementGroup /Name](#)를 선호되는 명령 방법으로 지원
- 태그 업데이트 시 헤드 노드 교체를 방지하기 위해 시작 템플릿에서 인스턴스 정의로 헤드 노드 태그로 이동
- 시작 템플릿에서 설정된 CpuOptions 를 통하지 않고 cloud-init 를 통해 실행되는 멀티스레딩 비활성화
- 인프라API, Docker 컨테이너 및 클러스터 Lambda 리소스에서 API Python을 버전 3.9로, NodeJS를 버전 16으로 업그레이드합니다.

- aws-parallelcluster-batch-cli 에서 Python 3.6에 대한 지원 제거
- Slurm을 버전 21.08.8-2 에서 버전 22.05.5로 업그레이드
- NVIDIA드라이버를 버전 (에서) 으로 업그레이드합니다. 470.141.03 470.129.06
- NVIDIA패브릭 관리자를 버전 470.141.03 (에서470.129.06 ) 으로 업그레이드합니다.
- NVDIACUDA툴킷을 버전 11.7.1 () 로 업그레이드합니다. from 11.4.4
- AWS ParallelCluster 가상 환경에서 사용되는 Python을 에서 로 업그레이드하십시오. 3.7.13 3.9.15
- EFA설치 프로그램을 버전 1.18.0으로 업그레이드하십시오.
  - Efa-driver: efa-1.16.0-1 (변경 없음)
  - Efa-config: from efa-config-1.10-1 에서 efa-config-1.11-1 로
  - Efa-profile: efa-profile-1.5-1 (변경 없음)
  - Libfabric-aws: libfabric-aws-1.16

- .0~amzn2.0-1 에서  
libfabric-aws-1.16  
.0~amzn4.0-1 로
- Rdma-core: rdma-core  
-37.0 에서 rdma-core  
-41.0-2 로
- 열기MPI: openmpi40-  
aws-4.1.4-2 (에서)  
openmpi40-aws-4.1.  
1-2
- NICE DCV 버전으로 업그레이드 2022.1-13300 (에서 2022.0-12760 ).
- Queues을 위해 SingleSubnetValidator 금지 활성화
- 에필로그가 아직 실행 중일 수 있으므로 노드가 COMPLETING 상태일 때는 DRAIN 노드 교체 금지

#### 버그 수정:

- 잘못된 필터가 전달될 때 AWS ParallelCluster ListClusterLogStreams 명령의 필터 매개 변수 유효성 검사가 실패하도록 수정했습니다.
- 매개 변수 FileSystemId /가 다른 [SharedStorage SharedStorage /EfsSettings](#) 매개 변수와 함께 지정된 경우 유효성 검사에 [EfsSettings](#) 실패하도

록 수정했습니다. 이전에는 `FileSystemId` 이 포함되지 않았습니다.

- 구성의 다른 변경 사항과 [SharedStorage](#) 를 함께 순서를 변경할 때 클러스터 업데이트를 수정하세요.
- 로그를 AWS ParallelCluster API 업로드할 대상을 `UpdateParallelClusterLambdaRole` `CloudWatch` 수정했습니다.
- 쿡북을 실행하기 전에 패키지를 설치할 때 Cinc가 로컬 CA 인증서 번들을 사용하지 않는 문제를 수정했습니다.
- `Build:UpdateOsPackages:Enabled:true` 가 설정되었을 때 `pcluster build-image` 로 `ubuntu`를 업그레이드할 때 멈추는 문제를 수정했습니다.
- 중복 키 오류가 발생하여 YAML 클러스터 구성 파싱이 실패하는 문제를 해결합니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster 설명서 참조만 추가되었습니다. API](#)

AWS ParallelCluster 설명서 전용 업데이트 2022년 10월 27일

- 설명서에 버전 3 [AWS ParallelCluster API 레퍼런스](#)를 추가했습니다.

## [AWS ParallelCluster 버전](#)

### [3.2.1 출시](#)

## AWS ParallelCluster 버전

3.2.1이 출시되었습니다.

2022년 10월 3일

### 개선 사항:

- 호스트 라우팅 테이블을 여러 네트워크 카드에 연결하는 로직을 개선하여 여러 네트워크 카드가 있는 Amazon EC2 인스턴스를 더 잘 지원합니다. NICs.

### 변경 사항:

- NVIDIA 드라이버를 버전 470.141.03으로 업그레이드 하십시오.
- NVIDIA 패브릭 관리자를 버전 470.141.03으로 업그레이드합니다.
- 노드 성능에 부정적인 영향을 줄 수 있는 cron 작업 태스크 man-db 및 mlocate를 비활성화
- 인텔 MPI 라이브러리를 2021.6.0.602로 업그레이드 하십시오.
- 이러한 보안 위협에 대응하여 Python을 3.7.10에서 3.7.13으로 업그레이드하세요.

### 버그 수정:

- 클러스터 구성을 사용할 수 없을 때 DescribeCluster 에 오류가 발생하는 것을 방지

변경 사항에 대한 자세한 내용은 [aws-parallel클러스터용 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub



[AWS ParallelCluster 버전 3.2.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.2.0이 출시되었습니다.

2022년 7월 27일

개선 사항:

- Slurm에 [메모리 기반 스케줄링](#)에 대한 지원 추가
  - Slurm클러스터 구성에서 컴퓨팅 노드의 실제 메모리를 구성합니다.
  - Slurm에서 메모리 기반 스케줄링을 활성화하기 위해 새로운 구성 파라미터 [Scheduling /SlurmSettings /EnableMemoryBasedScheduling](#) 를 추가
  - 컴퓨팅 노드의 스케줄러에 표시되는 메모리의 기본값을 재정의하기 위해 새로운 구성 파라미터 [Scheduling /SlurmQueues /ComputeResources /ScheduleableMemory](#) 를 추가
- 가능한 경우 전체 클러스터가 중지되었다가 다시 시작되지 않도록 클러스터 구성 업데이트의 유연성 개선  
컴퓨팅 노드에 구성 업데이트 및 교체가 필요할 때 사용할 기본 전략을 설정하

기 위해 새 구성 파라미터 [Scheduling /SlurmSettings /QueueUpdateStrategy](#) 를 추가

- Amazon 인스턴스의 용량 부족 문제가 발생할 때 사용 가능한 컴퓨팅 리소스에 대한 장애 조치 메커니즘을 개선합니다. EC2 용량 부족으로 노드 시작이 실패할 경우 [구성 가능한 시간까지 컴퓨팅 노드를 비활성화](#)
- 기존 [FSx for ONTAP](#) 및 [FSx for OpenZFS](#) 파일 시스템을 탑재하기 위한 지원 추가
- 기존 [Amazon Elastic File Systems](#)의 여러 인스턴스 탑재 지원 ([FSxLustre용, 개방형 ZFS 파일 FSx 시스템용 ONTAP](#)) FSx 을 위한 지원을 추가합니다.
- 새 파일 시스템을 생성할 때 [Lustre Persistent\\_2 배포 유형에 FSx 대한](#) 지원을 추가 하십시오.
- 마법사를 사용할 때 지원되는 인스턴스 유형을 EFA 활성화하라는 메시지를 사용자에게 표시합니다. [pcluster configure](#)
- Slurm을 사용한 컴퓨팅 노드 재부팅 지원 추가

- 노드의 수동 Slurm 전원 차단도 고려하도록 전원 상태 처리를 개선하십시오.
- NVIDIAGDRCopyAMIs2.3을 제품에 설치하면 지연 시간이 짧은 GPU 메모리 복사가 가능합니다.

#### 변경 사항:

- EFA설치 프로그램을 버전 1.17.2로 업그레이드합니다.
  - EFA드라이버: efa-1.16.0-1
  - EFA구성: efa-config-1.10-1
  - EFA프로필: efa-profile-1.5-1
  - Libfabric: libfabric-aws-1.16.0~amzn2.0-1
  - RDMA코어: rdma-core-41.0-2
  - 오픈MPI: openmpi40-aws-4.1.4-2
- 2022.0-12760 NICE DCV 버전으로 업그레이드하세요.
- 드라이버를 버전 470.129.06으로 업그레이드합니다NVIDIA.
- NVIDIA패브릭 매니저를 버전 470.129.06으로 업그레이드합니다.

- 루트 볼륨과 EBS 추가 볼륨 모두에서 기본 볼륨 유형을 gp2에서 gp3으로 변경합니다.
- Lustre 파일 FSx 시스템의 변경 내용은 다음과 같습니다.  
AWS ParallelCluster
  - 기본 배포 유형을 Scratch\_2 로 변경
  - Lustre 서버 버전을 2.12로 변경
- 기존 Placement Group /Id를 전달할 때 [Placement Group /Enabled](#)를 true로 설정할 필요가 없습니다.
- Placement Group /Enabled가 명시적으로 false로 설정된 경우 PlacementGroup /Id 설정을 허용하지 않습니다.
- AWS ParallelCluster에서 만든 모든 리소스에 parallelcluster:cluster-name 태그 추가
- 클러스터 lambda:ListTags 업데이트를 lambda:UntagResource 위해 AWS ParallelCluster API 스택에 추가하고 ParallelClusterUserRole 사용합니다.
- 구성 매개변수 HeadNode Imds Secured //가 활성화

된 경우 루트 및 클러스터 관리자 사용자만 IPv6 액세스할 수 있도록 제한합니다.

#### IMDS

- 사용자 AMI 지정에서는 ParallelCluster 기본값인 35GiB 대신 AMI 루트 볼륨 크기를 사용합니다. 클러스터 구성 파일에서 값을 변경할 수 있습니다.
- 구성 파라미터 `Scheduling /SlurmQueues /ComputeResources /SpotPrice` 가 필요한 최소 스팟 요청 이행 가격보다 낮으면 컴퓨팅 플릿이 자동으로 비활성화됩니다.
- 업데이트 중에 섹션을 추가하거나 제거할 때 변경 세트의 `requested_value` 값과 `current_value` 값을 표시합니다.
- 여러 네트워크 카드로 인스턴스를 구성할 `configure_nw_interface.sh` 때 충돌을 피하려면 AMIs 디렉터리에서 사용할 수 있는 `aws-ubuntu-eni-helper` 서비스를 비활성화하십시오.
- Python 3.6에 대한 지원 제거
- 여러 네트워크 카드로 인스턴스를 구성할 때는 모든 네트워크 인터페이스를 9001로 설정하십시오 MTU.

- 컴퓨팅 노드를 구성할 때 후행 점을 제거합니다. FQDN
- POWERING\_DOWN 에서 정적 노드를 관리
- 작업이 아직 실행 중일 수 있으므로 POWER\_DOWN 의 동적 노드를 교체하지 않습니다.
- 클러스터 구성에서 Scheduling 파라미터가 업데이트된 경우에만 클러스터 업데이트 시기에 clustermgtd 및 slurmctld 대몬(daemon)을 재시작
- slurmctld 및 slurmd systemd 서비스 파일 업데이트
- 구성 파라미터 HeadNode Imds Secured //가 활성화된 경우 루트 및 클러스터 관리자 사용자로만 IPv6 액세스를 제한하십시오. IMDS
- 노드를 사용할 수 없을 때 작업을 다시 시작하기 전에 대기 중인 작업을 기다려야 하는 시간을 AuthInfo=cred\_expire=70 줄이도록 Slurm 구성을 설정하십시오.
- 서드 파티 쿨북 종속성 업그레이드:
  - apt-7.4.0에서 apt-7.4.2로
  - 라인-4.0.1번에서 라인-4.5.2번으로

- openssh-2.9.1에서 openssh-2.10.3으로
- pyenv-3.4.2에서 pyenv-3.5.1로
- selinux-3.1.1에서 selinux-6.0.4로
- yum-6.1.1에서 yum-7.4.0으로
- yum-epel-4.1.2에서 yum-epel-4.5.0으로

#### 버그 수정:

- 사용자 지정 빌드 시 AWS ParallelCluster 검증 및 테스트 단계를 건너뛰도록 기본 동작을 수정했습니다. AMI
- `computemgtd` 의 파일 핸들 누수 문제를 수정했습니다.
- 응답에서 실행된 인스턴스를 아직 사용할 수 없어서 간헐적으로 시작된 인스턴스가 즉시 종료되는 경합 상태를 수정하십시오. `EC2 DescribeInstances`
- Arm 프로세서의 인스턴스 유형에서 `DisableSimultaneousMultithreading` 파라미터에 대한 지원을 수정했습니다.
- 이전 버전에서 업그레이드할 때 발생하는 AWS ParallelCluster API 스택 업데이트 실패를 수정합니다.

EcrImageDeletionLambdaRole 의 ListImagePipelineImages 작업에 사용되는 리소스 패턴 추가

- FSxLustre용 파일 시스템을 생성할 때 Amazon S3에서 가져오거나 내보내는 데 필요한 누락된 권한을 AWS ParallelCluster API 추가하는 문제를 수정했습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub



[AWS ParallelCluster 올해 현재  
까지 설명서 전용 업데이트](#)

AWS ParallelCluster 설명서 전용 업데이트. 2022년 7월 6일

새로운 섹션:

- [모범 사례: 예산 알림 V3](#)
- [모범 사례: 클러스터를 새 AWS ParallelCluster 마이너 또는 패치 버전으로 이동 V3](#)
- [Amazon S3 작업 V3](#)
- [스팟 인스턴스 작업 V3](#)
- [Slurm 클러스터 보호 모드 V3](#)
- [AWS ParallelCluster 리소스 및 태깅 V3](#)
- [아마존 CloudWatch 대시보드 V3](#)
- [Amazon CloudWatch Logs 와 통합 V3](#)
- [Elastic Fabric Adapter V3](#)
- [AWS ParallelCluster AMI 사용자 지정 V3](#)
- [온디맨드 용량 예약 \(ODCR\) 으로 인스턴스 시작 V3](#)
- [AMI 패치 적용 및 Amazon EC2 인스턴스 교체 V3](#)
- [AWS ParallelCluster 작동 방식 V3](#)
- [AWS KMS 키를 사용한 공유 스토리지 암호화 구성 V3](#)
- [다중 대기열 모드 클러스터에서 작업 실행 V3](#)

- [AWS ParallelCluster API 사용하기 V3](#)

섹션 업데이트:

- [모범 사례: 네트워크 성능 V3: Elastic Fabric Adaptor](#) 사용에 대한 모범 사례가 추가되었습니다.
- [AWS Identity and Access Management 권한 입력 AWS ParallelCluster V3: 다양한 업데이트 및 Amazon FSx for Lustre를 사용할 때의 추가 AWS ParallelCluster pcluster 사용자 정책](#) 추가
- [AWS ParallelCluster 문제 해결 V3: 다양한 업데이트](#)

## [AWS ParallelCluster 버전](#)

### [3.1.4 출시](#)

AWS ParallelCluster 버전  
3.1.4가 출시되었습니다.

2022년 5월 16일

개선 사항:

- 비밀이 존재하지 않는 경우 [Directory Service / PasswordSecretArn](#) 가 실패하도록 검증 추가

JWT인증 Slurm 활성화에 대한 지원을 추가합니다.

변경 사항:

- 버전 Slurm 21.08.8-2로 업그레이드하십시오.
- 지원을 받아 빌드하세요.  
Slurm JWT
- 기존 Placement Group /Id를 전달할 때 [Placement Group /Enabled](#)를 true로 설정할 필요가 없습니다.
- 클러스터 생성 및 이미지 `lambda:TagResource` 생성에 `ParallelClusterUserRole` 사용 되는 ParallelCluster API 스택에 추가합니다.

버그 수정:

- `--filters` 옵션과 함께 `export-cluster-logs` 명령을 사용할 때 클러스터의 로그를 내보내는 기능을 수정했습니다.
- `/home` 공유 디렉터리를 사용하여 다중 노드 병렬 작업 실행을 조정하도록 AWS Batch Docker 진입점을 수정했습니다.
- 용량이 충분하지 않아 장애가 발생한 정적 노드를 부스트랩 장애 노드로 처리하지 않도록 Slurm 비정상 정적 노드를 다운으로 설정할 때 노드 주소를 재설정합니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster 버전 3.1.3이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.1.3이 출시되었습니다.

2022년 4월 20일

개선 사항:

- 예를 들어 SSH 로그인 중, 다른 사용자로 전환할 때, 다른 사용자로 명령을 실행할 때 HOME 디렉토리 생성과 함께 SSH 키 생성을 실행합니다.
- 구성 파라미터 [Directory Service DomainName](#) /에 LDAP 고유 이름 FQDN 모두와 고유 이름에 대한 지원을 추가합니다. 이제 새 유효성 검사기가 두 구문을 모두 검사합니다.
- 헤드 노드에 배포된 새 `update_directory_service_password.sh` 스크립트는 SSSD 구성에서 Active Directory 암호의 수동 업데이트를 지원합니다. 암호는 클러스터 구성의 AWS Secrets Manager as에 의해 검색됩니다.
- 기본값이 VPC 없는 환경에서 API 인프라를 배포할 수 있는 지원을 추가합니다.

변경 사항:

- `build-image` 명령을 통해 AMIs 생성된 공식 x86\_64에 서 더 깊은 C-State를 비활성화하여 고성능 AMIs 및 짧은 지연 시간을 보장하십시오.
- OS 패키지 업데이트 및 보안 수정
- Amazon Linux 2 기본 이미지를 커널 AMIs 5.10과 함께 사용하도록 변경합니다.

#### 버그 수정:

- 새로운 Image Builder 정책으로 인해 이미지 빌드가 성공한 `DELETE_FAILED` 후 빌드 EC2 이미지 스택을 수정했습니다.
- 구성 매개변수에 여러 도메인 주소가 포함된 경우 구성 매개 [Directory Service](#) 변수/`ldap_uri` [DomainAddr](#) SSSD속성으로의 변환을 수정하십시오.

변경 사항에 대한 자세한 내용은 `CHANGELOG` [aws-parallelcluster](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster 버전 3.1.2가 출시되었습니다.](#)

AWS ParallelCluster 버전 3.1.2가 출시되었습니다.

2022년 3월 2일

변경 사항:

- Slurm을 버전 21.08.5에서 버전 21.08.6로 업그레이드

버그 수정:

- 인터넷 액세스 없이 서브넷에 클러스터를 배포할 때 컴퓨팅 노드의 /etc/hosts 파일 업데이트를 수정
- 컴퓨팅 노드 부트스트랩이 클러스터에 가입하기 전에 임시 드라이브가 초기화될 때까지 대기하도록 수정했습니다.

변경 사항에 대한 자세한 내용은 의 [aws-parallelcluster](#) 패키지 CHANGELOG 파일을 참조하십시오. GitHub

[AWS ParallelCluster 버전](#)[3.1.1 출시](#)

## AWS ParallelCluster 버전

3.1.1이 출시되었습니다.

2022년 2월 10일

- AWS Directory Service를 통해 관리되는 [Active Directory \(AD\) 도메인과 통합](#)하여 다중 사용자 클러스터 환경에 대한 지원 추가
- 클러스터 구성 파일의 [UseEc2Hostnames](#)에 대한 지원 추가 true로 설정하면 컴퓨팅 노드에 Amazon EC2 기본 호스트 이름 (예: ip-1-2-3-4)을 사용합니다.
- [인터넷에 접속할 수 없는 서브넷](#)에서의 클러스터 생성 지원
- 대기열당 여러 컴퓨팅 인스턴스 유형에 대한 지원 추가
- 카드가 있는 온 인스턴스를 사용하여 GPU 스케줄링하기 위한 지원을 추가합니다. Slurm ARM NVIDIA
- 예 `()`, `cluster-name (-n)`, `region ()` 및 `cluster-configuration /image-configuration ()`에 대한 약식 플래그를 추가합니다. `-r image-id -i -c`  
AWS ParallelCluster CLI
- Lustre 매개 변수에 대한 NEW\_CHANGED\_DELETED FSx 옵션 지원을 추가합니다. [AutoImportPolicy](#)



- 컴퓨팅 노드에서 사용하는 EC2 LaunchTemplates 리소스에 `parallelcluster:compute-resource-name` 태그를 추가합니다.
- 일부 헤드 노드 및/또는 대기열에 SecurityGroups 파라미터가 지정된 경우 사용자 지정 보안 그룹으로부터의 인바운드 연결을 허용하도록 클러스터 내에 생성된 보안 그룹을 개선
- 에 대한 NVIDIA 드라이버와 CUDA 라이브러리를 설치합니다ARM.

#### 변경 사항:

- Slurm을 버전 20.11.8에서 버전 21.08.5로 업그레이드
- Slurm플러그인을 버전으로 21.08 (에서20.11) 업그레이드하세요.
- NICE DCV버전으로 업그레이드 2021.3-11591 (from2021.1-10851 ).
- NVIDIA드라이버를 버전 470.103.01 (에서470.57.02 )으로 업그레이드합니다.
- NVIDIA패브릭 관리자를 버전 470.103.01 (에

서470.57.02 ) 으로 업그레이드합니다.

- CUDA를 버전 11.4.0에서 버전 11.4.4로 업그레이드
- [인텔은](#) 버전 2021 업데이트 4로 MPI 업데이트했습니다 (버전 2019 업데이트 8에서 업데이트). 자세한 내용은 [인텔® MPI 라이브러리 2021 업데이트 4](#)를 참조하십시오.
- PMIx을 버전 3.1.5에서 버전 3.2.3로 업그레이드
- /home/logs/compute 에 장애가 발생한 컴퓨팅 노드의 덤핑 제거 컴퓨팅 노드 로그 파일은 Amazon 콘솔 CloudWatch 로그와 Amazon EC2 콘솔 로그에서 사용할 수 있습니다.
- SlurmQueues 및 ComputeResources 길이 검사기를 차단할 수 있게 했습니다.
- Amazon Linux 2에서 인스턴스 시작 시 패키지 업데이트 비활성화
- AWS ParallelCluster 사용자 지정 이미지를 EC2 ImageBuilder 만들 때 Amazon 고급 이미지 메타데이터를 비활성화합니다.
- cloud-init 데이터 소스를 로 명시적으로 설정합니다. EC2 이를 통해 Ubuntu

및 CentOS 플랫폼의 부팅 시간이 절약됩니다.

- 컴퓨팅 플릿 시작 템플릿 이름에 인스턴스 유형 대신 컴퓨팅 리소스 이름 사용
- pcluster 출력에 원치 않는 텍스트가 포함되지 않도록 stderr 및 stdout을 CLI 로그 파일로 리디렉션하십시오.  
CLI
- 구성/설치 레시피를 기본 콕북과 호출되는 별도의 콕북으로 이동 기존 진입점은 유지되며 이전 버전과 호환됩니다.
- 클러스터 생성 중에 인터넷에 접속하지 않도록 AMI 빌드 중에 인텔 HPC 플랫폼의 종속성을 다운로드하십시오.
- Slurm 노드를 구성할 때 컴퓨팅 리소스 이름에서 - 제거 금지
- NVIDIA 드라이버가 설치되지 않은 GPUs Slurm 상태에서는 구성하지 마십시오.
- BatchUserRole 에서 `ecs:ListContainerInstances` 권한 수정
- 이전에 None 접두사로 내보낸 접두사가 지정되지 않은 경우 클러스터 로그 내보내기를 수정
- 클러스터 업데이트 실패 시 롤백이 수행되지 않는 문제를 수정

- BatchUserRole 에서 `ecs:ListContainerInstances` 권한 수정
- 지원되지 않는 KmsKeyId 항목이 지정된 경우 오류를 발생시켜 HeadNode에 대한 RootVolume 스키마를 수정
- Amazon에서 FSx 누락된 지표가 CloudWatch 대시보드에 표시되도록 수정합니다.
- EfaSecurityGroupValidator 수정 이전에는 사용자 지정 보안 그룹이 제공되고 EFA 활성화된 경우 잘못된 오류가 발생할 가능성이 있었습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 [aws-parallelcluster-cookbook](#) 패키지를 참조하십시오. [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster 버전 3.0.3이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.0.3이 출시되었습니다.

2022년 1월 17일

- Amazon Linux 2에서 log4j-cve-2021-44228-hotpatch 에이전트(Log4jHotPatch )를 비활성화하여 잠재적인 성능 저하 방지 자세한 내용은 [Apache Log4j용 Amazon Linux 핫패치 발표](#)를 참조하세요.

변경 사항에 대한 자세한 내용은 의 [aws-parallel클러스터](#) 및 패키지 CHANGELOG 파일을 참조하십시오. [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster 버전 3.0.2가 출시되었습니다.](#)

AWS ParallelCluster 버전 3.0.2가 출시되었습니다.

2021년 11월 5일

[Elastic Fabric Adapter](#) 설치 프로그램을 1.14.1로 업그레이드

- EFA구성: efa-config-1.9-1 (부터) efa-config-1.9
- EFA프로필: efa-profile-1.5-1 (보낸 efa-profile-1.5 사람)
- EFA커널 모듈: efa-1.14.2 (from efa-1.13.0 )
- RDMA코어: rdma-core-37.0 (보낸 사람 rdma-core-35 )
- Libfabric: libfabric-1.13.0 에서 libfabric-1.13.2 로
- 오픈MPI: openmpi4aws-4.1.1-2 (변경 없음)

GPUDirectRDMA인스턴스 유형에서 지원하는 경우 항상 활성화됩니다. [GdrSupport](#) 구성 옵션은 효과가 없습니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster의 CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

## [AWS ParallelCluster 버전 3.0.1이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.0.1이 출시되었습니다.

2021년 10월 27일

### 클러스터 구성 마이그레이션 도구

- 이제 고객은 클러스터 구성을 AWS ParallelCluster 버전 2 형식에서 YAML 기반 AWS ParallelCluster 버전 3 형식으로 마이그레이션할 수 있습니다. 자세한 내용은 [pcluster3-config-converter](#) 단원을 참조하십시오.

### 헤드 노드를 중지할 수 있습니다.

- 컴퓨팅 플릿을 중지한 후에는 Amazon EC2 콘솔 또는 [AWS CLI stop-instances](#) 명령을 사용하여 헤드 노드를 중지했다가 나중에 다시 시작할 수 있습니다.

### 기본 파일 읽기 AWS 리전

~/.aws/config

- [pcluster](#) 명령의 경우 구성 파일, 환경 또는 명령줄에 AWS 리전 지정되지 않은 경우 ~/.aws/config 파일 [default] 섹션의 region 설정에 AWS 리전

지정된 기본값이 사용됩니다.

변경 사항에 대한 자세한 내용은 [aws-parallelcluster CHANGELOG](#) 파일 및 패키지를 참조하십시오. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub



[AWS ParallelCluster 버전 3.0.0이 출시되었습니다.](#)

AWS ParallelCluster 버전 3.0.0이 출시되었습니다.

2021년 9월 10일

### Amazon API Gateway를 통한 클러스터 관리 지원

- 이제 고객은 Amazon API Gateway를 사용하여 HTTP 엔드포인트를 통해 클러스터를 관리하고 배포할 수 있습니다. 이를 통해 스크립트 기반 또는 이벤트 기반 워크플로의 새로운 가능성이 열립니다.

AWS ParallelCluster 명령 줄 인터페이스 (CLI) 도 이와 API 호환되도록 재설계되었으며 새로운 JSON 출력 옵션을 포함합니다. 이 새로운 기능을 통해 고객은 도 사용하여 유사한 빌딩 블록 기능을 구현할 수 CLI 있습니다.

사용자 지정 AMI 생성 기능이 개선되었습니다.

- 이제 고객은 EC2 Image Builder를 AMIs 사용하여 사용자 지정을 생성하고 관리하는 보다 강력한 프로세스를 이용할 수 있습니다. 이제 별도의 AWS ParallelCluster 구성 파일을 통해 사용자 정의를 AMIs 관리할 수 있

으며 명령줄 인터페이스에  
서 [pcluster build-image](#)  
[ge](#) 명령을 사용하여 사용자  
정의를 생성할 수 있습니다.

AWS ParallelCluster

변경 사항에 대한 자세한  
내용은 [aws-parallelcluster](#)  
[CHANGELOG](#) 파일 및 패키지를  
참조하십시오. [aws-parallelcluster-cookbook](#)  
[aws-parallelcluster-node](#) GitHub

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.