사용자 가이드

AWS 데드라인 클라우드

버전 latest

AWS 데드라인 클라우드: 사용자 가이드

Table of Contents

데드라인 클라우드란? 1
데드라인 클라우드의 특징 1
개념 및 용어 2
데드라인 클라우드 시작하기 4
데드라인 클라우드 액세스 4
관련 서비스 5
데드라인 클라우드 작동 방식 5
..... 6
데드라인 클라우드의 권한 6
데드라인 클라우드를 통한 소프트웨어 지원 7
시작하기 8
AWS 계정설정 8
모니터 설정 9
1단계: 모니터 설정 9
2단계: 팜 세부 정보 정의 12
3단계: 대기열 세부 정보 정의 13
4단계: 플릿 세부 정보 정의 14
5단계: 작업자 요구 사항 구성 14
6단계: 액세스 수준 정의 15
7단계: 검토 및 생성 15
개발자 워크스테이션 설정 15
1단계: 팜 생성 16
2단계: 워커 에이전트 실행 20
3단계: 작업 제출 및 실행 22
4단계: 첨부 파일이 있는 작업 실행 29
5단계: 서비스 관리형 플릿 추가 38
6단계: 팜 리소스 정리 41
제출자 설정 43
1단계: 데드라인 클라우드 제출자 설치 44
2단계: 데드라인 클라우드 모니터 설치 및 설정 51
3단계: 데드라인 클라우드 제출자 시작 54
팜 사용 58
모니터 사용 59
데드라인 클라우드 모니터 URL을 공유하세요. 59

데드라인 클라우드 모니터를 엽니다. 60
대기열 및 플릿 세부 정보 보기 61
작업, 단계, 작업 보기 및 관리 62
직무 세부 정보 보기 63
단계 보기 64
작업 보기 65
로그 보기 65
완료된 결과물 다운로드 66
농장 68
팜 생성 68
팜 삭제 68
팜 편집 69
대기열 70
대기열 생성 70
대기열 환경 만들기 72
기본 Conda 대기열 환경 73
대기열 삭제 74
대기열 편집 74
대기열과 플릿을 연결합니다. 74
플릿 관리 75
서비스 관리 플릿 75
VFX 플랫폼 76
고객 관리 차량 77
CMF 만들기 78
작업자 호스트 설정 83
액세스 관리 88
작업용 소프트웨어 설치 90
보안 인증 구성 91
AMI 생성 92
플릿 인프라 생성 95
라이선스 엔드포인트에 연결 105
사용자 관리 109
모니터의 사용자 및 그룹을 관리합니다. 109
팜, 큐, 플릿의 사용자 및 그룹을 관리합니다. 111
작업 113
작업 제출 114

작업 제출을 위한 추가 옵션 116
작업 일정 잡기 117
플릿 호환성을 결정하세요. 118
플릿 스케일링 119
세션 120
단계 종속성 121
작업 상태 123
작업 수정 125
작업 처리 130
작업 문제 해결 131
작업 생성이 실패한 이유는 무엇인가요? 131
내 작업이 호환되지 않는 이유는 무엇인가요? 131
작업이 준비되지 않은 이유는 무엇인가요? 132
작업이 실패한 이유는 무엇인가요? 132
내 단계가 보류 중인 이유는 무엇인가요? 132
스토리지 133
Job 첨부 133
작업 첨부 S3 버킷의 암호화 134
S3 버킷의 작업 첨부 파일 관리 135
가상 파일 시스템 135
공유 스토리지 138
데드라인 클라우드의 스토리지 프로필 138
예산 및 사용량 관리 140
비용 가정 140
예산 관리자 사용 141
전제 조건 141
액세스 예산 관리자 142
예산 생성 142
예산 보기 143
예산 편집 144
예산 비활성화하기 144
사용 현황 탐색기 사용 144
전제 조건 145
사용량 탐색기를 엽니다. 145
사용 현황 탐색기를 사용하세요. 144
비용 관리 147

비용 관리 모범 사례 ... 148
보안 ... 151
데이터 보호 ... 152
저장된 데이터 암호화 ... 153
전송 중 암호화 ... 153
키 관리 ... 153
인터네트워크 트래픽 개인 정보 보호 ... 163
옵트아웃 ... 163
ID 및 액세스 관리 ... 164
고객 ... 164
ID를 통한 인증 ... 165
정책을 사용한 액세스 관리 ... 168
데드라인 클라우드가 IAM과 함께 작동하는 방식 ... 170
자격 증명 기반 정책 예시 ... 177
AWS 관리형 정책 ... 180
문제 해결 ... 184
규정 준수 확인 ... 186
복원력 ... 187
인프라 보안 ... 187
구성 및 취약성 분석 ... 188
교차 서비스 혼동된 대리인 방지 ... 188
AWS PrivateLink ... 190
고려 사항 ... 190
Deadline Cloud 엔드포인트 ... 190
엔드포인트 생성 ... 191
보안 모범 사례 ... 192
데이터 보호 ... 192
IAM 권한 ... 193
사용자 및 그룹으로 작업 실행 ... 193
네트워킹 ... 193
Job 데이터 ... 194
팜 구조 ... 194
Job 첨부 대기열 ... 195
사용자 지정 소프트웨어 버킷 ... 197
워커 호스트 ... 197
워크스테이션 ... 199

모니터링 200
로 로그하기 CloudTrail 201
데드라인 클라우드 정보는 CloudTrail 201
데드라인 클라우드 로그 파일 항목 이해 205
를 통한 모니터링 CloudWatch 206
이벤트에 따른 조치 EventBridge 208
차량 크기 권장 변경 208
할당량 210
AWS CloudFormation 리소스 211
데드라인 클라우드 및 AWS CloudFormation 템플릿 211
에 대해 자세히 알아보십시오. AWS CloudFormation 211
사용 설명서 기록 212
AWS 용어집 213
..... ccxiv

AWS 데드라인 클라우드란?

데드라인 클라우드는 디지털 콘텐츠 생성 파이프라인 및 워크스테이션에서 직접 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스에서 렌더링 프로젝트 및 작업을 생성하고 관리하는 데 사용할 수 있습니다. AWS 서비스

데드라인 클라우드는 콘솔 인터페이스, 로컬 애플리케이션, 명령줄 도구 및 API를 제공합니다. Deadline Cloud를 사용하면 팜, 플릿, 작업, 사용자 그룹 및 스토리지를 생성, 관리 및 모니터링할 수 있습니다. 또한 하드웨어 요구 사항을 지정하고, 특정 워크로드를 위한 환경을 만들고, 프로덕션에 필요한 콘텐츠 제작 도구를 Deadline Cloud 파이프라인에 통합할 수 있습니다.

Deadline Cloud는 모든 렌더링 프로젝트를 한 곳에서 관리할 수 있는 통합 인터페이스를 제공합니다. 사용자를 관리하고, 사용자에게 프로젝트를 할당하고, 작업 역할에 대한 권한을 부여할 수 있습니다.

주제

- 데드라인 클라우드의 특징
- 데드라인 클라우드의 개념 및 용어
- 데드라인 클라우드 시작하기
- 데드라인 클라우드 액세스
- 관련 서비스
- 데드라인 클라우드 작동 방식

데드라인 클라우드의 특징

Deadline Cloud가 시각적 컴퓨팅 워크로드를 실행하고 관리하는 데 도움이 되는 몇 가지 주요 방법은 다음과 같습니다.

- 팜, 대기열, 플릿을 빠르게 생성하세요. 농장의 상태를 모니터링하고 농장 운영 및 작업에 대한 통찰력을 얻으세요.
- Deadline Cloud 사용자 및 그룹을 중앙에서 관리하고 권한을 할당하세요.
- 를 사용하여 프로젝트 사용자 및 외부 ID 공급자의 로그인 보안을 관리하세요. AWS IAM Identity Center
- AWS Identity and Access Management (IAM) 정책 및 역할을 사용하여 프로젝트 리소스에 대한 액세스를 안전하게 관리합니다.

- 태그를 사용하여 프로젝트 리소스를 구성하고 빠르게 찾을 수 있습니다.
- 프로젝트 리소스 사용량과 프로젝트의 예상 비용을 관리하세요.
- 클라우드 또는 오프라인 렌더링을 지원하는 다양한 컴퓨팅 관리 옵션을 제공합니다.

데드라인 클라우드의 개념 및 용어

데드라인 클라우드를 시작하는 데 도움이 되도록 이 항목에서는 AWS 데드라인 클라우드의 몇 가지 주요 개념과 용어에 대해 설명합니다.

예산 관리자

예산 관리자는 데드라인 클라우드 모니터의 일원입니다. 예산 관리자를 사용하여 예산을 만들고 관리할 수 있습니다. 또한 이를 사용하여 활동을 예산 범위 내로 제한할 수 있습니다.

데드라인 클라우드 클라이언트 라이브러리

클라이언트 라이브러리에는 Deadline Cloud를 관리하기 위한 명령줄 인터페이스와 라이브러리가 포함되어 있습니다. 기능에는 Open Job Description 사양에 기반한 작업 번들을 Deadline Cloud에 제출하고, 작업 첨부 출력을 다운로드하고, 명령줄 인터페이스를 사용한 팜 모니터링이 포함됩니다.

디지털 콘텐츠 제작 애플리케이션 (DCC)

디지털 콘텐츠 제작 애플리케이션 (DCC) 은 디지털 콘텐츠를 만드는 타사 제품입니다. DCC의 예로는,, 등이 있습니다. Maya Nuke Houdini 데드라인 클라우드는 특정 DCC를 위한 구직자 통합 플러그인을 제공합니다.

팜

팜은 프로젝트 리소스가 있는 곳입니다. 대기열과 플릿으로 구성되어 있습니다.

플릿

플릿은 렌더링을 수행하는 작업자 노드 그룹입니다. 작업자 노드는 작업을 처리합니다. 플릿을 여러 대기열에 연결할 수 있고 대기열을 여러 플릿에 연결할 수 있습니다.

작업

작업은 렌더링 요청입니다. 사용자가 작업을 제출합니다. 작업에는 단계 및 작업으로 요약된 특정 작업 속성이 포함됩니다.

Job 첨부

작업 첨부는 작업의 입력 및 출력을 관리하는 데 사용할 수 있는 Deadline Cloud 기능입니다. 작업 파일은 렌더링 프로세스 중에 작업 첨부 파일로 업로드됩니다. 이러한 파일은 텍스처, 3D 모델, 조명 장비 및 기타 유사한 항목일 수 있습니다.

작업 속성

Job 속성은 렌더 작업을 제출할 때 정의하는 설정입니다. 일부 예로는 프레임 범위, 출력 경로, 작업 첨부 파일, 렌더링 가능한 카메라 등이 있습니다. 속성은 렌더가 제출된 DCC에 따라 달라집니다.

작업 템플릿

작업 템플릿은 런타임 환경과 Deadline Cloud 작업의 일부로 실행되는 모든 프로세스를 정의합니다.

대기열

큐는 제출된 작업을 찾고 렌더링을 예약하는 곳입니다. 렌더링을 성공적으로 만들려면 대기열을 플릿과 연결해야 합니다. 대기열은 여러 플릿과 연결될 수 있습니다.

대기열-플릿 연결

대기열이 플릿과 연결되면 대기열-집합 연결이 있습니다. 연결을 사용하여 플릿의 작업자를 해당 대기열에 있는 작업으로 스케줄링할 수 있습니다. 연결을 시작하고 중지하여 작업 일정을 제어할 수 있습니다.

단계

단계는 작업에서 실행하는 특정 프로세스 중 하나입니다.

데드라인 클라우드 제출자

데드라인 클라우드 제출자는 디지털 콘텐츠 제작 (DCC) 플러그인입니다. 아티스트는 이를 사용하여 익숙한 타사 DCC 인터페이스에서 작업을 제출합니다.

Tags

태그는 AWS 리소스에 할당할 수 있는 레이블입니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다.

태그를 사용하면 AWS 리소스를 다양한 방식으로 분류할 수 있습니다. 예를 들어, 계정의 Amazon EC2 인스턴스에 대해 각 인스턴스의 소유자나 스택 수준을 추적하는 데 도움이 되는 태그 세트를 정의할 수 있습니다.

목적, 소유자 또는 환경별로 AWS 리소스를 분류할 수도 있습니다. 이 방법은 같은 유형의 리소스가 많을 때 유용합니다. 할당한 태그를 기반으로 특정 리소스를 빠르게 식별할 수 있습니다.

작업

작업은 렌더링 단계의 단일 구성 요소입니다.

사용 기반 라이선스 (UBL)

UBL (사용 기반 라이선스) 은 일부 타사 제품에 사용할 수 있는 온디맨드 라이선스 모델입니다. 이 모델은 종량 과금제이며 사용한 시간과 분 수에 따라 요금이 부과됩니다.

사용량 탐색기

사용 탐색기는 데드라인 클라우드 모니터의 기능입니다. 대략적인 예상 비용 및 사용량을 제공합니다.

작업자

작업자는 플릿에 속하며 Deadline Cloud에서 지정한 작업을 실행하여 단계와 작업을 완료합니다. 작업자는 Amazon Logs에 작업 작업 CloudWatch 로그를 저장합니다. 또한 작업자는 작업 첨부 기능을 사용하여 입력과 출력을 Amazon Simple Storage Service (Amazon S3) 버킷에 동기화할 수 있습니다.

데드라인 클라우드 시작하기

데드라인 클라우드를 사용하면 Amazon EC2 인스턴스 구성 및 Amazon Simple Storage Service (Amazon S3) 버킷과 같은 기본 설정 및 리소스가 포함된 렌더 팜을 빠르게 만들 수 있습니다.

렌더 팜을 생성할 때 설정과 리소스를 정의할 수도 있습니다. 이 방법을 사용하면 기본 설정과 리소스를 사용하는 것보다 시간이 더 걸리지만 더 세밀하게 제어할 수 있습니다.

Deadline Cloud Concepts 및 용어에 익숙해지면 시작하기를 참조하여 팜 만들기, 사용자 추가, 유용한 정보 링크에 대한 step-by-step 지침을 확인하세요.

데드라인 클라우드 액세스

다음과 같은 방법으로 데드라인 클라우드에 액세스할 수 있습니다.

- Deadline Cloud 콘솔 - 브라우저에서 콘솔에 액세스하여 팜과 해당 리소스를 만들고 사용자 액세스를 관리합니다. 자세한 내용은 시작하기를 참조하세요.
- Deadline Cloud 모니터 — 우선 순위 및 작업 상태 업데이트를 포함하여 렌더링 작업을 관리합니다. 팜을 모니터링하고 로그와 작업 상태를 확인하세요. 소유자 권한이 있는 사용자의 경우 Deadline Cloud Monitor는 사용량을 탐색하고 예산을 생성할 수 있는 액세스 권한도 제공합니다. 데드라인 클라우드 모니터는 웹 브라우저와 데스크톱 애플리케이션으로 모두 사용할 수 있습니다.

- AWS SDK 및 AWS CLI — AWS Command Line Interface (AWS CLI) 를 사용하여 로컬 시스템의 명령줄에서 Deadline Cloud API 작업을 호출할 수 있습니다. 자세한 내용은 개발자 워크스테이션 설정을 참조하십시오.

관련 서비스

데드라인 클라우드는 다음과 AWS 서비스함께 작동합니다.

- Amazon CloudWatch — 를 사용하면 프로젝트 및 관련 AWS 리소스를 모니터링할 수 있습니다. CloudWatch 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.
- Amazon EC2 — 클라우드에서 애플리케이션을 실행하는 가상 서버를 AWS 서비스 제공합니다. 워크로드에 Amazon EC2 인스턴스를 사용하도록 프로젝트를 구성할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스를 참조하십시오.
- Amazon EC2 Auto Scaling — Auto Scaling을 사용하면 인스턴스에 대한 수요 변화에 따라 인스턴스 수를 자동으로 늘리거나 줄일 수 있습니다. Auto Scaling을 사용하면 인스턴스에 장애가 발생하더라도 원하는 수의 인스턴스를 실행하고 있는지 확인할 수 있습니다. 데드라인 클라우드를 통한 Auto Scaling을 활성화하면 Auto Scaling에서 시작된 인스턴스가 자동으로 워크로드에 등록됩니다. 마찬가지로, Auto Scaling에 의해 종료된 인스턴스는 워크로드에서 자동으로 등록 취소됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하십시오.
- AWS PrivateLink— 트래픽을 퍼블릭 인터넷에 노출시키지 않고 가상 사설 클라우드 (VPC) 와 온프레미스 네트워크 간의 사설 연결을 AWS PrivateLink 제공합니다. AWS 서비스 AWS PrivateLink 다양한 계정과 VPC에서 서비스를 쉽게 연결할 수 있습니다. 자세한 정보는 AWS PrivateLink을 참조하세요.
- Amazon S3 — Amazon S3는 객체 스토리지 서비스입니다. 데드라인 클라우드는 Amazon S3 버킷을 사용하여 작업 첨부 파일을 저장합니다.
- IAM ID 센터 — IAM ID 센터는 사용자에게 지정된 모든 계정과 애플리케이션에 대한 싱글 사인온 액세스를 한 곳에서 제공할 수 AWS 서비스 있는 곳입니다. 또한 AWS Organizations에서 모든 계정에 대한 다중 계정 액세스 및 사용자 권한을 중앙에서 관리할 수 있습니다. 자세한 내용은 AWS IAM Identity Center FAQ를 참조하세요.

데드라인 클라우드 작동 방식

Deadline Cloud를 사용하면 디지털 콘텐츠 제작 (DCC) 파이프라인과 워크스테이션에서 직접 렌더링 프로젝트와 작업을 만들고 관리할 수 있습니다.

AWS SDK, AWS Command Line Interface (AWS CLI) 또는 데드라인 클라우드 작업 제출자를 사용하여 데드라인 클라우드에 작업을 제출합니다. 데드라인 클라우드는 작업 템플릿 사양에 대한 오픈 잡 디스크립션 (OpenJD) 을 지원합니다. 자세한 내용은 GitHub 웹 사이트의 Open Job Description을 참조하십시오.

데드라인 클라우드는 구직 제출자를 제공합니다. 작업 제출자는 타사 DCC 인터페이스 (예: 또는) 에서 렌더링 작업을 제출하기 위한 DCC 플러그인입니다. Maya Nuke 제출자를 통해 아티스트는 타사 인터페이스의 렌더링 작업을 Deadline Cloud로 제출할 수 있습니다. Deadline Cloud에서는 프로젝트 리소스를 관리하고 작업을 모두 한 곳에서 모니터링할 수 있습니다.

Deadline Cloud 팜을 사용하면 대기열과 플릿을 만들고, 사용자를 관리하고, 프로젝트 리소스 사용 및 비용을 관리할 수 있습니다. 팜은 대기열과 플릿으로 구성되어 있습니다. 큐는 제출된 작업을 찾고 렌더링을 예약하는 곳입니다. 플릿은 작업을 완료하기 위한 작업을 실행하는 작업자 노드 그룹입니다. 작업을 렌더링할 수 있으려면 대기열을 플릿과 연결해야 합니다. 단일 플릿은 여러 대기열을 지원할 수 있고 대기열은 여러 플릿에서 지원할 수 있습니다.

작업은 여러 단계로 구성되며 각 단계는 특정 작업으로 구성됩니다. Deadline Cloud 모니터를 사용하면 작업, 단계, 작업에 대한 상태, 로그 및 기타 문제 해결 지표에 액세스할 수 있습니다.

데드라인 클라우드의 권한

데드라인 클라우드는 다음을 지원합니다.

- AWS Identity and Access Management (IAM) 을 사용하여 API 작업에 대한 액세스 관리
- 와의 통합을 사용하여 인력 사용자의 액세스 관리 AWS IAM Identity Center

누구나 프로젝트 작업을 수행할 수 있으려면 먼저 해당 프로젝트 및 관련 팜에 대한 액세스 권한이 있어야 합니다. Deadline Cloud는 IAM Identity Center와 통합되어 직원 인증 및 권한 부여를 관리합니다. 사용자를 IAM Identity Center에 직접 추가하거나, 와 같이 기존 ID 공급자 (IdP) Okta 에 연결할 수 있습니다. Active Directory IT 관리자는 다양한 수준의 사용자 및 그룹에 액세스 권한을 부여할 수 있습니다. 각 후속 수준에는 이전 수준의 권한이 포함됩니다. 다음 목록은 가장 낮은 수준에서 가장 높은 수준까지의 네 가지 액세스 수준을 설명합니다.

- 뷰어 - 액세스할 수 있는 팜, 큐, 플릿, 작업의 리소스를 볼 수 있는 권한입니다. 뷰어는 작업을 제출하거나 변경할 수 없습니다.
- 기여자 - 뷰어와 동일하지만 대기열이나 팜에 작업을 제출할 수 있는 권한이 있습니다.
- 관리자 - 기여자와 동일하지만 액세스 권한이 있는 대기열의 작업을 편집하고 액세스 권한이 있는 리소스에 대한 권한을 부여할 수 있는 권한이 있습니다.

- 소유자 - 관리자와 동일하지만 예산을 조회 및 생성하고 사용량을 볼 수 있습니다.

> **Note**
>
> 이러한 권한은 사용자에게 Deadline Cloud 인프라에 대한 액세스 AWS Management Console 또는 수정 권한을 부여하지 않습니다.

사용자는 팜에 대한 액세스 권한이 있어야 관련 대기열 및 플릿에 액세스할 수 있습니다. 사용자 액세스 권한은 팜 내의 큐와 플릿에 개별적으로 할당됩니다.

사용자를 개인 또는 그룹의 일원으로 추가할 수 있습니다. 팜, 플릿 또는 큐에 그룹을 추가하면 대규모 사용자 그룹의 액세스 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 특정 프로젝트를 진행하는 팀이 있는 경우 각 팀원을 그룹에 추가할 수 있습니다. 그런 다음 해당 팜, 플릿 또는 큐의 전체 그룹에 액세스 권한을 부여할 수 있습니다.

데드라인 클라우드를 통한 소프트웨어 지원

Deadline Cloud는 명령줄 인터페이스에서 실행하고 매개변수 값을 사용하여 제어할 수 있는 모든 소프트웨어 애플리케이션과 호환됩니다. Deadline Cloud는 작업별로 매개 변수화된 소프트웨어 스크립트 단계 (예: 프레임 범위 전체) 를 사용하여 작업을 작업으로 설명하는 OpenJD 사양을 지원합니다. Deadline Cloud 도구 및 기능을 사용하여 OpenJD 작업 지침을 작업 번들로 취합하여 타사 소프트웨어 애플리케이션에서 단계를 만들고 실행하고 라이선스를 부여할 수 있습니다.

작업을 렌더링하려면 라이선스가 필요합니다. Deadline Cloud는 엄선된 소프트웨어 애플리케이션 라이선스에 대해 사용량 기반 라이선스 (UBL) 를 제공하며, 사용량에 따라 분 단위로 시간당 요금이 청구됩니다. Deadline Cloud를 사용하면 원하는 경우 자체 소프트웨어 라이선스를 사용할 수도 있습니다. 작업이 라이선스에 액세스할 수 없는 경우 렌더링되지 않고 Deadline Cloud 모니터의 작업 로그에 오류가 표시됩니다.

데드라인 클라우드 시작하기

AWS 데드라인 클라우드에서 팜을 만들려면 데드라인 클라우드 콘솔 또는 AWS Command Line Interface (AWS CLI) 를 사용할 수 있습니다. 콘솔을 사용하여 대기열 및 플릿을 포함한 팜 만들기 안내를 받아 보세요. AWS CLI 를 사용하여 서비스와 직접 작업하거나 Deadline Cloud와 호환되는 자체 도구를 개발할 수 있습니다.

팜을 만들고 데드라인 클라우드 모니터를 사용하려면 데드라인 클라우드용 계정을 설정하세요. 데드라인 클라우드 모니터 인프라는 계정당 한 번만 설정하면 됩니다. 팜에서 팜 및 해당 리소스에 대한 사용자 액세스를 포함하여 프로젝트를 관리할 수 있습니다.

Deadline Cloud 모니터 인프라를 설정하지 않고 팜을 만들려면 Deadline Cloud용 개발자 워크스테이션을 설정해야 합니다.

최소한의 리소스로 팜을 생성하여 작업을 수락하려면 콘솔 홈 페이지에서 Quickstart를 선택합니다. 데드라인 클라우드 모니터 설정해당 단계를 안내합니다. 이러한 팜은 자동으로 연결되는 대기열과 플릿으로 시작합니다. 이 방법을 사용하면 실험해 볼 수 있는 샌드박스 스타일의 농장을 만들 수 있는 편리한 방법입니다.

주제

- AWS 계정설정
- 데드라인 클라우드 모니터 설정
- 데드라인 클라우드용 개발자 워크스테이션 설정
- 데드라인 클라우드 제출자를 설정하세요.
- 팜을 사용하세요.

AWS 계정설정

AWS 데드라인 AWS 계정 클라우드를 사용하도록 설정하세요.

계정이 AWS 계정없는 경우 다음 단계를 완료하여 새로 만드세요.

가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

처음 계정을 AWS 계정만들 때는 계정의 모든 AWS 서비스 리소스와 모든 리소스에 완전히 액세스할 수 있는 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다.

> **Important**
>
> 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하십시오.

데드라인 클라우드 모니터 설정

시작하려면 Deadline Cloud 모니터 인프라를 만들고 팜을 정의해야 합니다. 그룹 및 사용자 추가, 서비스 역할 선택, 리소스에 태그 추가 등 선택적 추가 단계를 수행할 수도 있습니다.

1단계: 모니터 설정

데드라인 클라우드 모니터는 사용자를 인증하는 AWS IAM Identity Center 데 사용합니다. 데드라인 클라우드에 사용하는 IAM Identity Center 인스턴스는 AWS 리전 모니터와 동일한 위치에 있어야 합니다. 모니터를 생성할 때 콘솔에서 다른 지역을 사용하는 경우 IAM Identity Center 지역으로 변경하라는 알림이 표시됩니다.

모니터 인프라는 다음과 같은 구성 요소로 구성되어 있습니다.

- 모니터 디스플레이 이름: 모니터 디스플레이 이름은 모니터를 식별하는 방법 (예AnyCompany : 모니터) 입니다. 모니터 이름도 모니터 URL을 결정합니다.

> **Important**
>
> 설정을 완료한 후에는 모니터 표시 이름을 변경할 수 없습니다.

- 모니터 URL: 모니터 URL을 사용하여 모니터에 액세스할 수 있습니다. URL은 모니터 디스플레이 이름을 기반으로 합니다 (예: https://anycompanymonitor.awsapps.com).

> **Important**
>
> 설정을 완료한 후에는 모니터 URL을 변경할 수 없습니다.

- AWS 리전: AWS 데이터 센터 컬렉션의 실제 위치입니다. AWS 리전 모니터를 설정하면 기본적으로 가장 가까운 위치가 지역입니다. 지역을 변경하여 사용자와 가장 가까운 곳에 위치하도록 하는 것이 좋습니다. 이렇게 하면 지연이 줄어들고 데이터 전송 속도가 향상됩니다. AWS IAM Identity Center 데드라인 클라우드와 AWS 리전 동일하게 활성화해야 합니다.

> **Important**
>
> 데드라인 클라우드 설정을 완료한 후에는 지역을 변경할 수 없습니다.

이 섹션의 작업을 완료하여 모니터의 인프라를 구성하십시오.

모니터 인프라를 구성하려면

1. 에 AWS Management Console로그인하여 웰컴 투 데드라인 클라우드 설정을 시작한 후 다음을 선택합니다.
2. 모니터 디스플레이 이름 (예:) 을 입력합니다**`AnyCompany Monitor`**.
3. (선택 사항) 모니터 이름을 변경하려면 [URL 편집] 을 선택합니다.
4. (선택 사항) 사용자와 가장 가까운 위치로 변경하려면 지역 변경을 선택합니다. AWS 리전

   a. 사용자들과 가장 가까운 리전을 선택합니다.

   b. 리전 적용을 선택합니다.

   - (선택 사항) 그룹 및 사용자를 추가하려면 을 선택합니다 (선택 사항) 그룹 및 사용자 추가.
   - (선택 사항) 모니터 설정을 추가로 사용자 지정하려면 을 선택합니다 추가 설정.

5. 준비가 2단계: 팜 세부 정보 정의되면 [다음] 을 선택합니다.

(선택 사항) 그룹 및 사용자 추가

Deadline Cloud 모니터 설정을 완료하기 전에 모니터 사용자를 추가하고 그룹에 추가할 수 있습니다.

설정이 완료되면 새 사용자 및 그룹을 생성하고 그룹, 권한 및 애플리케이션을 할당하거나 모니터에서 사용자를 삭제하는 등 사용자를 관리할 수 있습니다.

추가 설정

데드라인 클라우드 설정에는 추가 설정이 포함됩니다. 이러한 설정을 사용하면 Deadline Cloud 설정에서 변경한 모든 내용을 확인하고 AWS 계정, 모니터 사용자 역할을 구성하고, 암호화 키 유형을 변경할 수 있습니다.

AWS IAM Identity Center

AWS IAM Identity Center 사용자 및 그룹을 관리하기 위한 클라우드 기반 싱글 사인온 서비스입니다. IAM Identity Center를 엔터프라이즈 Single Sign-On(SSO) 공급자와 통합하면 사용자가 회사 계정으로 로그인할 수도 있습니다.

데드라인 클라우드는 기본적으로 IAM Identity Center를 지원하며, 데드라인 클라우드를 설정하고 사용하려면 데드라인 클라우드가 필요합니다. 데드라인 클라우드에 사용하는 IAM ID 센터 인스턴스는 AWS 리전 모니터와 동일한 위치에 있어야 합니다. 자세한 내용은 AWS IAM Identity Center무엇입니까를 참조하십시오.

서비스 액세스 역할 구성

AWS 서비스는 사용자를 대신하여 작업을 수행하는 서비스 역할을 맡을 수 있습니다. Deadline Cloud에서 사용자에게 모니터의 리소스에 대한 액세스 권한을 부여하려면 모니터 사용자 역할이 필요합니다.

AWS Identity and Access Management (IAM) 관리형 정책을 모니터 사용자 역할에 연결할 수 있습니다. 정책을 통해 사용자는 특정 작업 (예: 특정 Deadline Cloud 애플리케이션에서 작업 생성) 을 수행할 수 있습니다. 애플리케이션은 관리형 정책의 특정 조건에 의존하므로 관리형 정책을 사용하지 않으면 애플리케이션이 예상대로 작동하지 않을 수 있습니다.

설정을 완료한 후 언제든지 모니터 사용자 역할을 변경할 수 있습니다. 사용자 역할에 대한 자세한 내용은 IAM 역할을 참조하세요.

다음 탭에는 두 가지 사용 사례에 대한 지침이 포함되어 있습니다. 새 서비스 역할을 생성하고 사용하려면 새 서비스 역할 탭을 선택합니다. 기존 서비스 역할을 사용하려면 기존 서비스 역할 탭을 선택합니다.

New service role

새 서비스 역할을 생성하고 사용하려면

1. 새 서비스 역할 생성 및 사용을 선택합니다.
2. (선택 사항) 서비스 사용자 역할 이름을 입력합니다.
3. 역할에 대한 자세한 내용을 보려면 권한 세부 정보 보기를 선택합니다.

Existing service role

기존 서비스 역할을 사용하려면

1. 기존 서비스 역할 사용을 선택합니다.
2. 드롭다운 목록을 열어 기존 서비스 역할을 선택합니다.
3. (선택 사항) 역할에 대한 자세한 내용을 보려면 IAM 콘솔에서 보기를 선택합니다.

2단계: 팜 세부 정보 정의

Deadline Cloud 콘솔로 돌아가서 다음 단계를 완료하여 팜 세부 정보를 정의합니다.

1. 팜 세부 정보에서 팜의 이름을 추가합니다.
2. 설명에 팜 설명을 입력합니다. 설명이 명확하면 팜의 용도를 빠르게 파악할 수 있습니다.
3. (선택 사항) 기본적으로 데이터는 보안을 위해 AWS 소유하고 관리하는 키로 암호화됩니다. 암호화 설정 사용자 지정 (고급) 을 선택하여 기존 키를 사용하거나 관리하는 새 키를 만들 수 있습니다.

   확인란을 사용하여 암호화 설정을 사용자 지정하려면 AWS KMS ARN을 입력하거나 새 KMS 키 생성을 선택하여 AWS KMS 새 ARN을 생성합니다.
4. (선택 사항) 새 태그 추가를 선택하여 팜에 하나 이상의 태그를 추가합니다.
5. 다음 옵션 중 하나를 선택하세요:

   - 검토 및 생성으로 건너뛰기를 선택하여 팜을 검토하고 생성하십시오.
   - 다음을 선택하여 추가 선택적 단계로 진행하십시오.

(선택 사항) 3단계: 대기열 세부 정보 정의

대기열은 작업 진행 상황을 추적하고 작업을 예약하는 역할을 합니다.

1. 대기열 세부 정보에서 시작하여 대기열의 이름을 입력합니다.
2. 설명에 대기열 설명을 입력합니다. 설명이 명확하면 대기열의 용도를 빠르게 파악할 수 있습니다.
3. 작업 첨부의 경우 새 Amazon S3 버킷을 만들거나 기존 Amazon S3 버킷을 선택할 수 있습니다. 기존 Amazon S3 버킷이 없는 경우 버킷을 생성해야 합니다.

   a. 새 Amazon S3 버킷을 생성하려면 새 작업 버킷 생성을 선택합니다. 루트 접두사 필드에 작업 버킷 이름을 정의할 수 있습니다. 버킷을 **`deadlinecloud-job-attachments-[MONITORNAME]`** 호출하는 것이 좋습니다.

      소문자와 대시만 사용할 수 있습니다. 공백이나 특수 문자는 사용할 수 없습니다.

   b. 기존 Amazon S3 버킷을 검색하고 선택하려면 기존 Amazon S3 버킷에서 선택을 선택합니다. 그런 다음 S3 찾아보기를 선택하여 기존 버킷을 검색합니다. 사용 가능한 Amazon S3 버킷 목록이 표시되면 대기열에 사용할 Amazon S3 버킷을 선택합니다.
4. 고객 관리형 플릿을 사용하는 경우 고객 관리 플릿과의 연결 활성화를 선택합니다.

   - 고객 관리형 플릿의 경우 대기열로 구성된 사용자를 추가한 다음 POSIX 및/또는 Windows 자격 증명을 설정합니다. 확인란을 선택하여 실행 기능을 우회할 수도 있습니다.
5. 대기열에는 사용자를 대신하여 Amazon S3에 액세스할 수 있는 권한이 필요합니다. 모든 대기열에 대해 새 서비스 역할을 생성하는 것이 좋습니다.

   a. 새 역할을 만들려면 다음 단계를 완료하세요.

      i. 새 서비스 역할 생성 및 사용을 선택합니다.

      ii. 큐 역할의 역할 이름을 입력하거나 제공된 역할 이름을 사용합니다.

      iii. (선택 사항) 큐 역할 설명을 추가합니다.

      iv. 권한 세부 정보 보기를 선택하여 대기열 역할에 대한 IAM 권한을 볼 수 있습니다.

   b. 또는 기존 서비스 역할을 선택할 수도 있습니다.
6. (선택 사항) 이름 및 값 쌍을 사용하여 큐 환경에 대한 환경 변수를 추가합니다.
7. (선택 사항) 키와 값 쌍을 사용하여 큐에 태그를 추가합니다.

대기열 세부 정보를 모두 입력한 후 다음을 선택합니다.

(선택 사항) 4단계: 플릿 세부 정보 정의

플릿은 렌더링 작업을 실행할 작업자를 할당합니다. 렌더링 작업에 플릿이 필요한 경우 플릿 생성 체크박스를 선택합니다.

1. 플릿 세부 정보

    a. 플릿의 이름과 설명 (선택 사항) 을 모두 입력하십시오.
    b. 컴퓨팅 리소스를 확장해야 하는 방식을 선택합니다. 서비스 관리 옵션을 사용하면 Deadline Cloud에서 컴퓨팅 리소스를 자동으로 확장할 수 있습니다. 고객 관리형 옵션을 사용하면 컴퓨팅 스케일링을 직접 제어할 수 있습니다.
2. 인스턴스 옵션 섹션에서 스팟 또는 온디맨드를 선택합니다. Amazon EC2 온디맨드 인스턴스는 더 빠른 가용성을 제공하며 Amazon EC2 스팟 인스턴스는 비용 절감 노력에 더 적합합니다.
3. 플릿의 인스턴스 수를 Auto Scaling하려면 최소 인스턴스 수와 최대 인스턴스 수를 모두 선택하십시오.

    추가 비용이 발생하지 않도록 항상 최소 인스턴스 수를 설정하는 **0** 것이 좋습니다.
4. 플릿에는 사용자를 CloudWatch 대신하여 글을 쓸 수 있는 권한이 필요합니다. 모든 플릿에 대해 새 서비스 역할을 생성하는 것이 좋습니다.

    a. 새 역할을 만들려면 다음 단계를 완료하세요.

        i. 새 서비스 역할 생성 및 사용을 선택합니다.
        ii. 플릿 역할의 역할 이름을 입력하거나 제공된 역할 이름을 사용하십시오.
        iii. (선택 사항) 플릿 역할 설명을 추가합니다.
        iv. 권한 세부 정보 보기를 선택하여 플릿 역할에 대한 IAM 권한을 볼 수 있습니다.
    b. 또는 기존 서비스 역할을 사용할 수 있습니다.
5. (선택 사항) 키와 값 쌍을 사용하여 플릿에 태그를 추가합니다.

플릿 세부 정보를 모두 입력한 후 다음을 선택합니다.

(선택 사항) 5단계: 작업자 요구 사항 구성

작업자 인스턴스의 요구 사항을 정의하십시오.

1. 운영 체제 (OS) 및 CPU 아키텍처 설정의 인식을 검토하십시오.
2. 하드웨어 요구 사항에 맞게 vCPU의 최소 및 최대 수를 업데이트하십시오.

3. 하드웨어 요구 사항에 맞게 최소 및 최대 메모리 수 (GiB) 를 업데이트하십시오.
4. 작업자 인스턴스 유형을 허용하거나 제외하여 인스턴스 유형을 필터링할 수 있습니다. 두 필터링 옵션 모두에서 최대 10개의 Amazon EC2 인스턴스 유형을 필터링할 수 있습니다.
5. 추가 요구 사항 (선택 사항) 에서 크기 (GiB), IOPS 및 처리량 (MiB/s) 별로 루트 EBS 볼륨을 정의할 수 있습니다.
6. 작업자 요구 사항을 모두 설정한 후 [Next] 를 선택하여 그룹의 액세스 수준을 정의합니다.

(선택 사항) 6단계: 액세스 수준 정의

모니터에 연결된 그룹이 있는 경우 그룹의 액세스 수준을 정의할 수 있습니다. Deadline Cloud 기능 사용 권한은 액세스 수준에 따라 관리됩니다. 사용자 그룹에 다양한 액세스 수준을 할당할 수 있습니다.

1. Deadline Cloud 팜 액세스 수준 메뉴를 사용하여 그룹에 대한 권한 수준을 선택합니다.
2. 계속하고 입력된 모든 팜 세부 정보를 검토하려면 [다음] 을 선택합니다.

7단계: 검토 및 생성

입력한 모든 정보를 검토하여 팜을 만드세요. 준비가 되면 [Create farm] 을 선택합니다.

팜 생성 진행 상황이 팜 페이지에 표시됩니다. 팜을 사용할 준비가 되면 성공 메시지가 표시됩니다.

데드라인 클라우드용 개발자 워크스테이션 설정

이 자습서에서는 간단한 개발자 AWS CloudShell 팜을 만들고 작업자 에이전트를 실행하는 데 사용합니다. 그런 다음 매개 변수와 첨부 파일이 포함된 간단한 작업을 제출 및 실행하고, 서비스 관리 플릿을 추가하고, 완료되면 팜 리소스를 정리할 수 있습니다.

다음 섹션에서는 Deadline Cloud의 다양한 기능과 이들이 어떻게 작동하고 함께 작동하는지 소개합니다. 다음 단계를 따르면 새로운 워크로드 및 사용자 지정을 개발하고 테스트하는 데 유용합니다.

주제

- 1단계: 데드라인 클라우드 팜 만들기
- 2단계: Deadline Cloud의 개발자 모드에서 작업자 에이전트 실행
- 3단계: Deadline Cloud를 사용하여 작업 제출 및 실행
- 4단계: 데드라인 클라우드에서 작업 첨부 파일이 있는 작업 실행
- 5단계: Deadline Cloud의 개발자 팜에 서비스 관리 플릿 추가

- 6단계: 데드라인 클라우드에서 팜 리소스 정리

1단계: 데드라인 클라우드 팜 만들기

AWS Deadline Cloud에서 개발자 팜을 만들고 리소스를 대기열에 추가하려면 다음 절차에 표시된 대로 AWS Command Line Interface (AWS CLI) 를 사용합니다. 또한 AWS Identity and Access Management (IAM) 역할과 고객 관리형 플릿 (CMF) 을 생성하고 플릿을 대기열에 연결합니다. 그런 다음 를 구성하고 팜이 AWS CLI 지정된 대로 설정되고 작동하는지 확인할 수 있습니다.

이 팜을 사용하여 Deadline Cloud의 기능을 탐색한 다음 새 워크로드, 사용자 지정 및 파이프라인 통합을 개발하고 테스트할 수 있습니다.

팜을 만들려면

1. 아직 설치하지 않았다면 AWS Command Line Interface (AWS CLI) 를 설치하고 구성하십시오. 자세한 내용은 의 최신 버전 설치 또는 업데이트를 참조하십시오 AWS CLI.
2. 팜 이름을 만들고 해당 팜 이름을 에 추가합니다`~/.bashrc`. 이렇게 하면 다른 터미널 세션에서도 사용할 수 있게 됩니다.

```
echo "DEV_FARM_NAME=DeveloperFarm" >> ~/.bashrc
source ~/.bashrc
```

3. 팜 리소스를 만들고 해당 팜 ID를 추가합니다`~/.bashrc`.

```
aws deadline create-farm \
    --display-name "$DEV_FARM_NAME"

echo "DEV_FARM_ID=\$(aws deadline list-farms \
        --query \"farms[?displayName=='\$DEV_FARM_NAME'].farmId \
        | [0]\" --output text)" >> ~/.bashrc
source ~/.bashrc
```

4. 큐 리소스를 만들고 큐 ID를 다음에 추가합니다. `~/.bashrc.`

```
aws deadline create-queue \
    --farm-id $DEV_FARM_ID \
    --display-name "$DEV_FARM_NAME Queue" \
    --job-run-as-user '{"posix": {"user": "job-user", "group": "job-group"},
 "runAs":"QUEUE_CONFIGURED_USER"}'
```

```
echo "DEV_QUEUE_ID=\$(aws deadline list-queues \
        --farm-id \$DEV_FARM_ID \
        --query \"queues[?displayName=='\$DEV_FARM_NAME Queue'].queueId \
        | [0]\" --output text)" >> ~/.bashrc
source ~/.bashrc
```

5. 플릿에 대한 IAM 역할을 생성합니다. 이 역할은 플릿의 작업자 호스트에게 대기열에서 작업을 실행하는 데 필요한 보안 자격 증명을 제공합니다.

```
aws iam create-role \
    --role-name "${DEV_FARM_NAME}FleetRole" \
    --assume-role-policy-document \
        '{
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "credentials.deadline.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }'
aws iam put-role-policy \
    --role-name "${DEV_FARM_NAME}FleetRole" \
    --policy-name WorkerPermissions \
    --policy-document \
        '{
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": [
                        "deadline:AssumeFleetRoleForWorker",
                        "deadline:UpdateWorker",
                        "deadline:DeleteWorker",
                        "deadline:UpdateWorkerSchedule",
                        "deadline:BatchGetJobEntity",
                        "deadline:AssumeQueueRoleForWorker"
                    ],
                    "Resource": "*",
                    "Condition": {
```

```
                        "StringEquals": {
                            "aws:PrincipalAccount": "${aws:ResourceAccount}"
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "logs:CreateLogStream"
                    ],
                    "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
                    "Condition": {
                        "StringEquals": {
                            "aws:PrincipalAccount": "${aws:ResourceAccount}"
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "logs:PutLogEvents",
                        "logs:GetLogEvents"
                    ],
                    "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
                    "Condition": {
                        "StringEquals": {
                            "aws:PrincipalAccount": "${aws:ResourceAccount}"
                        }
                    }
                }
            ]
        }'
```

6. 고객 관리형 플릿 (CMF) 을 생성하고 해당 플릿 ID를 추가합니다~/.bashrc.

```
FLEET_ROLE_ARN="arn:aws:iam::$(aws sts get-caller-identity \
        --query "Account" --output text):role/${DEV_FARM_NAME}FleetRole"
aws deadline create-fleet \
    --farm-id $DEV_FARM_ID \
    --display-name "$DEV_FARM_NAME CMF" \
    --role-arn $FLEET_ROLE_ARN \
    --max-worker-count 5 \
    --configuration \
        '{
```

```
            "customerManaged": {
                "mode": "NO_SCALING",
                "workerCapabilities": {
                    "vCpuCount": {"min": 1},
                    "memoryMiB": {"min": 512},
                    "osFamily": "linux",
                    "cpuArchitectureType": "x86_64"
                }
            }
        }'

echo "DEV_CMF_ID=\$(aws deadline list-fleets \
        --farm-id \$DEV_FARM_ID \
        --query \"fleets[?displayName=='\$DEV_FARM_NAME CMF'].fleetId \
        | [0]\" --output text)" >> ~/.bashrc
source ~/.bashrc
```

7. 데드라인 클라우드에 액세스할 수 있는지 확인하세요.

```
pip install deadline
```

8. CMF를 대기열에 연결하세요.

```
aws deadline create-queue-fleet-association \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --fleet-id $DEV_CMF_ID
```

9. 기본 팜을 팜 ID로 설정하고 큐를 이전에 만든 큐 ID로 설정하려면 다음 명령을 사용합니다.

```
deadline config set defaults.farm_id $DEV_FARM_ID
deadline config set defaults.queue_id $DEV_QUEUE_ID
```

10. (선택 사항) 팜이 사양에 맞게 설정되었는지 확인하려면 다음 명령을 사용하십시오.

    - 모든 팜 목록 작성 — **`deadline farm list`**
    - 기본 팜의 모든 대기열 나열 — **`deadline queue list`**
    - 기본 팜의 모든 플릿 목록 나열 — **`deadline fleet list`**
    - 기본 팜 가져오기 — **`deadline farm get`**
    - 기본 대기열 가져오기 — **`deadline queue get`**
    - 기본 대기열과 연결된 모든 플릿 가져오기 — **`deadline fleet get`**

2단계: Deadline Cloud의 개발자 모드에서 작업자 에이전트 실행

개발자 팜의 대기열에 제출한 작업을 실행하려면 먼저 작업자 호스트의 개발자 모드에서 AWS Deadline Cloud 작업자 에이전트를 실행해야 합니다.

이 자습서의 나머지 부분에서는 두 개의 AWS CloudShell 탭을 사용하여 개발자 팜에서 AWS CLI 작업을 수행해 보겠습니다. 첫 번째 탭에서는 작업을 제출할 수 있습니다. 두 번째 탭에서는 작업자 에이전트를 실행할 수 있습니다.

> **Note**
>
> CloudShell 세션을 20분 이상 유휴 상태로 두면 제한 시간이 초과되어 작업자 에이전트가 중지됩니다. 작업자 에이전트를 다시 시작하려면 다음 절차의 지침을 따르십시오.

개발자 모드에서 작업자 에이전트를 실행하려면

1. 아직 설치하지 않았다면 AWS Command Line Interface (AWS CLI) 를 설치하고 구성하십시오. 자세한 내용은 의 최신 버전 설치 또는 업데이트를 참조하십시오 AWS CLI.
2. 첫 번째 CloudShell 탭에 팜이 열려 있는 상태에서 두 번째 CloudShell 탭을 연 다음 `demoenv-logs` 및 `demoenv-persist` 디렉터리를 생성하십시오.

   ```
   mkdir ~/demoenv-logs
   mkdir ~/demoenv-persist
   ```

3. PyPI에서 데드라인 클라우드 워커 에이전트 패키지를 다운로드하여 설치합니다.

   > **Note**
   >
   > 이제 에이전트 파일을 Python의 글로벌 사이트 패키지 디렉터리에 설치해야 합니다. Windows Python 가상 환경은 현재 지원되지 않습니다.

   ```
   python -m pip install deadline-cloud-worker-agent
   ```

4. 작업자 에이전트가 작업 실행을 위한 임시 디렉터리를 생성할 수 있도록 하려면 디렉터리를 생성하십시오.

   ```
   sudo mkdir /sessions
   ```

```
sudo chmod 750 /sessions
sudo chown cloudshell-user /sessions
```

5. 에 추가한 변수를 DEV_FARM_ID 사용하여 Deadline Cloud 작업자 에이전트를 개발자 모드에서 실행합니다. DEV_CMF_ID ~/.bashrc

```
deadline-worker-agent \
    --farm-id $DEV_FARM_ID \
    --fleet-id $DEV_CMF_ID \
    --run-jobs-as-agent-user \
    --logs-dir ~/demoenv-logs \
    --persistence-dir ~/demoenv-persist
```

작업자 에이전트가 UpdateWorkerSchedule API 작업을 초기화하고 폴링하면 다음과 같은 출력이 표시됩니다.

```
INFO    Worker Agent starting
[2024-03-27 15:51:01,292][INFO    ] # Worker Agent starting
[2024-03-27 15:51:01,292][INFO    ] AgentInfo
Python Interpreter: /usr/bin/python3
Python Version: 3.9.16 (main, Sep  8 2023, 00:00:00)  - [GCC 11.4.1 20230605 (Red
 Hat 11.4.1-2)]
Platform: linux
...
[2024-03-27 15:51:02,528][INFO    ] # API.Resp # [deadline:UpdateWorkerSchedule]
(200) params={'assignedSessions': {}, 'cancelSessionActions': {},
 'updateIntervalSeconds': 15} ...
[2024-03-27 15:51:17,635][INFO    ] # API.Resp # [deadline:UpdateWorkerSchedule]
(200) params=(Duplicate removed, see previous response) ...
[2024-03-27 15:51:32,756][INFO    ] # API.Resp # [deadline:UpdateWorkerSchedule]
(200) params=(Duplicate removed, see previous response) ...
...
```

6. 첫 번째 CloudShell 탭을 선택한 다음 플릿의 작업자를 나열하십시오.

```
deadline worker list --fleet-id $DEV_CMF_ID
```

다음과 같은 출력이 표시됩니다.

```
Displaying 1 of 1 workers starting at 0
```

```
- workerId: worker-8c9af877c8734e89914047111f
  status: STARTED
  createdAt: 2023-12-13 20:43:06+00:00
```

프로덕션 구성에서 Deadline Cloud 작업자 에이전트는 호스트 컴퓨터의 관리자 권한으로 여러 사용자 및 구성 디렉터리를 설정해야 합니다. 사용자만 액세스할 수 있는 자체 개발 팜에서 작업을 실행하므로 이러한 설정을 재정의할 수 있습니다.

3단계: Deadline Cloud를 사용하여 작업 제출 및 실행

AWS Deadline Cloud를 사용하여 작업을 실행하려면 다음 절차를 사용하십시오. 첫 번째 AWS CloudShell 탭을 사용하여 개발자 팜에 작업을 제출하십시오. 두 번째 CloudShell 탭에서는 작업자 에이전트 결과를 볼 수 있습니다.

주제

- simple_job샘플 제출
- 매개변수와 simple_job 함께 제출하십시오.
- 파일 I/O가 포함된 simple_file_job 작업 번들을 생성하십시오.

simple_job샘플 제출

팜을 만들고 작업자 에이전트를 실행한 후 Deadline Cloud에 simple_job 샘플을 제출할 수 있습니다.

데드라인 클라우드에 simple_job 샘플을 제출하려면

1. 아직 설치하지 않았다면 AWS Command Line Interface (AWS CLI) 를 설치하고 구성하세요. 자세한 내용은 의 최신 버전 설치 또는 업데이트를 참조하십시오 AWS CLI.
2. 에서 샘플을 다운로드하십시오 GitHub.

```
cd ~
git clone https://github.com/aws-deadline/deadline-cloud-samples.git
```

3. 첫 번째 CloudShell 탭을 선택한 다음 작업 번들 샘플 디렉토리로 이동합니다.

```
cd ~/deadline-cloud-samples/job_bundles/
```

4. simple_job샘플을 제출하십시오.

```
deadline bundle submit simple_job
```

5. 두 번째 CloudShell 탭을 선택하면 통화BatchGetJobEntities, 세션 가져오기, 세션 작업 실행에 대한 로깅 출력을 볼 수 있습니다.

```
...
[2024-03-27 16:00:21,846][INFO    ] # Session.Starting
 # [session-053d77cef82648fe2] Starting new Session.
 [queue-3ba4ff683ff54db09b851a2ed8327d7b/job-d34cc98a6e234b6f82577940ab4f76c6]
[2024-03-27 16:00:21,853][INFO    ] # API.Req # [deadline:BatchGetJobEntity]
 resource={'farm-id': 'farm-3e24cfc9bbcd423e9c1b6754bc1',
 'fleet-id': 'fleet-246ee60f46d44559b6cce010d05', 'worker-id':
 'worker-75e0fce9c3c344a69bff57fcd83'} params={'identifiers': [{'jobDetails':
 {'jobId': 'job-d34cc98a6e234b6f82577940ab4'}}]} request_url=https://
scheduling.deadline.us-west-2.amazonaws.com/2023-10-12/farms/
farm-3e24cfc9bbcd423e /fleets/fleet-246ee60f46d44559b1 /workers/worker-
 75e0fce9c3c344a69b /batchGetJobEntity
[2024-03-27 16:00:22,013][INFO    ] # API.Resp # [deadline:BatchGetJobEntity](200)
 params={'entities': [{'jobDetails': {'jobId': 'job-d34cc98a6e234b6f82577940ab6',
 'jobRunAsUser': {'posix': {'user': 'job-user', 'group': 'job-group'},
 'runAs': 'QUEUE_CONFIGURED_USER'}, 'logGroupName': '/aws/deadline/
farm-3e24cfc9bbcd423e9c1b6754bc1/queue-3ba4ff683ff54db09b851a2ed83', 'parameters':
 '*REDACTED*', 'schemaVersion': 'jobtemplate-2023-09'}}], 'errors': []}
 request_id=a3f55914-6470-439e-89e5-313f0c6
[2024-03-27 16:00:22,013][INFO    ] # Session.Add #
 [session-053d77cef82648fea9c69827182] Appended new SessionActions.
 (ActionIds: ['sessionaction-053d77cef82648fea9c69827182-0'])
 [queue-3ba4ff683ff54db09b851a2ed8b/job-d34cc98a6e234b6f82577940ab6]
[2024-03-27 16:00:22,014][WARNING ] # Session.User #
 [session-053d77cef82648fea9c69827182] Running as the Worker Agent's
 user. (User: cloudshell-user) [queue-3ba4ff683ff54db09b851a2ed8b/job-
d34cc98a6e234b6f82577940ac6]
[2024-03-27 16:00:22,015][WARNING ] # Session.AWSCreds #
 [session-053d77cef82648fea9c69827182] AWS Credentials are not available: Queue has
 no IAM Role. [queue-3ba4ff683ff54db09b851a2ed8b/job-d34cc98a6e234b6f82577940ab6]
[2024-03-27 16:00:22,026][INFO    ] # Session.Logs #
 [session-053d77cef82648fea9c69827182] Logs streamed to: AWS CloudWatch
 Logs. (LogDestination: /aws/deadline/farm-3e24cfc9bbcd423e9c1b6754bc1/
queue-3ba4ff683ff54db09b851a2ed83/session-053d77cef82648fea9c69827181)
 [queue-3ba4ff683ff54db09b851a2ed83/job-d34cc98a6e234b6f82577940ab4]
[2024-03-27 16:00:22,026][INFO    ] # Session.Logs #
 [session-053d77cef82648fea9c69827182] Logs streamed to: local
```

```
 file. (LogDestination: /home/cloudshell-user/demoenv-logs/
queue-3ba4ff683ff54db09b851a2ed8b/session-053d77cef82648fea9c69827182.log)
 [queue-3ba4ff683ff54db09b851a2ed83/job-d34cc98a6e234b6f82577940ab4]
...
```

> **Note**
>
> 작업자 에이전트의 로깅 출력만 표시됩니다. 작업을 실행하는 세션에는 별도의 로그가 있습니다.

6. 첫 번째 탭을 선택한 다음 작업자 에이전트가 작성하는 로그 파일을 살펴보세요.

   a. 작업자 에이전트 로그 디렉터리로 이동하여 해당 내용을 확인합니다.

   ```
   cd ~/demoenv-logs
   ls
   ```

   b. 작업자 에이전트가 만든 첫 번째 로그 파일을 인쇄합니다.

   ```
   cat worker-agent-bootstrap.log
   ```

   이 파일에는 Deadline Cloud API를 호출하여 플릿에 작업자 리소스를 생성한 다음 플릿 역할을 맡은 방법에 대한 작업자 에이전트 출력이 포함되어 있습니다.

   c. 워커 에이전트가 플릿에 가입할 때 로그 파일 출력을 인쇄합니다.

   ```
   cat worker-agent.log
   ```

   이 로그에는 작업자 에이전트가 수행하는 모든 작업에 대한 출력이 포함되지만 해당 리소스의 ID를 제외하고 작업을 실행하는 대기열에 대한 출력은 포함되지 않습니다.

   d. 큐 리소스 ID와 이름이 같은 디렉터리에 있는 각 세션의 로그 파일을 인쇄합니다.

   ```
   cat $DEV_QUEUE_ID/session-*.log
   ```

   작업이 성공하면 로그 파일 출력은 다음과 비슷합니다.

   ```
    cat $DEV_QUEUE_ID/$(ls -t $DEV_QUEUE_ID | head -1)
   2024-03-27 16:00:22,026 WARNING Session running with no AWS Credentials.
   2024-03-27 16:00:22,404 INFO
   ```

```
2024-03-27 16:00:22,405 INFO ================================================
2024-03-27 16:00:22,405 INFO --------- Running Task
2024-03-27 16:00:22,405 INFO ================================================
2024-03-27 16:00:22,406 INFO ------------------------------------------------
2024-03-27 16:00:22,406 INFO Phase: Setup
2024-03-27 16:00:22,406 INFO ------------------------------------------------
2024-03-27 16:00:22,406 INFO Writing embedded files for Task to disk.
2024-03-27 16:00:22,406 INFO Mapping: Task.File.runScript -> /sessions/
session-053d77cef82648fea9c698271812a/embedded_fileswa_gj55_/tmp2u9yqtsz
2024-03-27 16:00:22,406 INFO Wrote: runScript -> /sessions/
session-053d77cef82648fea9c698271812a/embedded_fileswa_gj55_/tmp2u9yqtsz
2024-03-27 16:00:22,407 INFO ------------------------------------------------
2024-03-27 16:00:22,407 INFO Phase: Running action
2024-03-27 16:00:22,407 INFO ------------------------------------------------
2024-03-27 16:00:22,407 INFO Running command /sessions/
session-053d77cef82648fea9c698271812a/tmpzuzxpslm.sh
2024-03-27 16:00:22,414 INFO Command started as pid: 471
2024-03-27 16:00:22,415 INFO Output:
2024-03-27 16:00:22,420 INFO Welcome to AWS Deadline Cloud!
2024-03-27 16:00:22,571 INFO
2024-03-27 16:00:22,572 INFO ================================================
2024-03-27 16:00:22,572 INFO --------- Session Cleanup
2024-03-27 16:00:22,572 INFO ================================================
2024-03-27 16:00:22,572 INFO Deleting working directory: /sessions/
session-053d77cef82648fea9c698271812a
```

7. 작업에 대한 정보를 인쇄합니다.

```
deadline job get
```

작업을 제출하면 시스템에서 해당 작업을 기본값으로 저장하므로 작업 ID를 입력할 필요가 없습니다.

매개변수와 simple_job 함께 제출하십시오.

매개 변수를 사용하여 작업을 제출할 수 있습니다. 다음 절차에서는 사용자 지정 메시지를 포함하도록 simple_job 템플릿을 편집하고 제출한 다음 세션 로그 파일을 인쇄하여 메시지를 확인합니다. simple_job

매개 변수와 함께 simple_job 샘플을 제출하려면

1. 첫 번째 CloudShell 탭을 선택한 다음 작업 번들 샘플 디렉토리로 이동합니다.

```
cd ~/deadline-cloud-samples/job_bundles/
```

2. simple_job템플릿의 내용을 인쇄합니다.

```
cat simple_job/template.yaml
```

Message매개변수가 있는 parameterDefinitions 섹션은 다음과 같아야 합니다.

```
parameterDefinitions:
- name: Message
  type: STRING
  default: Welcome to AWS Deadline Cloud!
```

3. 매개 변수 값과 함께 simple_job 샘플을 제출한 다음 작업 실행이 완료될 때까지 기다립니다.

```
deadline bundle submit simple_job \
    -p "Message=Greetings from the developer getting started guide."
```

4. 사용자 지정 메시지를 보려면 가장 최근의 세션 로그 파일을 확인하십시오.

```
cd ~/demoenv-logs
cat $DEV_QUEUE_ID/$(ls -t $DEV_QUEUE_ID | head -1)
```

파일 I/O가 포함된 simple_file_job 작업 번들을 생성하십시오.

렌더 작업은 장면 정의를 읽고 여기에서 이미지를 렌더링한 다음 해당 이미지를 출력 파일에 저장해야 합니다. 이미지를 렌더링하는 대신 입력의 해시를 계산하도록 하여 이 동작을 시뮬레이션할 수 있습니다.

파일 I/O가 포함된 simple_file_job 작업 번들을 만들려면

1. 첫 번째 CloudShell 탭을 선택한 다음 작업 번들 샘플 디렉토리로 이동합니다.

```
cd ~/deadline-cloud-samples/job_bundles/
```

2. 새 simple_job 이름으로 사본을 만드십시오simple_file_job.

```
cp -r simple_job simple_file_job
```

3. 다음과 같이 작업 템플릿을 편집합니다.

> **Note**
>
> 이 단계에는 nano 사용하는 것이 좋습니다. 사용하려면 를 사용하여 Vim 붙여넣기 모드를 설정해야 합니다:`set paste`.

a. 텍스트 편집기에서 템플릿을 엽니다.

```
nano simple_file_job/template.yaml
```

b. 다음 `typeobjectType`, 및 을 추가합니다 `dataFlowparameterDefinitions`.

```
- name: InFile
  type: PATH
  objectType: FILE
  dataFlow: IN
- name: OutFile
  type: PATH
  objectType: FILE
  dataFlow: OUT
```

c. 입력 파일에서 읽고 출력 파일에 쓰는 다음 bash 스크립트 명령을 파일 끝에 추가합니다.

```
        # hash the input file, and write that to the output
        sha256sum "{{Param.InFile}}" > "{{Param.OutFile}}"
```

업데이트 내용은 다음과 정확히 `template.yaml` 일치해야 합니다.

```
specificationVersion: 'jobtemplate-2023-09'
name: Simple File Job Bundle Example
parameterDefinitions:
- name: Message
  type: STRING
  default: Welcome to AWS Deadline Cloud!
- name: InFile
  type: PATH
  objectType: FILE
  dataFlow: IN
- name: OutFile
```

```
  type: PATH
  objectType: FILE
  dataFlow: OUT
steps:
- name: WelcomeToDeadlineCloud
  script:
    actions:
      onRun:
        command: '{{Task.File.runScript}}'
    embeddedFiles:
    - name: runScript
      type: TEXT
      runnable: true
      data: |
        #!/usr/bin/env bash
        echo "{{Param.Message}}"

        # hash the input file, and write that to the output
        sha256sum "{{Param.InFile}}" > "{{Param.OutFile}}"
```

> **Note**
>
> 에서 간격을 조정하려면 들여쓰기 대신 공백을 사용해야 합니다. `template.yaml`

d. 파일을 저장하고 텍스트 편집기를 종료합니다.

4. simple_file_job을 제출하기 위한 입력 및 출력 파일의 매개 변수 값을 제공하십시오.

```
deadline bundle submit simple_file_job \
    -p "InFile=simple_job/template.yaml" \
    -p "OutFile=hash.txt"
```

5. 작업에 대한 정보를 인쇄합니다.

```
deadline job get
```

- 다음과 같은 출력이 표시됩니다.

```
parameters:
  Message:
    string: Welcome to AWS Deadline Cloud!
  InFile:
```

```
    path: /local/home/cloudshell-user/BundleFiles/JobBundle-Examples/simple_job/
template.yaml
  OutFile:
    path: /local/home/cloudshell-user/BundleFiles/JobBundle-Examples/hash.txt
```

- 상대 경로만 제공했지만 매개변수에는 전체 경로가 설정되어 있습니다. 는 AWS CLI 경로에 해당 유형이 PATH 있을 때 현재 작업 디렉토리를 매개 변수로 제공된 모든 경로에 연결합니다.
- 다른 터미널 창에서 실행 중인 작업자 에이전트가 작업을 선택하여 실행합니다. 이 작업을 수행하면 다음 명령으로 볼 수 있는 `hash.txt` 파일이 생성됩니다.

  ```
  cat hash.txt
  ```

  이 명령은 다음과 비슷한 출력을 인쇄합니다.

  ```
  eaa2df5d34b54be5ac34c56a24a8c237b8487231a607eaf530a04d76b89c9cd3 /local/home/
  cloudshell-user/BundleFiles/JobBundle-Examples/simple_job/template.yaml
  ```

4단계: 데드라인 클라우드에서 작업 첨부 파일이 있는 작업 실행

많은 팜은 공유 파일 시스템을 사용하여 작업을 제출하는 호스트와 작업을 실행하는 호스트 간에 파일을 공유합니다. 예를 들어 이전 `simple_file_job` 예제에서 로컬 파일 시스템은 터미널 창 간에 공유되며 AWS CloudShell 터미널 창은 작업을 제출하는 탭 1에서 실행되고 작업자 에이전트를 실행하는 탭 2에서 실행됩니다.

제출자 워크스테이션과 작업자 호스트가 동일한 근거리 통신망에 있는 경우에는 공유 파일 시스템이 유리합니다. 데이터에 액세스하는 워크스테이션 근처의 온프레미스에 데이터를 저장하는 경우 클라우드 기반 팜을 사용하면 지연 시간이 긴 VPN을 통해 파일 시스템을 공유하거나 클라우드에서 파일 시스템을 동기화해야 합니다. 이 두 옵션 모두 설정하거나 운영하기가 쉽지 않습니다.

AWS Deadline Cloud는 이메일 첨부 파일과 유사한 작업 첨부 파일이 포함된 간단한 솔루션을 제공합니다. 작업 첨부 파일을 사용하면 작업에 데이터를 첨부할 수 있습니다. 그러면 데드라인 클라우드가 Amazon Simple Storage Service (Amazon S3) 버킷에 작업 데이터를 전송하고 저장하는 세부 정보를 처리합니다.

콘텐츠 생성 워크플로는 종종 반복적입니다. 즉, 사용자가 수정된 파일의 작은 하위 집합을 사용하여 작업을 제출합니다. Amazon S3 버킷은 콘텐츠 주소 지정이 가능한 스토리지에 작업 첨부 파일을 저장하기 때문에 각 객체의 이름은 객체 데이터의 해시를 기반으로 하며 디렉터리 트리의 콘텐츠는 작업에 연결된 매니페스트 파일 형식으로 저장됩니다.

작업 첨부 파일이 있는 작업을 실행하려면 다음 단계를 완료하십시오.

주제

- 대기열에 작업 첨부 파일 구성을 추가합니다.
- 작업 첨부 파일과 simple_file_job 함께 제출
- Amazon S3에 작업 첨부 파일이 저장되는 방식 이해

대기열에 작업 첨부 파일 구성을 추가합니다.

대기열에서 작업 첨부 파일을 활성화하려면 계정의 대기열 리소스에 작업 첨부 파일 구성을 추가하십시오.

대기열에 작업 첨부 구성을 추가하려면

1. 아직 설치하지 않았다면 AWS Command Line Interface (AWS CLI) 를 설치하고 구성하십시오. 자세한 내용은 의 최신 버전 설치 또는 업데이트를 참조하십시오 AWS CLI.
2. 첫 번째 CloudShell 탭을 선택한 다음 다음 명령 중 하나를 입력하여 Amazon S3 버킷을 작업 첨부에 사용합니다.
   - 기존 프라이빗 Amazon S3 버킷이 없는 경우 새 S3 버킷을 생성하여 사용할 수 있습니다.

     ```
     DEV_FARM_BUCKET=$(echo $DEV_FARM_NAME \
         | tr '[:upper:]' '[:lower:]')-$(xxd -l 16 -p /dev/urandom)
     if [ "$AWS_REGION" == "us-east-1" ]; then LOCATION_CONSTRAINT=
     else LOCATION_CONSTRAINT="--create-bucket-configuration \
         LocationConstraint=${AWS_REGION}"
     fi
     aws s3api create-bucket \
         $LOCATION_CONSTRAINT \
         --acl private \
         --bucket ${DEV_FARM_BUCKET}
     ```

   - 이미 프라이빗 Amazon S3 버킷이 *MY_BUCKET_NAME*있는 경우 해당 버킷의 이름으로 대체하여 사용할 수 있습니다.

     ```
     DEV_FARM_BUCKET=MY_BUCKET_NAME
     ```

3. Amazon S3 버킷을 만들거나 선택한 후, 버킷 이름을 에 `~/.bashrc` 추가하여 버킷을 다른 터미널 세션에서 사용할 수 있도록 합니다.

```
echo "DEV_FARM_BUCKET=$DEV_FARM_BUCKET" >> ~/.bashrc
```

4. 대기열에 대한 AWS Identity and Access Management (IAM) 역할을 생성합니다.

```
aws iam create-role --role-name "${DEV_FARM_NAME}QueueRole" \
    --assume-role-policy-document \
        '{
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "credentials.deadline.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }'
aws iam put-role-policy \
    --role-name "${DEV_FARM_NAME}QueueRole" \
    --policy-name S3BucketsAccess \
    --policy-document \
            '{
                "Version": "2012-10-17",
                "Statement": [
                {
                    "Action": [
                        "s3:GetObject*",
                        "s3:GetBucket*",
                        "s3:List*",
                        "s3:DeleteObject*",
                        "s3:PutObject",
                        "s3:PutObjectLegalHold",
                        "s3:PutObjectRetention",
                        "s3:PutObjectTagging",
                        "s3:PutObjectVersionTagging",
                        "s3:Abort*"
                    ],
                    "Resource": [
                        "arn:aws:s3:::'$DEV_FARM_BUCKET'",
                        "arn:aws:s3:::'$DEV_FARM_BUCKET'/*"
                    ],
```

```
                    "Effect": "Allow"
                }
            ]
            }'
```

5. 작업 첨부 설정 및 IAM 역할을 포함하도록 대기열을 업데이트하십시오.

```
QUEUE_ROLE_ARN="arn:aws:iam::$(aws sts get-caller-identity \
        --query "Account" --output text):role/${DEV_FARM_NAME}QueueRole"
aws deadline update-queue \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --role-arn $QUEUE_ROLE_ARN \
    --job-attachment-settings \
        '{
            "s3BucketName": "'$DEV_FARM_BUCKET'",
            "rootPrefix": "JobAttachments"
        }'
```

6. 대기열을 업데이트했는지 확인하세요.

```
deadline queue get
```

다음과 같은 출력이 표시됩니다.

```
...
jobAttachmentSettings:
  s3BucketName: DEV_FARM_BUCKET
  rootPrefix: JobAttachments
roleArn: arn:aws:iam::ACCOUNT_NUMBER:role/DeveloperFarmQueueRole
...
```

작업 첨부 파일과 simple_file_job 함께 제출

작업 첨부 파일을 사용하는 경우 작업 번들은 Deadline Cloud에 작업의 데이터 흐름을 결정하는 데 필요한 충분한 정보 (예: PATH 매개변수 사용) 를 제공해야 합니다. 의 경우 Deadline Cloud에 데이터 흐름이 입력 `template.yaml` 파일과 출력 파일에 있음을 알리도록 파일을 편집했습니다. simple_file_job

작업 첨부 파일 구성을 대기열에 추가한 후 simple_file_job 샘플을 작업 첨부 파일과 함께 제출할 수 있습니다. 이렇게 하면 로깅 및 작업 출력을 보고 작업 첨부 파일이 제대로 작동하는지 확인할 수 있습니다. simple_file_job

simple_file_job 작업 번들을 작업 첨부 파일과 함께 제출하려면

1. 첫 번째 CloudShell 탭을 선택한 다음 디렉토리를 여십시오. `JobBundle-Samples`
2.
```
cd ~/AmazonDeadlineCloud-DocumentationAndSamples/JobBundle-Samples
```
3. simple_file_job을 대기열에 제출하십시오. 업로드를 확인하라는 메시지가 표시되면 입력합니다. **y**

```
deadline bundle submit simple_file_job \
    -p InFile=simple_job/template.yaml \
    -p OutFile=hash-jobattachments.txt
```
4. 작업 첨부 파일 데이터 전송 세션 로그 출력을 보려면 두 번째 CloudShell 탭을 선택합니다.

```
JOB_ID=$(deadline config get defaults.job_id)
SESSION_ID=$(aws deadline list-sessions \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --job-id $JOB_ID \
        --query "sessions[0].sessionId" \
        --output text)
cat ~/demoenv-logs/$DEV_QUEUE_ID/$SESSION_ID.log
```
5. 세션 내에서 실행된 세션 작업을 나열하십시오.

```
aws deadline list-session-actions \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --job-id $JOB_ID \
    --session-id $SESSION_ID
```

다음과 같은 출력이 표시됩니다.

```
{
    "sessionactions": [
        {
            "sessionActionId": "sessionaction-123-0",
            "status": "SUCCEEDED",
```

```
            "startedAt": "<timestamp>",
            "endedAt": "<timestamp>",
            "progressPercent": 100.0,
            "definition": {
                "syncInputJobAttachments": {}
            }
        },
        {
            "sessionActionId": "sessionaction-123-1",
            "status": "SUCCEEDED",
            "startedAt": "<timestamp>",
            "endedAt": "<timestamp>",
            "progressPercent": 100.0,
            "definition": {
                "taskRun": {
                    "taskId": "task-abc-0",
                    "stepId": "step-def"
                }
            }
        }
    ]
}
```

첫 번째 세션 작업은 입력 작업 첨부 파일을 다운로드한 반면, 두 번째 작업은 이전과 같이 작업을 실행한 다음 출력 작업 첨부 파일을 업로드했습니다.

6. 출력 디렉터리를 나열합니다.

```
ls *.txt
```

와 같은 `hash.txt` 출력이 표시되지만 `hash-jobattachments.txt` 존재하지 않습니다.

7. 가장 최근 작업의 결과를 다운로드합니다.

```
deadline job download-output
```

8. 다운로드한 파일의 출력을 볼 수 있습니다.

```
cat hash-jobattachments.txt
```

다음과 같은 출력이 표시됩니다.

```
eaa2df5d34b54be5ac34c56a24a8c237b8487231a607eaf530a04d76b89c9cd3  /tmp/openjd/
session-123/assetroot-abc/simple_job/template.yaml
```

Amazon S3에 작업 첨부 파일이 저장되는 방식 이해

AWS Command Line Interface (AWS CLI) 를 사용하여 Amazon S3 버킷에 저장된 작업 첨부 데이터를 업로드하거나 다운로드할 수 있습니다. Deadline Cloud가 Amazon S3에 작업 첨부 파일을 저장하는 방법을 이해하면 워크로드 및 파이프라인 통합을 개발할 때 도움이 됩니다.

데드라인 클라우드 작업 첨부 파일이 Amazon S3에 저장되는 방식을 검사하려면

1. 첫 번째 CloudShell 탭을 선택한 다음 작업 번들 샘플 디렉토리를 여십시오.

   ```
   cd ~/AmazonDeadlineCloud-DocumentationAndSamples/JobBundle-Samples
   ```

2. 작업 속성을 살펴보세요.

   ```
   deadline job get
   ```

   다음과 같은 출력이 표시됩니다.

   ```
   parameters:
     Message:
       string: Welcome to Amazon Deadline Cloud!
     InFile:
       path: /home/cloudshell-user/AmazonDeadlineCloud-DocumentationAndSamples/
   JobBundle-Samples/simple_job/template.yaml
     OutFile:
       path: /home/cloudshell-user/AmazonDeadlineCloud-DocumentationAndSamples/
   JobBundle-Samples/hash-jobattachments.txt
   attachments:
     manifests:
     - rootPath: /home/cloudshell-user/AmazonDeadlineCloud-DocumentationAndSamples/
   JobBundle-Samples
       rootPathFormat: posix
       outputRelativeDirectories:
       - .
       inputManifestPath: farm-3040c59a5b9943d58052c29d907a645d/queue-
   cde9977c9f4d4018a1d85f3e6c1a4e6e/Inputs/
   f46af01ca8904cd8b514586671c79303/0d69cd94523ba617c731f29c019d16e8_input.xxh128
   ```

```
    inputManifestHash: f95ef91b5dab1fc1341b75637fe987ee
  fileSystem: COPIED
```

첨부 파일 필드에는 작업 실행 시 사용하는 입력 및 출력 데이터 경로를 설명하는 매니페스트 구조 목록이 포함됩니다. 작업을 제출한 컴퓨터의 로컬 디렉터리 경로를 `rootPath` 보려면 보십시오. 매니페스트 파일이 포함된 Amazon S3 객체 접미사를 보려면 를 참조하십시오. `inputManifestFile` 매니페스트 파일에는 작업 입력 데이터의 디렉터리 트리 스냅샷에 대한 메타데이터가 들어 있습니다.

3. Amazon S3 매니페스트 객체를 예쁘게 인쇄하여 작업의 입력 디렉터리 구조를 확인합니다.

```
MANIFEST_SUFFIX=$(aws deadline get-job \
     --farm-id $DEV_FARM_ID \
     --queue-id $DEV_QUEUE_ID \
     --job-id $JOB_ID \
     --query "attachments.manifests[0].inputManifestPath" \
     --output text)
 aws s3 cp s3://$DEV_FARM_BUCKET/JobAttachments/Manifests/$MANIFEST_SUFFIX - | jq .
```

다음과 같은 출력이 표시됩니다.

```
{
     "hashAlg": "xxh128",
     "manifestVersion": "2023-03-03",
     "paths": [
     {
         "hash": "2ec297b04c59c4741ed97ac8fb83080c",
         "mtime": 1698186190000000,
         "path": "simple_job/template.yaml",
         "size": 445
     }
     ],
     "totalSize": 445
 }
```

4. 출력 작업 첨부 파일의 매니페스트가 들어 있는 Amazon S3 접두사를 생성하고 그 아래에 객체를 나열합니다.

```
SESSION_ACTION=$(aws deadline list-session-actions \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --job-id $JOB_ID \
```

```
    --session-id $SESSION_ID \
    --query "sessionActions[?definition.taskRun != null] | [0]")
STEP_ID=$(echo $SESSION_ACTION | jq -r .definition.taskRun.stepId)
TASK_ID=$(echo $SESSION_ACTION | jq -r .definition.taskRun.taskId)
TASK_OUTPUT_PREFIX=JobAttachments/Manifests/$DEV_FARM_ID/$DEV_QUEUE_ID/$JOB_ID/
$STEP_ID/$TASK_ID/
aws s3api list-objects-v2 --bucket $DEV_FARM_BUCKET --prefix $TASK_OUTPUT_PREFIX
```

출력 작업 첨부 파일은 작업 리소스에서 직접 참조되지 않고 대신 팜 리소스 ID를 기반으로 Amazon S3 버킷에 배치됩니다.

5. 특정 세션 작업 ID에 대한 최신 매니페스트 객체 키를 가져온 다음 매니페스트 객체를 예쁘게 인쇄하십시오.

```
SESSION_ACTION_ID=$(echo $SESSION_ACTION | jq -r .sessionActionId)
 MANIFEST_KEY=$(aws s3api list-objects-v2 \
     --bucket $DEV_FARM_BUCKET \
     --prefix $TASK_OUTPUT_PREFIX \
     --query "Contents[*].Key" --output text \
     | grep $SESSION_ACTION_ID \
     | sort | tail -1)
 MANIFEST_OBJECT=$(aws s3 cp s3://$DEV_FARM_BUCKET/$MANIFEST_KEY -)
 echo $MANIFEST_OBJECT | jq .
```

출력에서 다음과 같은 파일 속성을 확인할 `hash-jobattachments.txt` 수 있습니다.

```
{
     "hashAlg": "xxh128",
     "manifestVersion": "2023-03-03",
     "paths": [
     {
         "hash": "f60b8e7d0fabf7214ba0b6822e82e08b",
         "mtime": 1698785252554950,
         "path": "hash-jobattachments.txt",
         "size": 182
     }
     ],
     "totalSize": 182
 }
```

작업에는 작업 실행당 매니페스트 개체가 하나밖에 없지만 일반적으로 작업 실행당 개체 수가 더 많을 수 있습니다.

6. 접두사 아래에 있는 콘텐츠 주소 지정이 가능한 Amazon S3 스토리지 출력을 확인합니다. `Data`

```
FILE_HASH=$(echo $MANIFEST_OBJECT | jq -r .paths[0].hash)
FILE_PATH=$(echo $MANIFEST_OBJECT | jq -r .paths[0].path)
aws s3 cp s3://$DEV_FARM_BUCKET/JobAttachments/Data/$FILE_HASH -
```

다음과 같은 출력이 표시됩니다.

```
eaa2df5d34b54be5ac34c56a24a8c237b8487231a607eaf530a04d76b89c9cd3  /tmp/openjd/
session-123/assetroot-abc/simple_job/template.yaml
```

5단계: Deadline Cloud의 개발자 팜에 서비스 관리 플릿 추가

AWS CloudShell 대규모 워크로드를 테스트하기에 충분한 컴퓨팅 파워를 제공하지 않습니다. 또한 여러 작업자 호스트에 작업을 분배하는 작업과 함께 작동하도록 구성되지 않았습니다.

를 사용하는 CloudShell 대신 개발자 팜에 Auto Scaling 서비스 관리 플릿 (SMF) 을 추가할 수 있습니다. SMF는 대규모 워크로드에 충분한 컴퓨팅 파워를 제공하며 여러 작업자 호스트에 작업을 분산해야 하는 작업을 처리할 수 있습니다. CMF 작업자를 종료하지 않는 한 스케줄러는 SMF 작업자와 CMF 작업자를 모두 사용하여 작업을 실행합니다.

개발자 팜에 서비스 관리 플릿을 추가하려면

1. 아직 설치하지 않았다면 AWS Command Line Interface (AWS CLI) 를 설치하고 구성하세요. 자세한 내용은 의 최신 버전 설치 또는 업데이트를 참조하십시오 AWS CLI.
2. 첫 번째 AWS CloudShell 탭을 선택한 다음 서비스 관리형 플릿을 생성하고 플릿 ID를 추가합니다 `.bashrc`. 이렇게 하면 다른 터미널 세션에서도 사용할 수 있습니다.

```
FLEET_ROLE_ARN="arn:aws:iam::$(aws sts get-caller-identity \
         --query "Account" --output text):role/${DEV_FARM_NAME}FleetRole"
 aws deadline create-fleet \
     --farm-id $DEV_FARM_ID \
     --display-name "$DEV_FARM_NAME SMF" \
     --role-arn $FLEET_ROLE_ARN \
     --max-worker-count 5 \
```

```
    --configuration \
        '{
            "serviceManagedEc2": {
                "instanceCapabilities": {
                    "vCpuCount": {
                        "min": 2,
                        "max": 4
                    },
                    "memoryMiB": {
                        "min": 512
                    },
                    "osFamily": "linux",
                    "cpuArchitectureType": "x86_64"
                },
                "instanceMarketOptions": {
                    "type": "spot"
                }
            }
        }'

echo "DEV_SMF_ID=$(aws deadline list-fleets \
        --farm-id $DEV_FARM_ID \
        --query "fleets[?displayName=='$DEV_FARM_NAME SMF'].fleetId \
        | [0]" --output text)" >> ~/.bashrc
source ~/.bashrc
```

3. SMF를 대기열에 연결합니다.

```
aws deadline create-queue-fleet-association \
    --farm-id $DEV_FARM_ID \
    --queue-id $DEV_QUEUE_ID \
    --fleet-id $DEV_SMF_ID
```

4.

> **Note**
>
> CMF 작업자를 종료하지 않는 한 스케줄러는 SMF 작업자와 CMF 작업자를 모두 사용하여 작업을 실행합니다.

대기열에 제출하십시오simple_file_job. 업로드를 확인하라는 메시지가 표시되면 입력합니다**y**.

```
deadline bundle submit simple_file_job \
```

```
    -p InFile=simple_job/template.yaml \
    -p OutFile=hash-jobattachments.txt
```

5. SMF가 제대로 작동하는지 확인합니다.

```
deadline fleet get
```

- 작업자가 시작하는 데 몇 분 정도 걸릴 수 있습니다.
- 고객 관리형 플릿 및 서비스 관리형 플릿은 다음과 같습니다`ACTIVE`. `queueFleetAssociationsStatus`
- `autoScalingStatus`SMF가 에서 `GROWING` 로 `STEADY` 변경됩니다.

상태는 다음과 비슷하게 표시됩니다.

```
fleetId: fleet-2cc78e0dd3f04d1db427e7dc1d51ea44
farmId: farm-63ee8d77cdab4a578b685be8c5561c4a
displayName: DeveloperFarm SMF
description: ''
status: ACTIVE
autoScalingStatus: STEADY
targetWorkerCount: 0
workerCount: 0
minWorkerCount: 0
maxWorkerCount: 5
```

6. 제출한 작업에 대한 로그를 보십시오. 이 로그는 CloudShell 파일 시스템이 아닌 Amazon CloudWatch Logs의 로그에 저장됩니다.

```
JOB_ID=$(deadline config get defaults.job_id)
SESSION_ID=$(aws deadline list-sessions \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --job-id $JOB_ID \
        --query "sessions[0].sessionId" \
        --output text)
aws logs tail /aws/deadline/$DEV_FARM_ID/$DEV_QUEUE_ID \
    --log-stream-names $SESSION_ID
```

6단계: 데드라인 클라우드에서 팜 리소스 정리

새 워크로드 및 파이프라인 통합을 개발하고 테스트하려면 이 자습서를 위해 만든 Deadline Cloud 개발자 팜을 계속 사용할 수 있습니다. 개발자 팜이 더 이상 필요하지 않은 경우 팜, 플릿, 대기열, AWS Identity and Access Management (IAM) 역할 및 Amazon CloudWatch Logs의 로그를 포함한 리소스를 삭제할 수 있습니다. 이러한 리소스를 삭제한 후에는 자습서를 다시 시작해야 리소스를 사용할 수 있습니다. 자세한 정보는 데드라인 클라우드용 개발자 워크스테이션 설정을 참조하세요.

개발자 팜 리소스를 정리하려면

1. 아직 설치하지 않았다면 AWS Command Line Interface (AWS CLI) 를 설치하고 구성하세요. 자세한 내용은 의 최신 버전 설치 또는 업데이트를 참조하십시오 AWS CLI.
2. 첫 번째 CloudShell 탭을 선택한 다음 해당 대기열의 모든 대기열 연결을 중지하십시오.

```
FLEETS=$(aws deadline list-queue-fleet-associations \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --query "queueFleetAssociations[].fleetId" \
        --output text)
for FLEET_ID in $FLEETS; do
    aws deadline update-queue-fleet-association \
        --farm-id $DEV_FARM_ID \
        --queue-id $DEV_QUEUE_ID \
        --fleet-id $FLEET_ID \
        --status STOP_SCHEDULING_AND_CANCEL_TASKS
done
```

3. 큐 플릿 연결을 나열하세요.

```
aws deadline list-queue-fleet-associations \
     --farm-id $DEV_FARM_ID \
     --queue-id $DEV_QUEUE_ID
```

출력이 `"status": "STOPPED"` 보고될 때까지 명령을 다시 실행해야 할 수도 있습니다. 그러면 다음 단계로 넘어갈 수 있습니다. 이 프로세스는 완료하는 데 몇 분 정도 걸립니다.

```
{
    "queueFleetAssociations": [
        {
            "queueId": "queue-abcdefgh01234567890123456789012id",
            "fleetId": "fleet-abcdefgh01234567890123456789012id",
```

```
            "status": "STOPPED",
            "createdAt": "2023-11-21T20:49:19+00:00",
            "createdBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName",
            "updatedAt": "2023-11-21T20:49:38+00:00",
            "updatedBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName"
        },
        {
            "queueId": "queue-abcdefgh01234567890123456789012id",
            "fleetId": "fleet-abcdefgh01234567890123456789012id",
            "status": "STOPPED",
            "createdAt": "2023-11-21T20:32:06+00:00",
            "createdBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName",
            "updatedAt": "2023-11-21T20:49:39+00:00",
            "updatedBy": "arn:aws:sts::123456789012:assumed-role/RoleToBeAssumed/
MySessionName"
        }
    ]
}
```

4. 큐의 모든 큐-플릿 연결을 삭제하십시오.

```
for FLEET_ID in $FLEETS; do
     aws deadline delete-queue-fleet-association \
         --farm-id $DEV_FARM_ID \
         --queue-id $DEV_QUEUE_ID \
         --fleet-id $FLEET_ID
 done
```

5. 대기열에 연결된 모든 플릿을 삭제하십시오.

```
for FLEET_ID in $FLEETS; do
     aws deadline delete-fleet \
         --farm-id $DEV_FARM_ID \
         --fleet-id $FLEET_ID
 done
```

6. 대기열을 삭제합니다.

```
aws deadline delete-queue \
     --farm-id $DEV_FARM_ID \
```

```
    --queue-id $DEV_QUEUE_ID
```

7. 팜을 삭제합니다.

```
aws deadline delete-farm \
    --farm-id $DEV_FARM_ID
```

8. 팜의 다른 AWS 리소스를 삭제하세요.

   a. 플릿 AWS Identity and Access Management (IAM) 역할을 삭제합니다.

```
aws iam delete-role-policy \
    --role-name "${DEV_FARM_NAME}FleetRole" \
    --policy-name WorkerPermissions
aws iam delete-role \
    --role-name "${DEV_FARM_NAME}FleetRole"
```

   b. 큐 IAM 역할을 삭제합니다.

```
aws iam delete-role-policy \
    --role-name "${DEV_FARM_NAME}QueueRole" \
    --policy-name S3BucketsAccess
aws iam delete-role \
    --role-name "${DEV_FARM_NAME}QueueRole"
```

   c. Amazon CloudWatch 로그 로그 그룹을 삭제합니다. 각 대기열과 플릿에는 고유한 로그 그룹이 있습니다.

```
aws logs delete-log-group \
    --log-group-name "/aws/deadline/$DEV_FARM_ID/$DEV_QUEUE_ID"
aws logs delete-log-group \
    --log-group-name "/aws/deadline/$DEV_FARM_ID/$DEV_CMF_ID"
aws logs delete-log-group \
    --log-group-name "/aws/deadline/$DEV_FARM_ID/$DEV_SMF_ID"
```

데드라인 클라우드 제출자를 설정하세요.

이 프로세스는 AWS Deadline Cloud 제출자를 설치, 설정 및 시작하려는 관리자와 아티스트를 위한 것입니다. 데드라인 클라우드 제출자는 디지털 콘텐츠 제작 (DCC) 플러그인입니다. 아티스트는 이를 사용하여 익숙한 타사 DCC 인터페이스에서 작업을 제출합니다.

> **Note**
>
> 아티스트가 렌더링을 제출하는 데 사용할 모든 워크스테이션에서 이 프로세스를 완료해야 합니다.

주제

- 1단계: 데드라인 클라우드 제출자 설치
- 2단계: 데드라인 클라우드 모니터 설치 및 설정
- 3단계: 데드라인 클라우드 제출자 시작

1단계: 데드라인 클라우드 제출자 설치

다음 섹션에서는 Deadline Cloud 제출자를 설치하는 단계를 안내합니다.

제출자 설치 프로그램 다운로드

Deadline Cloud 제출자를 설치하려면 먼저 제출자 설치 프로그램을 다운로드해야 합니다. 현재 데드라인 클라우드 제출자 설치 프로그램은 및 만 지원합니다. Windows Linux

1. 데드라인 클라우드 콘솔에 AWS Management Console 로그인하고 엽니다.
2. 측면 탐색 창에서 다운로드를 선택합니다.
3. 데드라인 클라우드 제출자 설치 프로그램 섹션을 찾으세요.
4. 컴퓨터 운영 체제의 설치 프로그램을 선택한 다음 다운로드를 선택합니다.

(선택 사항) 다운로드한 소프트웨어의 정품 확인

다운로드한 소프트웨어가 정품인지 확인하려면 다음 중 하나 Windows 또는 Linux 에 대한 절차를 사용하십시오.

> **Note**
>
> 이 지침에 따라 먼저 설치 프로그램을 확인한 다음 Deadline Cloud 모니터를 다운로드한 후 다음 섹션 (2단계) 에서 확인할 수 있습니다.

Windows

다운로드한 파일의 진위 여부를 확인하려면 다음 단계를 완료하세요.

1. 다음 명령에서 확인하려는 파일로 *file* 바꾸십시오. 예를 들어 ***C:\PATH\TO\MY\DeadlineCloudSubmitter-windows-x64-installer.exe*** 입니다. 또한 설치된 SignTool SDK 버전으로 교체하십시오*signtool-sdk-version*. 예를 들어 **10.0.22000.0**입니다.

   ```
   "C:\Program Files (x86)\Windows Kits\10\bin\signtool-sdk-version\x86\signtool.exe" verify /vfile
   ```

2. 예를 들어 다음 명령어를 실행하여 Deadline Cloud 제출자 설치 프로그램 파일을 확인할 수 있습니다.

   ```
   "C:\Program Files (x86)\Windows Kits\10\bin
   \10.0.22000.0\x86\signtool.exe" verify /v DeadlineCloudSubmitter-
   windows-x64-installer.exe
   ```

Linux

다운로드한 파일의 진위 여부를 확인하려면 명령줄 도구를 사용하세요. gpg

1. 다음 명령을 실행하여 Deadline Cloud 제출자 설치 프로그램의 OpenPGP 키를 가져옵니다.

   ```
    gpg --import --armor <<EOF
   -----BEGIN PGP PUBLIC KEY BLOCK-----

   mQINBGX6GQsBEADduUtJgqSXI+q76O6fsFwEYKmbnlyL0xKvlq32EZuyv0otZo5L
   le4m5Gg52AzrvPvDiUTLooAlvYeozaYyirIGsK08Ydz0Ftdjroiuh/mw9JSJDJRI
   rnRn5yKet1JFezkjopA3pjsTBP6lW/mb1bDBDEwwwtH0x9lV7A03FJ9T7Uzu/qSh
   qO/UYdkafro3cPASvkqgDt2tCvURfBcUCAjZVFcLZcVD5iwXacxvKsxxS/e7kuVV
   I1+VGT8Hj8XzWYhjCZxOLZk/fvpYPMyEEujN0fYUp6RtMIXve0C9awwMCy5nBG2J
   eE2Ol5DsCpTaBd4Fdr3LWcSs8JFA/YfP9auL3NczOozPoVJt+fw8CBlVIXO0J7l5
   hvHDjcC+5v0wxqAlMG6+f/SX7CT8FXK+L3iOJ5gBYUNXqHSxUdv8kt76/KVmQa1B
   Akl+MPKpMq+lhw++S3G/lXqwWaDNQbRRw7dSZHymQVXvPp1nsqc3hV7KlOM+6s6g
   1g4mvFY4lf6DhptwZLWyQXU8rBQpojvQfiSmDFrFPWFi5BexesuVnkGIolQoklKx
   AVUSdJPVEJCteyy7td4FPhBaSqT5vW3+ANbr9b/uoRYWJvn17dN0cc9HuRh/Ai+I
   nkfECo2WUDLZ0fEKGjGyFX+todWvJXjvc5kmE9Ty5vJp+M9Vvb8jd6t+mwARAQAB
   tCxBV1MgRGVhZGxpbmUgQ2xvdWQgPGF3cy1kZWFkbGluZUBhbWF6b24uY29tPokC
   VwQTAQgAQRYhBLhAwIwpqQeWoHH6pfbNPOa3bzzvBQJl+hkLAxsvBAUJA8JnAAUL
   CQgHAgIiAgYVCgkICwIDFgIBAh4HAheAAAoJEPbNPOa3bzzvKswQAJXzKSAY8sY8
   ```

```
F6Eas2oYwIDDdDurs8FiEnFghjUEO6MTt9AykF/jw+CQg2UzFtEyObHBymhgmhXE
3buVeom96tgM3ZDfZu+sxi5pGX6oAQnZ6riztN+VpkpQmLgwtMGpSMLl3KLwnv2k
WK8mrR/fPMkfdaewB7A6RIUYiW33GAL4KfMIs8/vIwIJw99NxHpZQVoU6dFpuDtE
1OuxGcCqGJ7mAmo6H/YawSNp2Ns80gyqIKYo7o3LJ+WRroIRlQyctq8gnR9JvYXX
42ASqLq5+OXKo4qh81blXKYqtc176BbbSNFjWnzIQgKDgNiHFZCdcOVgqDhwO15r
NICbqqwwNLj/Fr2kecYx180Ktpl0jOOw5IOyh3bf3MVGWnYRdjvA1v+/CO+55N4g
z0kf50Lcdu5RtqV10XBCifn28pecqPaSdYcssYSRl5DLiFktGbNzTGcZZwITTKQc
af8PPdTGtnnb6P+cdbW3bt9MVtN5/dgSHLThnS8MPEuNCtkTnpXshuVuBGgwBMdb
qUC+HjqvhZzbwns8dr5WI+6HWNBFgGANn6ageYl58vVp0UkuNP8wcWjRARciHXZx
ku6W2jPTHDWGNrBQO2Fx7fd2QYJheIPPAShHcfJO+xgWCof45D0vAxAJ8gGg9Eq+
gFWhsx4NSHn2gh1gDZ41Ou/4exJ1lwPM
=uVaX
-----END PGP PUBLIC KEY BLOCK-----
EOF
```

2. 키를 신뢰할 수 있는지 여부를 결정하세요. OpenPGP 위 키의 신뢰 여부를 결정할 때 고려해야 할 몇 가지 요소는 다음과 같습니다.

   - 이 웹사이트에서 GPG 키를 받을 때 사용한 인터넷 연결은 안전합니다.
   - 이 웹사이트에 접속하는 기기는 안전합니다.
   - AWS 이 웹 사이트의 OpenPGP 공개 키 호스팅을 보호하기 위한 조치를 취했습니다.

3. OpenPGP키를 신뢰하기로 결정했다면 다음 예와 gpg 마찬가지로 키를 신뢰할 수 있도록 편집하십시오.

```
$ gpg --edit-key 0xB840C08C29A90796A071FAA5F6CD3CE6B76F3CEF

    gpg (GnuPG) 2.0.22; Copyright (C) 2013 Free Software Foundation, Inc.
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law.


    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: unknown       validity: unknown
    [ unknown] (1). AWS Deadline Cloud example@example.com

    gpg> trust
    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: unknown       validity: unknown
    [ unknown] (1). AWS Deadline Cloud aws-deadline@amazon.com

    Please decide how far you trust this user to correctly verify other users'
 keys
```

```
   (by looking at passports, checking fingerprints from different sources,
etc.)

     1 = I don't know or won't say
     2 = I do NOT trust
     3 = I trust marginally
     4 = I trust fully
     5 = I trust ultimately
     m = back to the main menu

   Your decision? 5
   Do you really want to set this key to ultimate trust? (y/N) y

   pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                        trust: ultimate      validity: unknown
   [ unknown] (1). AWS Deadline Cloud aws-deadline@amazon.com
   Please note that the shown key validity is not necessarily correct
   unless you restart the program.

   gpg> quit
```

4. 설치 프로그램 확인

   설치 프로그램을 확인하려면 다음 단계를 완료하십시오.

   a. Deadline Cloud 콘솔 다운로드 페이지로 돌아가서 Deadline Cloud 제출자 설치 프로그램의 서명 파일을 다운로드합니다.

   b. 다음을 실행하여 데드라인 클라우드 제출자 설치 프로그램의 서명을 확인합니다.

      ```
      gpg --verify ./DeadlineCloudSubmitter-linux-x64-
      installer.run.sig ./DeadlineCloudSubmitter-linux-x64-
      installer.run
      ```

5. 데드라인 클라우드 모니터 확인

   > **Note**
   >
   > 서명 파일 또는 플랫폼별 방법을 사용하여 Deadline Cloud 모니터 다운로드를 확인할 수 있습니다. 플랫폼별 방법은 다운로드한 파일 유형에 따른 Linux (DEB) Linux (AppImage) 탭 또는 탭을 참조하십시오.

서명 파일이 있는 Deadline Cloud 모니터 데스크톱 애플리케이션을 확인하려면 다음 단계를 완료하십시오.

a. Deadline Cloud 콘솔 다운로드 페이지로 돌아가서 해당.sig 파일을 다운로드한 다음 실행합니다.

   .deb의 경우:

   ```
   gpg --verify ./deadline-cloud-
   monitor_<APP_VERSION>_amd64.deb.sig ./deadline-cloud-
   monitor_<APP_VERSION>_amd64.deb
   ```

   를 위해. AppImage:

   ```
   gpg --verify ./deadline-cloud-
   monitor_<APP_VERSION>_amd64.AppImage.sig ./deadline-cloud-
   monitor_<APP_VERSION>_amd64.AppImage
   ```

b. 출력이 다음과 비슷한지 확인합니다.

   ```
   gpg: Signature made Mon Apr 1 21:10:14 2024 UTC
   ```

   ```
   gpg: using RSA key B840C08C29A90796A071FAA5F6CD3CE6B7
   ```

   출력에 다음과 `Good signature from "AWS Deadline Cloud"` 같은 문구가 포함되어 있으면 서명이 성공적으로 확인되었음을 의미하며 Deadline Cloud 모니터 설치 스크립트를 실행할 수 있습니다.

Linux (DEB)

Linux.deb 바이너리를 사용하는 패키지를 확인하려면 먼저 탭에서 1-3단계를 완료하십시오. Linux

dpkg는 대부분의 기반 배포판의 핵심 패키지 관리 도구입니다. debian Linux 도구를 사용하여.deb 파일을 확인할 수 있습니다.

1. 데드라인 클라우드 콘솔 다운로드 페이지에서 데드라인 클라우드 모니터 .deb 파일을 다운로드합니다.
2. 확인하려는.deb 파일의 버전으로 ***<APP_VERSION>*** 바꾸십시오.

   ```
   dpkg-sig --verify deadline-cloud-monitor_<APP_VERSION>_amd64.deb
   ```

3. 출력은 다음과 비슷할 것입니다.

```
Processing deadline-cloud-monitor_1.1.1_amd64.deb... GOODSIG
_gpgbuilder B840C08C29A90796A071FAA5F6CD3C 171200
```

4. .deb 파일을 확인하려면 해당 파일이 출력에 GOODSIG 있는지 확인하십시오.

Linux (AppImage)

a를 사용하는 패키지를 확인하려면 Linux AppImage 바이너리라면 먼저 Linux 탭에서 1~3단계를 완료하세요.

1. 데드라인 클라우드 콘솔 다운로드 페이지에서 데드라인 클라우드 모니터를 다운로드합니다. AppImage 파일.
2. <APP_VERSION>의 버전으로 바꾸려면. AppImage 확인하려는 파일은 다음 단계를 완료하십시오.

   a. 에서 서명을 작성합니다. AppImage 파일을.sig 파일에 저장합니다.

   ```
   ./deadline-cloud-monitor_<APP_VERSION>_amd64.AppImage
   --appimage-signature > ./deadline-cloud-
   monitor_<APP_VERSION>_amd64_.AppImage.sig
   ```

   b. 생성된 .sig 파일을 사용하여 다음 명령을 사용하여 확인합니다.

   ```
   gpg --verify ./deadline-cloud-
   monitor_<APP_VERSION>_amd64.AppImage.sig
   ```

   c. (선택 사항) 권한 거부 오류가 표시되면 다음 명령을 사용하여 실행 권한을 추가합니다.

   ```
   chmod +x ./deadline-cloud-monitor_<APP_VERSION>_amd64.AppImage
   ```

   d. 출력이 다음과 비슷한지 확인합니다.

   ```
   gpg: Signature made Mon Apr 1 21:10:14 2024 UTC

   gpg: using RSA key B840C08C29A90796A071FAA5F6CD3CE6B7
   ```

   출력에 다음과 Good signature from "AWS Deadline Cloud" 같은 문구가 포함되어 있으면 서명이 성공적으로 확인되었음을 의미하며 Deadline Cloud 모니터 설치 스크립트를 실행할 수 있습니다.

데드라인 클라우드 제출자를 설치합니다.

또는 를 사용하여 데드라인 클라우드 제출자를 설치할 수 있습니다. Windows Linux 설치 프로그램을 사용하여 다음 제출자를 설치할 수 있습니다.

- 마야 2024
- 뉴크 14.0 - 15.0
- 후디니 19.5
- 키샷 12
- 블렌더 3.6
- 언리얼 엔진 5

Windows

1. 파일 브라우저에서 인스톨러를 다운로드한 폴더로 이동한 다음 선택합니다. `DeadlineCloudSubmitter-windows-x64-installer.exe`
   a. Windows에서 PC를 보호한 팝업이 표시되는 경우 추가 정보를 선택합니다.
   b. 어쨌든 [실행] 을 선택하십시오.
2. AWS 데드라인 클라우드 제출자 설정 마법사가 열리면 다음을 선택합니다.
3. 다음 단계 중 하나를 완료하여 설치 범위를 선택합니다.
   - 현재 사용자만 사용할 수 있도록 설치하려면 [사용자] 를 선택합니다.
   - 모든 사용자를 대상으로 설치하려면 [시스템] 을 선택합니다.

     시스템을 선택한 경우 다음 단계를 완료하여 설치 프로그램을 종료하고 관리자 권한으로 다시 실행해야 합니다.

     a. 를 **DeadlineCloudSubmitter-windows-x64-installer.exe** 마우스 오른쪽 단추로 클릭한 다음 관리자 권한으로 실행을 선택합니다.
     b. 관리자 자격 증명을 입력한 다음 [Yes] 를 선택합니다.
     c. 설치 범위로 [System] 을 선택합니다.
4. 설치 범위를 선택한 후 다음을 선택합니다.
5. 설치 디렉터리를 승인하려면 [다음] 을 다시 선택합니다.
6. 통합 제출자 또는 Nuke 설치하려는 제출자를 선택합니다.
7. 다음을 선택합니다.

8. 설치를 검토하고 다음을 선택합니다.
9. [다음] 을 다시 선택한 다음 [마침] 을 선택합니다.

Linux

> **Note**
>
> 데드라인 클라우드 통합 Nuke 설치 프로그램 Linux 및 데드라인 클라우드 모니터는 GLIBC 2.31 이상이 설치된 Linux 배포판에만 설치할 수 있습니다.

1. 터미널 창을 엽니다.
2. 설치 프로그램을 시스템 설치하려면 명령을 **sudo -i** 입력하고 Enter 키를 눌러 루트로 전환하십시오.
3. 설치 프로그램을 다운로드한 위치로 이동합니다.

   예를 들어 **cd /home/*USER*/Downloads**입니다.
4. 설치 프로그램을 실행 가능하게 만들려면 를 입력합니다. **chmod +x DeadlineCloudSubmitter-linux-x64-installer.run**
5. Deadline Cloud 제출자 설치 프로그램을 실행하려면 를 입력합니다. **./DeadlineCloudSubmitter-linux-x64-installer.run**
6. 설치 프로그램이 열리면 화면의 지시에 따라 설치 마법사를 완료합니다.

여기에 나열되지 않은 다른 제출자를 설치할 수 있습니다. Deadline Cloud 라이브러리를 사용하여 제출자를 빌드합니다. AWS Deadline 조직에서 이러한 라이브러리 및 제출자의 소스 코드를 찾을 수 있습니다. GitHub

2단계: 데드라인 클라우드 모니터 설치 및 설정

Windows또는 를 사용하여 데드라인 클라우드 모니터 데스크톱 애플리케이션을 설치할 수 Linux 있습니다.

Windows

1. 아직 로그인하지 않았다면 Deadline Cloud 콘솔에 로그인하여 여십시오. AWS Management Console
2. 왼쪽 탐색 창에서 다운로드를 선택합니다.

3. Deadline Cloud 모니터 섹션에서 컴퓨터 운영 체제에 맞는 파일을 선택합니다.
4. 데드라인 클라우드 모니터를 다운로드하려면 다운로드를 선택합니다.

Linux

RPM AppImage 배포판에 데드라인 클라우드 모니터를 설치하려면

1. 최신 데드라인 클라우드 모니터를 다운로드하세요. AppImage
2. AppImage 실행 파일을 만들려면 를 입력합니다**chmod a+x deadline-cloud-monitor_*<APP_VERSION>*_amd64.AppImage**.
3. 올바른 SSL 인증서 경로를 설정하려면 를 입력합니다. **sudo ln -sf /etc/ssl/certs/ca-bundle.crt /etc/ssl/certs/ca-certificates.crt**

데비안 AppImage 배포판에 데드라인 클라우드 모니터를 설치하려면

1. 최신 데드라인 클라우드 모니터를 다운로드하세요. AppImage
2. > **Note**
   >
   > 이 단계는 우분투 22 이상용입니다. 다른 버전의 Ubuntu의 경우 이 단계를 건너뛰십시오.

   libfuse2를 설치하려면 다음을 입력하십시오. **sudo apt udpate**

   sudo apt install libfuse2.
3. AppImage 실행 파일을 만들려면 를 입력합니다. **chmod a+x deadline-cloud-monitor_*<APP_VERSION>*_amd64.AppImage**

데비안 배포판에 데드라인 클라우드 모니터 데비안 패키지를 설치하려면

1. 최신 데드라인 클라우드 모니터 데비안 패키지를 다운로드하세요.
2. > **Note**
   >
   > 이 단계는 Ubuntu 22 이상용입니다. 다른 버전의 Ubuntu의 경우 이 단계를 건너뛰십시오.

libssl1.1을 설치하려면 다음을 입력하십시오. **wget http://nz2.archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.*<APP_VERSION>*.1f-1ubuntu2.22_amd64.deb**

sudo dpkg -i libssl1.*<APP_VERSION>*.1f-1ubuntu2.22_amd64.deb.

3. 데드라인 클라우드 모니터 데비안 패키지를 설치하려면 다음을 입력하십시오. **sudo apt update**

   sudo apt install ./deadline-cloud-monitor_*<APP_VERSION>*_amd64.deb.

4. 종속성이 충족되지 않은 패키지에서 설치가 실패하는 경우 손상된 패키지를 수정한 후 다음 명령을 실행하십시오.

   sudo apt --fix-missing update

   sudo apt update

   sudo apt install -f

다운로드를 완료한 후 다운로드한 소프트웨어의 신뢰성을 확인할 수 있습니다. 1단계에서 다운로드한 소프트웨어의 정품 확인을 참조하십시오.

Deadline Cloud Monitor를 다운로드하고 신뢰성을 확인한 후 다음 절차를 사용하여 Deadline Cloud 모니터를 설정합니다.

데드라인 클라우드 모니터를 설정하려면

1. 데드라인 클라우드 모니터를 엽니다.
2. 새 프로필을 만들라는 메시지가 표시되면 다음 단계를 완료하세요.
   a. URL 입력에 모니터 URL을 입력합니다. 모양은 다음과 같습니다. **https://*MY-MONITOR*.deadlinecloud.amazonaws.com/**
   b. 프로필 이름을 입력합니다.
   c. 프로필 만들기를 선택합니다.

      프로필이 생성되고 이제 생성한 프로필 이름을 사용하는 모든 소프트웨어와 자격 증명이 공유됩니다.

3. Deadline Cloud 모니터 프로필을 만든 후에는 프로필 이름이나 스튜디오 URL을 변경할 수 없습니다. 변경이 필요한 경우 대신 다음을 수행하세요.

   a. 프로필을 삭제합니다. 왼쪽 탐색 창에서 데드라인 클라우드 모니터, 설정, 삭제를 선택합니다.

   b. 원하는 대로 변경하여 새 프로필을 생성합니다.

4. 왼쪽 탐색 창에서 >Deadline Cloud 모니터 옵션을 사용하여 다음 작업을 수행합니다.

   - 데드라인 클라우드 모니터 프로필을 변경하여 다른 모니터에 로그인하십시오.
   - 자동 로그인을 활성화하면 이후에 Deadline Cloud 모니터를 열 때 모니터 URL을 입력할 필요가 없습니다.

5. 데드라인 클라우드 모니터 창을 닫습니다. 백그라운드에서 계속 실행되며 15분마다 자격 증명을 동기화합니다.

6. 렌더링 프로젝트에 사용하려는 각 디지털 콘텐츠 제작 (DCC) 응용 프로그램에 대해 다음 단계를 완료하십시오.

   a. 데드라인 클라우드 제출자에서 데드라인 클라우드 워크스테이션 구성을 엽니다.

   b. 워크스테이션 구성에서 데드라인 클라우드 모니터에서 만든 프로필을 선택합니다. 이제 Deadline Cloud 자격 증명이 이 DCC와 공유되므로 도구가 예상대로 작동할 것입니다.

3단계: 데드라인 클라우드 제출자 시작

다음 섹션에서는,, 및 에서 Deadline Cloud 제출자 플러그인을 시작하는 단계를 안내합니다. Blender Nuke Maya Houdini

데드라인 클라우드 제출자를 실행하려면 Blender

> **Note**
>
> 에 대한 Blender 지원은 서비스 관리 플릿에 대한 Conda 환경을 사용하여 제공됩니다. 자세한 정보는 기본 Conda 대기열 환경을 참조하세요.

1. Blender를 엽니다.
2. 에셋 루트 디렉터리 내에 종속성이 있는 Blender 씬을 엽니다.
3. 렌더 메뉴에서 데드라인 클라우드 다이얼로그를 선택합니다.

a. 데드라인 클라우드 제출자에서 아직 인증을 받지 않은 경우 자격 증명 상태에 NEEDS_LOGIN이 표시됩니다.
b. 로그인을 선택합니다.
c. 로그인 브라우저 창이 표시됩니다. 사용자 자격 증명으로 로그인합니다.
d. 허용을 선택합니다. 이제 로그인되었으며 자격 증명 상태가 인증됨으로 표시됩니다.

4. 제출을 선택합니다.

데드라인 클라우드 제출자를 시작하려면 Foundry Nuke

> **Note**
>
> 에 대한 Nuke 지원은 서비스 관리 플릿에 대한 Conda 환경을 사용하여 제공됩니다. 자세한 정보는 기본 Conda 대기열 환경을 참조하세요.

1. Nuke를 엽니다.
2. 에셋 루트 디렉터리 내에 종속성이 있는 Nuke 스크립트를 엽니다.
3. Thinkbox를 선택한 다음 Deadline Cloud에 제출을 선택하여 제출자를 시작합니다.

a. 데드라인 클라우드 제출자에서 아직 인증을 받지 않은 경우 자격 증명 상태는 NEEDS_LOGIN으로 표시됩니다.
b. 로그인을 선택합니다.
c. 로그인 브라우저 창에서 사용자 자격 증명으로 로그인합니다.
d. 허용을 선택합니다. 이제 로그인되었으며 자격 증명 상태가 인증됨으로 표시됩니다.

4. 제출을 선택합니다.

데드라인 클라우드 제출자를 시작하려면 Maya

> **Note**
>
> 서비스 관리 Arnold for Maya(MtoA) 플릿을 위한 Conda 환경을 사용하여 Maya 및 지원을 제공합니다. 자세한 정보는 기본 Conda 대기열 환경을 참조하세요.

1. Maya를 엽니다.

2. 프로젝트를 설정하고 에셋 루트 디렉터리 내에 있는 파일을 엽니다.
3. Windows → 설정/환경설정 → 플러그인 관리자를 선택합니다.
4. 제출자를 검색합니다. DeadlineCloud
5. 데드라인 클라우드 제출자 플러그인을 로드하려면 로드됨을 선택합니다.

    a. 데드라인 클라우드 제출자에서 아직 인증을 받지 않은 경우 자격 증명 상태가 NEEDS_LOGIN으로 표시됩니다.

    b. 로그인을 선택합니다.

    c. 로그인 브라우저 창이 표시됩니다. 사용자 자격 증명으로 로그인합니다.

    d. 허용을 선택합니다. 이제 로그인되었으며 자격 증명 상태가 인증됨으로 표시됩니다.
6. (선택 사항) Deadline Cloud 제출자 플러그인을 열 Maya 때마다 로드하려면 자동 로드를 선택합니다.
7. Deadline Cloud 셸프를 선택한 다음 녹색 버튼을 선택하여 제출자를 시작합니다.

데드라인 클라우드 제출자를 실행하려면 Houdini

> **Note**
>
> 에 대한 Houdini 지원은 서비스 관리 플릿에 대한 Conda 환경을 사용하여 제공됩니다. 자세한 정보는 기본 Conda 대기열 환경을 참조하세요.

1. Houdini를 엽니다.
2. 네트워크 편집기에서 /out 네트워크를 선택합니다.
3. 탭을 누르고 **`deadline`** Enter를 누릅니다.
4. 데드라인 클라우드 옵션을 선택하고 기존 네트워크에 연결합니다.
5. 데드라인 클라우드 노드를 두 번 클릭합니다.

데드라인 클라우드 제출자를 실행하려면 KeyShot

이는 데드라인 클라우드와 2를 이미 다운로드했다고 가정합니다. PySide

1. Deadline-Cloudfor-keyshot/Keyshot_Script/Submit to AWS Deadline Cloud.py 파일을 스크립트 폴더에 복사하거나 링크합니다. KeyShot

예를 Windows 들어 on의 경우 스크립트 폴더 위치는 다음과 같습니다. **C:/Users/*USER*/Documents/KeyShot 12/Scripts**

2. 다음 환경 변수를 설정합니다.

   a. 환경 변수를 **DEADLINE_PYTHON** 데드라인-클라우드와 PySide 2가 있는 Python 설치 경로로 설정합니다.

      예를 들어, onWindows, Python 3.10을 사용하는 경우 명령은 다음과 같을 수 **set DEADLINE_PYTHON=C:/Users/*USER*/AppData/Local/Programs/Python/Python310/python** 있습니다.

   b. 환경 변수를 **DEADLINE_KEYSHOT** keyshot_submitter 폴더의 경로로 설정합니다.

      예를 들어, 소스가 Windows 데스크톱에 있는 경우 명령어는 on일 수 있습니다. **set DEADLINE_KEYSHOT=C:/Users/*USER*/Desktop/deadline-cloud-for-keyshot/src/deadline/keyshot_submitter**

3. 환경 변수를 설정한 상태에서 실행합니다 KeyShot.
4. 제출자를 실행하려면 스크립팅 콘솔 KeyShot, Deadline Cloud에 AWS 제출, 실행을 선택합니다 Windows.

데드라인 클라우드 제출자를 실행하려면 Unreal Engine

이는 데드라인 클라우드를 이미 다운로드했다고 가정합니다.

1. Unreal Engine프로젝트에 사용할 폴더를 만들거나 여십시오.
2. 명령줄을 열고 다음 명령을 실행합니다.

   - `git clone https://github.com/aws-deadline/deadline-cloud-for-unreal-engine`
   - `cd deadline-cloud-for-unreal/test_projects`
   - `git lfs fetch –all`

3. 플러그인을 Unreal Engine 다운로드하려면 Unreal Engine 프로젝트 폴더를 열고 deadline-cloud-forunreal/test_projects/pull_ue_plugin.bat 를 실행합니다.

   그러면 플러그인 파일이 C:/LocalProjects/UnrealDeadlineCloudTestUnrealDeadlineCloudService/Plugins/에 저장됩니다.

4. 제출자를 다운로드하려면 폴더를 열고 실행합니다. UnrealDeadlineCloudService **deadline-cloud-forunreal/ test_projects/Plugins/UnrealDeadlineCloudService/ install_unreal_submitter.bat**
5. 에서 Unreal Engine 제출자를 실행하려면 다음 단계를 완료하십시오.

    a. 편집, > 프로젝트 설정을 선택합니다.
    b. 검색 창에 **movie render pipeline**를 입력합니다.
    c. 다음 무비 렌더 파이프라인 설정을 조정합니다.

        i. 기본 원격 실행기의 경우 를 입력합니다**MoviePipelineDeadlineCloudRemote Executor**.
        ii. 기본 실행자 작업의 경우 다음을 입력합니다. **MoviePipelineDeadlineCloudExecutorJob**
        iii. 기본 Job Settings 클래스의 경우 더하기 기호를 선택한 다음 를 입력합니다**DeadlineCloudRenderStepSetting**.

    이러한 설정을 사용하여 Deadline Cloud 플러그인을 선택할 수 Unreal Engine 있습니다.

팜을 사용하세요.

시작 지침을 모두 따랐다면 로컬 워크스테이션에서 팜으로 작업을 제출하고 해당 작업과 리소스를 모니터링하는 데 필요한 모든 것을 설정한 것입니다. 모든 종류의 작업 제출 또는 모니터링에 대한 자세한 내용은 아래 관련 항목을 참조하십시오.

- 작업
- 모니터 사용

데드라인 클라우드 모니터 사용

AWS 데드라인 클라우드 모니터는 시각적 컴퓨팅 작업에 대한 전반적인 보기를 제공합니다. 이를 사용하여 작업을 모니터링 및 관리하고, 차량 내 작업자 활동을 확인하고, 예산 및 사용량을 추적하고, 작업 결과를 다운로드할 수 있습니다.

각 대기열에는 작업, 단계 및 작업의 상태를 보여주는 작업 모니터가 있습니다. 모니터는 모니터에서 직접 작업을 관리하는 방법을 제공합니다. 우선 순위를 변경하고, 작업을 취소하고, 작업을 다시 요청할 수 있습니다.

Deadline Cloud Monitor에는 작업의 요약 상태를 보여주는 테이블이 있으며, 작업을 선택하여 작업 관련 문제를 해결하는 데 도움이 되는 세부 작업 로그를 볼 수 있습니다.

Deadline Cloud 모니터를 사용하면 작업이 생성될 때 지정된 워크스테이션의 위치로 결과를 다운로드할 수 있습니다.

Deadline Cloud 모니터는 사용량을 모니터링하고 비용을 관리하는 데도 도움이 됩니다. 자세한 정보는 데드라인 클라우드의 예산 및 사용량 관리을 참조하세요.

주제

- 데드라인 클라우드 모니터 URL을 공유하세요.
- 데드라인 클라우드 모니터를 엽니다.
- 데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다.
- 데드라인 클라우드에서 작업, 단계 및 작업을 보고 관리합니다.
- 데드라인 클라우드에서 작업 세부 정보 보기
- 데드라인 클라우드에서 단계 보기
- 데드라인 클라우드에서 작업 보기
- 데드라인 클라우드에서 로그 보기
- 데드라인 클라우드에서 완료된 결과를 다운로드하세요.

데드라인 클라우드 모니터 URL을 공유하세요.

데드라인 클라우드 서비스를 설정할 때 기본적으로 계정의 데드라인 클라우드 모니터를 여는 URL이 생성됩니다. 이 URL을 사용하여 브라우저 또는 데스크톱에서 모니터를 열 수 있습니다. 다른 사용자가 Deadline Cloud 모니터에 액세스할 수 있도록 URL을 공유하십시오.

사용자가 Deadline Cloud 모니터를 열 수 있으려면 먼저 사용자에게 액세스 권한을 부여해야 합니다. 액세스 권한을 부여하려면 모니터의 승인된 사용자 목록에 사용자를 추가하거나 모니터에 액세스할 수 있는 그룹에 사용자를 추가하십시오. 자세한 정보는 데드라인 클라우드에서의 사용자 관리을 참조하세요.

모니터 URL을 공유하려면

1. 데드라인 클라우드 콘솔을 엽니다.
2. 시작하기에서 데드라인 클라우드 대시보드로 이동을 선택합니다.
3. 탐색 창에서 대시보드를 선택합니다.
4. 계정 개요 섹션에서 계정 세부 정보를 선택합니다.
5. URL을 복사한 다음 Deadline Cloud 모니터에 액세스해야 하는 모든 사람에게 안전하게 보낼 수 있습니다.

데드라인 클라우드 모니터를 엽니다.

다음 방법 중 하나로 데드라인 클라우드 모니터를 열 수 있습니다.

- 콘솔 — 데드라인 클라우드 콘솔에 AWS Management Console 로그인하고 엽니다.
- 웹 — 데드라인 클라우드를 설정할 때 만든 모니터 URL로 이동합니다.
- 모니터링 - 데스크톱 데드라인 클라우드 모니터를 사용합니다.

콘솔을 사용할 때는 AWS Identity and Access Management ID를 사용하여 로그인한 다음 AWS IAM Identity Center 자격 증명을 AWS 사용하여 모니터에 로그인할 수 있어야 합니다. IAM Identity Center 자격 증명만 있는 경우 모니터 URL 또는 데스크톱 애플리케이션을 사용하여 로그인해야 합니다.

데드라인 클라우드 모니터 (웹) 를 열려면

1. 브라우저를 사용하여 데드라인 클라우드를 설정할 때 만든 모니터 URL을 엽니다.
2. 사용자 자격 증명으로 로그인합니다.

데드라인 클라우드 모니터를 열려면 (콘솔)

1. 데드라인 클라우드 콘솔을 엽니다.
2. 탐색 창에서 팜을 선택합니다.

3. 팜을 선택한 다음 작업 관리를 선택하여 Deadline Cloud 모니터 페이지를 엽니다.
4. 사용자 자격 증명으로 로그인하세요.

데드라인 클라우드 모니터 (데스크톱) 를 열려면

1. 데드라인 클라우드 콘솔을 엽니다.

   -또는-

   모니터 URL에서 데드라인 클라우드 모니터 - 웹을 엽니다.
2. • 데드라인 클라우드 콘솔에서 다음을 수행하십시오.
   1. 모니터에서 Deadline Cloud 대시보드로 이동을 선택한 다음 왼쪽 메뉴에서 다운로드를 선택합니다.
   2. 데드라인 클라우드 모니터에서 데스크톱용 모니터 버전을 선택합니다.
   3. 다운로드를 선택합니다.
   • 데드라인 클라우드 모니터 - 웹에서 다음을 수행하십시오.
   • 왼쪽 메뉴에서 워크스테이션 설정을 선택합니다. 워크스테이션 설정 항목이 보이지 않는 경우 화살표를 사용하여 왼쪽 메뉴를 여십시오.
   • 다운로드를 선택합니다.
   • OS 선택에서 운영 체제를 선택합니다.
3. 데드라인 클라우드 모니터 다운로드 - 데스크탑.
4. 모니터를 다운로드하여 설치한 후 컴퓨터에서 엽니다.

   • Deadline Cloud 모니터를 처음 여는 경우 모니터 URL을 제공하고 프로필 이름을 만들어야 합니다. 다음으로 Deadline Cloud 자격 증명을 사용하여 모니터에 로그인합니다.
   • 프로필을 만든 후 프로필을 선택하여 모니터를 엽니다. 데드라인 클라우드 자격 증명을 입력해야 할 수도 있습니다.

데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다.

Deadline Cloud 모니터를 사용하여 팜의 대기열 및 플릿 구성을 볼 수 있습니다. 또한 모니터를 사용하여 대기열에 있는 작업 또는 플릿에 있는 작업자 목록을 볼 수 있습니다.

대기열 및 플릿 세부 정보를 볼 수 있는 `VIEWING` 권한이 있어야 합니다. 세부 정보가 표시되지 않는 경우 관리자에게 문의하여 올바른 권한을 요청하십시오.

대기열 세부 정보를 보려면

1. 데드라인 클라우드 모니터를 엽니다..
2. 팜 목록에서 원하는 대기열이 포함된 팜을 선택합니다.
3. 대기열 목록에서 세부 정보를 표시할 대기열을 선택합니다. 둘 이상의 대기열 구성을 비교하려면 확인란을 두 개 이상 선택합니다.
4. 대기열에 있는 작업 목록을 보려면 대기열 목록이나 세부 정보 패널에서 대기열 이름을 선택합니다.

모니터가 이미 열려 있는 경우 왼쪽 탐색 창의 큐 목록에서 큐를 선택할 수 있습니다.

플릿 세부 정보를 보려면

1. 데드라인 클라우드 모니터를 엽니다..
2. 농장 목록에서 관심 있는 플릿이 포함된 농장을 선택합니다.
3. 팜 리소스에서 플릿을 선택합니다.
4. 플릿 목록에서 플릿의 세부 정보를 표시할 플릿을 선택합니다. 둘 이상의 플릿 구성을 비교하려면 확인란을 두 개 이상 선택합니다.
5. 플릿 내 근로자 목록을 보려면 플릿 목록 또는 세부 정보 패널에서 플릿 이름을 선택합니다.

모니터가 이미 열려 있는 경우 왼쪽 탐색 창의 플릿 목록에서 플릿을 선택할 수 있습니다.

데드라인 클라우드에서 작업, 단계 및 작업을 보고 관리합니다.

대기열을 선택하면 Deadline Cloud 모니터의 작업 모니터 섹션에 해당 대기열의 작업, 작업의 단계 및 각 단계의 작업이 표시됩니다. 작업, 단계 또는 작업을 선택하면 작업 메뉴를 사용하여 각 작업을 관리할 수 있습니다.

작업 모니터를 열려면 단계에 따라 대기열을 확인한 다음 작업할 작업, 단계 또는 작업을 선택합니다. 데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다.

작업, 단계 및 작업의 경우 다음을 수행할 수 있습니다.

- 상태를 [대기중], [성공], [실패] 또는 [취소됨] 으로 변경합니다.
- 작업, 단계 또는 작업에서 처리된 출력을 다운로드합니다.
- 작업, 단계 또는 작업의 ID를 복사합니다.

선택한 작업에 대해 다음을 수행할 수 있습니다.

- 작업을 아카이브합니다.
- 우선 순위 변경 또는 단계별 종속성 보기 등 작업 속성을 수정합니다.
- 작업 매개변수를 사용하여 추가 세부 정보를 볼 수 있습니다.

자세한 내용은 데드라인 클라우드에서 작업 세부 정보 보기 단원을 참조하세요.

각 단계에서 다음을 수행할 수 있습니다.

- 단계의 종속성을 확인하세요. 단계가 실행되기 전에 단계에 대한 종속성을 완료해야 합니다.

자세한 내용은 데드라인 클라우드에서 단계 보기단원을 참조하세요.

각 작업에 대해 다음을 수행할 수 있습니다.

- 작업에 대한 로그를 볼 수 있습니다.
- 작업 매개변수 보기

자세한 정보는 데드라인 클라우드에서 작업 보기을 참조하세요.

데드라인 클라우드에서 작업 세부 정보 보기

데드라인 클라우드 모니터의 Job monitor 페이지는 다음과 같은 기능을 제공합니다.

- 작업 진행 상황을 전체적으로 볼 수 있습니다.
- 작업을 구성하는 단계 및 작업 보기

목록에서 작업을 선택하여 해당 작업의 단계 목록을 확인한 다음 단계 목록에서 단계를 선택하여 작업에 대한 작업을 확인합니다. 항목을 선택한 후 해당 항목의 작업 메뉴를 사용하여 세부 정보를 볼 수 있습니다.

작업 세부 정보를 보려면

1. 단계에 따라 대기열을 볼 수 있습니다데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다..
2. 탐색 창에서 작업을 제출한 대기열을 선택합니다.

3. 다음 방법 중 하나를 사용하여 작업을 선택합니다.

    a. 작업 목록에서 작업을 선택하여 세부 정보를 확인합니다.

    b. 검색 필드에 작업과 관련된 텍스트 (예: 작업 이름 또는 작업을 생성한 사용자) 를 입력합니다. 표시되는 결과에서 보려는 작업을 선택합니다.

작업 세부 정보에는 작업의 단계와 각 단계의 작업이 포함됩니다. 작업 메뉴를 사용하여 다음 작업을 수행할 수 있습니다.

- 작업 상태를 변경합니다.
- 작업 속성 보기 및 수정 작업의 단계 간 종속성을 보고 작업의 우선 순위를 변경할 수 있습니다. 일반적으로 우선 순위가 높은 작업은 더 빨리 완료됩니다.
- 작업이 제출될 때 설정된 작업의 매개 변수를 볼 수 있습니다.
- 작업 출력을 다운로드합니다. 작업 출력을 다운로드하면 작업의 단계 및 작업에서 생성된 모든 출력이 포함됩니다.

데드라인 클라우드에서 단계 보기

AWS 데드라인 클라우드 모니터를 사용하여 처리 작업의 단계를 볼 수 있습니다. Job Monitor의 단계 목록에는 선택한 작업을 구성하는 단계 목록이 표시됩니다. 단계를 선택하면 작업 목록에 해당 단계의 작업이 표시됩니다.

단계를 보려면

1. 에 나와 있는 데드라인 클라우드에서 작업 세부 정보 보기 단계에 따라 작업 목록을 확인하십시오.
2. 작업 목록에서 작업을 선택합니다.
3. 단계 목록에서 단계를 선택합니다.

작업 메뉴를 사용하여 다음 작업을 수행할 수 있습니다.

- 단계 상태를 변경합니다.
- 단계 출력을 다운로드합니다. 단계의 출력을 다운로드하면 해당 단계의 작업에서 생성된 모든 출력이 포함됩니다.
- 단계의 종속성을 볼 수 있습니다. 종속성 테이블에는 선택한 단계가 시작되기 전에 완료해야 하는 단계 목록과 이 단계가 완료될 때까지 대기 중인 단계 목록이 표시됩니다.

데드라인 클라우드에서 작업 보기

AWS 데드라인 클라우드 모니터를 사용하여 처리 작업의 작업을 볼 수 있습니다. Job Monitor의 작업 목록에는 단계 목록에서 선택한 단계를 구성하는 작업이 표시됩니다.

작업을 보려면

1. 작업 목록을 데드라인 클라우드에서 작업 세부 정보 보기 보려면 다음 단계를 따르세요.
2. 작업 목록에서 작업을 선택합니다.
3. 단계 목록에서 단계를 선택합니다.
4. 작업 목록에서 작업을 선택합니다.

작업 메뉴를 사용하여 다음 작업을 수행할 수 있습니다.

- 작업 상태를 변경합니다.
- 작업 로그 보기 자세한 정보는 데드라인 클라우드에서 로그 보기을 참조하세요.
- 태스크가 생성될 때 설정된 파라미터를 확인하세요.
- 작업 출력을 다운로드합니다. 작업 출력을 다운로드하면 선택한 작업에서 생성된 출력만 포함됩니다.

데드라인 클라우드에서 로그 보기

로그는 작업 상태 및 처리에 대한 자세한 정보를 제공합니다. AWS Deadline Cloud 모니터에서는 다음과 같은 두 가지 유형의 로그를 볼 수 있습니다.

- 세션 로그에는 다음을 포함하여 작업 타임라인이 자세히 설명되어 있습니다.
  - 설치 작업 (예: 첨부 파일 동기화 및 소프트웨어 환경 로드)
  - 작업 또는 작업 세트 실행
  - 종료 조치 (예: 작업자의 환경 종료)

  세션에는 하나 이상의 작업 처리가 포함되며 여러 작업이 포함될 수 있습니다. 세션 로그에는 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스 유형, vCPU 및 메모리에 대한 정보도 표시됩니다. 세션 로그에는 세션에서 사용된 작업자의 로그로 연결되는 링크도 포함됩니다.
- 작업자 로그는 작업자가 수명 주기 동안 처리하는 작업의 일정에 대한 세부 정보를 제공합니다. 작업자 로그에는 여러 세션에 대한 정보가 포함될 수 있습니다.

세션 및 작업자 로그를 다운로드하여 오프라인에서 검사할 수 있습니다.

세션 로그를 보려면

1. 작업 목록을 데드라인 클라우드에서 작업 세부 정보 보기 보려면 의 단계를 따르세요.
2. 작업 목록에서 작업을 선택합니다.
3. 단계 목록에서 단계를 선택합니다.
4. 작업 목록에서 작업을 선택합니다.
5. 작업 메뉴에서 로그 보기를 선택합니다.

타임라인 섹션에는 작업에 대한 작업 요약이 표시됩니다. 세션에서 실행되는 더 많은 작업을 확인하고 세션의 종료 동작을 보려면 모든 작업에 대한 로그 보기를 선택합니다.

작업의 작업자 로그를 보려면

1. 작업 목록을 데드라인 클라우드에서 작업 세부 정보 보기 보려면 다음 단계를 따르세요.
2. 작업 목록에서 작업을 선택합니다.
3. 단계 목록에서 단계를 선택합니다.
4. 작업 목록에서 작업을 선택합니다.
5. 작업 메뉴에서 로그 보기를 선택합니다.
6. 세션 정보를 선택합니다.
7. 작업자 로그 보기를 선택합니다.

플릿 세부 정보에서 작업자 로그를 보려면

1. 플릿을 데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다. 보려면 다음 단계를 따르세요.
2. 작업자 목록에서 작업자 ID를 선택합니다.
3. 작업 메뉴에서 작업자 로그 보기를 선택합니다.

데드라인 클라우드에서 완료된 결과를 다운로드하세요.

작업이 끝나면 AWS Deadline Cloud 모니터를 사용하여 결과를 워크스테이션에 다운로드할 수 있습니다. 출력 파일은 작업을 생성할 때 지정한 이름 및 위치와 함께 저장됩니다.

출력 파일은 무기한 저장됩니다. 스토리지 비용을 줄이려면 대기열의 Amazon S3 버킷에 대한 S3 수명 주기 구성을 생성하는 것이 좋습니다. 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 스토리지 수명 주기 관리를 참조하십시오.

작업, 단계 또는 작업의 완성된 출력을 다운로드하려면

1. 작업 목록을 데드라인 클라우드에서 작업 세부 정보 보기 보려면 다음 단계를 따르십시오.
2. 출력을 다운로드할 작업, 단계 또는 작업을 선택합니다.
   - 작업을 선택하면 해당 작업의 모든 단계에 있는 모든 작업에 대한 출력을 모두 다운로드할 수 있습니다.
   - 단계를 선택하면 해당 단계의 모든 작업에 대한 출력을 모두 다운로드할 수 있습니다.
   - 작업을 선택하면 해당 개별 작업의 출력을 다운로드할 수 있습니다.
3. 작업 메뉴에서 결과 다운로드를 선택합니다.
4. 작업이 제출되었을 때 설정된 위치로 출력이 다운로드됩니다.

> **Note**
>
> 메뉴를 사용한 출력 다운로드는 현재 Windows 및 버전에서만 지원됩니다Linux. 가 Mac 있고 다운로드 출력 메뉴 항목을 선택하면 렌더링된 출력을 다운로드하는 데 사용할 수 있는 AWS CLI 명령이 창에 표시됩니다.

데드라인 클라우드 팜

팜은 작업을 관리하는 대기열과 작업을 수행하는 컴퓨팅 리소스를 저장하는 컨테이너입니다.

주제

- 팜 생성
- 팜 삭제
- 팜 편집

팜 생성

1. 데드라인 클라우드 콘솔에서 대시보드로 이동을 선택합니다.
2. 데드라인 클라우드 대시보드의 팜 섹션에서 작업 → 팜 생성을 선택합니다.
   - 또는 왼쪽 패널에서 팜 및 기타 리소스를 선택한 다음 팜 만들기를 선택합니다.
3. 팜 이름을 추가합니다.
4. 설명에 팜 설명을 입력합니다. 설명이 명확하면 팜의 용도를 빠르게 파악할 수 있습니다.
5. (선택 사항) 기본적으로 데이터는 보안을 위해 AWS 소유하고 관리하는 키로 암호화됩니다. 암호화 설정 사용자 지정 (고급) 을 선택하여 기존 키를 사용하거나 관리하는 새 키를 만들 수 있습니다.

   확인란을 사용하여 암호화 설정을 사용자 지정하려면 AWS KMS ARN을 입력하거나 새 KMS 키 생성을 선택하여 AWS KMS 새 ARN을 생성합니다.
6. (선택 사항) 새 태그 추가를 선택하여 팜에 하나 이상의 태그를 추가합니다.
7. 팜 생성을 선택합니다. 생성 후에는 팜이 표시됩니다.

팜 삭제

1. 데드라인 클라우드 대시보드에서 팜 및 기타 리소스를 선택합니다.
2. 팜 목록에서 삭제하려는 팜을 선택한 다음 삭제를 선택합니다.

팜 편집

1. 데드라인 클라우드 대시보드에서 팜 및 기타 리소스를 선택합니다.
2. 팜 목록에서 삭제하려는 팜을 선택한 다음 편집을 선택합니다.
3. 표시되는 편집 창에서 팜 이름이나 설명을 변경한 다음 변경 사항 저장을 선택합니다.

데드라인 클라우드 대기열

대기열은 작업을 관리하고 처리하는 팜 리소스입니다.

대기열을 사용하려면 모니터와 팜이 이미 설정되어 있어야 합니다.

주제

- 대기열 생성
- 대기열 환경 만들기
- 대기열 삭제
- 대기열 편집
- 대기열과 플릿을 연결합니다.

대기열 생성

1. Deadline Cloud 콘솔 대시보드에서 대기열을 만들려는 팜을 선택합니다.
   - 또는 왼쪽 패널에서 팜 및 기타 리소스를 선택한 다음 대기열을 만들려는 팜을 선택합니다.
2. 큐 탭에서 큐 생성을 선택합니다.
3. 대기열 이름을 입력합니다.
4. 설명에 대기열 설명을 입력합니다. 설명은 대기열의 용도를 식별하는 데 도움이 됩니다.
5. 작업 첨부의 경우 새 Amazon S3 버킷을 만들거나 기존 Amazon S3 버킷을 선택할 수 있습니다.
   a. 새 Amazon S3 버킷을 만들려면
      i. 새 작업 버킷 생성을 선택합니다.
      ii. 버킷 이름을 입력합니다. 버킷 `deadlinecloud-job-attachments-[MONITORNAME]` 이름을 지정하는 것이 좋습니다.
      iii. Root 접두사를 입력하여 대기열의 루트 위치를 정의하거나 변경합니다.
   b. 기존 Amazon S3 버킷을 선택하려면
      i. 기존 S3 버킷 선택 > S3 찾아보기를 선택합니다.
      ii. 사용 가능한 버킷 목록에서 대기열에 사용할 S3 버킷을 선택합니다.
6. (선택 사항) 대기열을 고객 관리형 플릿과 연결하려면 고객 관리 플릿과의 연결 활성화를 선택합니다.

7. 고객 관리 플릿과의 연결을 활성화하는 경우 다음 단계를 완료해야 합니다.

> **Important**
>
> 실행 기능을 사용할 사용자 및 그룹을 지정하는 것이 좋습니다. 그렇지 않으면 작업자의 에이전트가 수행할 수 있는 모든 작업을 해당 작업으로 수행할 수 있기 때문에 팜의 보안 상태가 저하됩니다. 잠재적 보안 위험에 대한 자세한 내용은 사용자 및 그룹으로 작업 실행을 참조하십시오.

a. 사용자로 실행의 경우:

   대기열 작업에 대한 자격 증명을 제공하려면 대기열 구성 사용자를 선택합니다.

   또는 자신의 자격 증명을 설정하지 않고 작업자 에이전트 사용자로 작업을 실행하지 않으려면 작업자 에이전트 사용자를 선택합니다.

b. (선택 사항) 사용자 자격 증명으로 실행 자격 증명의 경우 큐 작업에 대한 자격 증명을 제공할 사용자 이름과 그룹 이름을 입력합니다.

   Windows플릿을 사용하는 경우 Run as 사용자의 AWS Secrets Manager 비밀번호가 포함된 비밀번호를 생성해야 합니다. 다음 지침에 따라 비밀번호를 생성하십시오. *jobuser#* 의 이름으로 바꾸십시오. `jobRunAsUser`

   i. 관리자 권한으로 PowerShell 또는 명령 프롬프트를 엽니다.

   ii. 사용자를 생성합니다.

   ```
   net user jobuser /add
   ```

   iii. 비밀번호를 설정합니다.

   ```
   net user jobuser *
   ```

   iv. 사용자의 로컬 프로필과 홈 디렉터리를 생성합니다. 다음 명령을 실행하고 암호를 묻는 메시지가 표시되면 사용자의 암호를 입력합니다.

   ```
   runas /profile /user:jobuser "cmd.exe /C"
   ```

8. 예산을 요구하면 대기열 비용을 관리하는 데 도움이 됩니다. 예산 필요 없음 또는 예산 필요를 선택합니다.

9. 대기열에는 사용자를 대신하여 Amazon S3에 액세스할 수 있는 권한이 필요합니다. 새 서비스 역할을 생성하거나 기존 서비스 역할을 사용할 수 있습니다. 기존 서비스 역할이 없는 경우 새 서비스 역할을 만들어 사용하세요.

    a. 기존 서비스 역할을 사용하려면 서비스 역할 선택을 선택한 다음 드롭다운에서 역할을 선택합니다.

    b. 새 서비스 역할을 만들려면 새 서비스 역할 만들기 및 사용을 선택한 다음 역할 이름과 설명을 입력합니다.

10. (선택 사항) 큐 환경에 환경 변수를 추가하려면 새 환경 변수 추가를 선택한 다음 추가하는 각 변수의 이름과 값을 입력합니다.

11. (선택 사항) 대기열에 하나 이상의 태그를 추가하려면 새 태그 추가를 선택합니다.

12. 기본 Conda 대기열 환경을 만들려면 확인란을 선택한 상태로 유지하십시오. 대기열 환경에 대한 자세한 내용은 대기열 환경 만들기를 참조하십시오. 고객 관리형 플릿에 대한 대기열을 생성하는 경우 확인란의 선택을 취소하십시오.

13. 대기열 생성을 선택합니다.

대기열 환경 만들기

대기열 환경은 플릿 작업자를 설정하는 일련의 환경 변수 및 명령입니다. 대기열 환경을 사용하여 소프트웨어 응용 프로그램, 환경 변수 및 기타 리소스를 대기열의 작업에 제공할 수 있습니다.

대기열을 만들 때 기본 Conda 대기열 환경을 만들 수 있습니다. 이 환경에서는 서비스 관리 플릿이 파트너 DCC 응용 프로그램 및 렌더러를 위한 패키지에 액세스할 수 있습니다. 자세한 정보는 기본 Conda 대기열 환경을 참조하세요.

콘솔을 사용하거나 json 또는 YAML 템플릿을 직접 편집하여 대기열 환경을 추가할 수 있습니다. 이 절차에서는 콘솔을 사용하여 환경을 만드는 방법을 설명합니다.

1. 대기열에 대기열 환경을 추가하려면 대기열로 이동하여 대기열 환경 탭을 선택합니다.
2. 작업을 선택한 다음 양식을 사용하여 새로 만들기를 선택합니다.
3. 대기열 환경의 이름과 설명을 입력합니다.
4. 새 환경 변수 추가를 선택한 다음 추가하는 각 변수의 이름과 값을 입력합니다.
5. (선택 사항) 큐 환경의 우선 순위를 입력합니다. 우선 순위는 이 대기열 환경이 작업자에서 실행되는 순서를 나타냅니다. 우선 순위가 높은 대기열 환경이 먼저 실행됩니다.

6. 대기열 환경 만들기를 선택합니다.

기본 Conda 대기열 환경

서비스 관리형 플릿과 관련된 대기열을 생성할 때 작업을 Conda위해 가상 환경에서 패키지를 다운로드하고 설치할 수 있도록 지원하는 기본 대기열 환경을 추가할 수 있습니다.

Conda채널의 패키지를 제공합니다. 채널은 패키지가 저장되는 위치입니다. Deadline Cloud는 파트너 DCC 애플리케이션 및 렌더러를 지원하는 패키지를 호스팅하는 채널을 제공합니다. `deadline-cloud` 패키지는 다음과 같습니다.

- 블렌더
  - `blender=3.6`
  - `blender-openjd`
- 후디니
  - `houdini=19.5`
  - `houdini-openjd`
- Maya
  - `maya=2024`
  - `maya-mtoa=2024.5.3`
  - `maya-openjd`
- 핵무기
  - `nuke=15`
  - `nuke-openjd`

기본 Conda 환경의 대기열에 작업을 제출하면 환경이 작업에 두 개의 매개 변수를 추가합니다. 이 매개 변수는 작업이 처리되기 전에 작업 환경을 구성하는 데 사용할 Conda 패키지와 채널을 지정합니다. 매개변수는 다음과 같습니다.

- `CondaPackages`— 공백으로 구분된 패키지 매칭 사양 목록 (예: 또는). `blender=3.6 numpy>1.22` 가상 환경 생성을 건너뛰기 위한 기본값은 비어 있습니다.
- `CondaChannels`— 공백으로 구분된 Conda채널 목록 (예: `deadline-cloudconda-forge`, 또는`s3://DOC-EXAMPLE-BUCKET/conda/channel`) 기본값은 파트너 DCC 애플리케이션 `deadline-cloud` 및 렌더러를 제공하는 서비스 관리 플릿에서 사용할 수 있는 채널입니다.

통합 제출자를 사용하여 DCC에서 Deadline Cloud로 작업을 보내면 제출자가 DCC 애플리케이션 및 제출자를 기반으로 매개변수 값을 채웁니다. `CondaPackages` 예를 들어 블렌더를 사용하는 경우 파라미터는 로 설정됩니다. `CondaPackage blender=3.6.* blender-openjd=0.4.*`

대기열 삭제

> **Warning**
>
> 대기열을 삭제하면 대기열에 있는 작업을 복구할 수 없습니다. 대기열을 삭제하면 해당 대기열에 있는 작업도 삭제됩니다.

1. Deadline Cloud 대시보드에서 팜 및 기타 리소스를 선택합니다.
2. 팜 목록에서 삭제할 대기열이 있는 팜을 선택합니다.
3. 큐를 선택한 다음 삭제를 선택합니다.
4. 확인 창에서 삭제를 선택합니다. 큐와 큐에 있는 모든 작업이 삭제됩니다.

대기열 편집

1. Deadline Cloud 대시보드에서 팜 및 기타 리소스를 선택합니다.
2. 팜 목록에서 편집할 대기열이 있는 팜을 선택합니다.
3. 큐를 선택한 다음 편집을 선택합니다.
4. 이름, 설명, 예산 요구 사항, 사용자 권한으로 실행 옵션, 할당된 서비스 역할을 편집할 수 있습니다. 기존 플릿을 대기열에 연결할 수도 있습니다.
5. 변경 사항 저장을 선택합니다.

대기열과 플릿을 연결합니다.

1. 플릿과 연결할 대기열을 선택합니다.
2. 대기열에 연결할 플릿을 선택하려면 Associate Fleets를 선택합니다.
3. 플릿 선택 드롭다운을 선택합니다. 사용 가능한 플릿 목록이 표시됩니다.
4. 사용 가능한 플릿 목록에서 대기열에 연결할 플릿 또는 플릿 옆의 확인란을 선택합니다.
5. 연결을 선택합니다. 이제 플릿 연결 상태가 Associate로 표시되어야 합니다.

데드라인 클라우드 플릿 관리

이 섹션에서는 Deadline Cloud의 서비스 관리형 플릿 (SMF) 및 고객 관리형 플릿 (CMF) 을 관리하는 방법을 설명합니다.

두 가지 유형의 데드라인 클라우드 플릿을 설정할 수 있습니다.

- 서비스 관리 플릿은 이 서비스에서 제공하는 기본 설정인 Deadline Cloud를 사용하는 워커 플릿입니다. 이러한 기본 설정은 효율적이고 비용 효율적이도록 설계되었습니다.
- 고객 관리형 차량 (CMF) 은 관리하는 작업자 집합입니다. CMF는 AWS 인프라 내부, 온프레미스 또는 코로케이션된 데이터 센터에 위치할 수 있습니다. CMF는 차량에 대한 완전한 통제와 책임을 제공합니다. 여기에는 차량 내 작업자의 공급, 운영, 관리 및 해체 작업이 포함됩니다.

주제

- Deadline Cloud 서비스가 관리하는 차량 관리
- 데드라인 클라우드 고객 관리 플릿 관리

Deadline Cloud 서비스가 관리하는 차량 관리

서비스 관리 플릿은 Deadline Cloud에서 제공하는 기본 설정이 있는 작업자 집합입니다. 이러한 기본 설정은 효율적이고 비용 효율적이도록 설계되었습니다.

1. 서비스 관리형 플릿 (SMF) 을 생성하려면 플릿을 생성하려는 팜으로 이동하십시오.
2. 플릿 탭을 선택합니다.
3. 플릿 생성을 선택합니다.
4. 플릿 이름을 입력합니다.
5. 설명(Description)을 입력합니다. 설명이 명확하면 플릿의 용도를 빠르게 파악할 수 있습니다.
6. 서비스 관리형 플릿 유형을 선택합니다.
7. 플릿에 맞는 스팟 또는 온디맨드 인스턴스 마켓 옵션을 선택하십시오. 스팟 인스턴스는 할인된 가격으로 사용할 수 있는 예약되지 않은 용량이지만 온디맨드 요청으로 인해 중단될 수 있습니다. 온디맨드 인스턴스는 초 단위로 요금이 부과되지만 장기 약정이 없으며 중단되지 않습니다. 기본적으로 플릿은 스팟 인스턴스를 사용합니다.

8. 선택 사항: 대기열에 있는 작업에 용량을 사용할 수 있도록 플릿을 확장할 최대 인스턴스 수를 설정합니다. 대기중인 작업이 없을 때 플릿이 모든 인스턴스를 **0** 해제할 수 있도록 최소 인스턴스 수를 로 유지하는 것이 좋습니다.
9. 플릿에 대한 서비스 액세스를 위해서는 기존 역할을 선택하거나 새 역할을 생성하십시오. 서비스 역할은 플릿의 인스턴스에 자격 증명을 제공하여 작업을 처리할 권한을 부여하고 모니터에 있는 사용자에게 로그 정보를 읽을 수 있도록 자격 증명을 제공합니다.
10. 다음을 선택합니다.
11. 플릿에 필요한 최소 및 최대 vCPU를 입력합니다.
12. 플릿에 필요한 최소 및 최대 메모리를 입력합니다.
13. 선택 사항 플릿에서 특정 인스턴스 유형을 허용하거나 제외하여 해당 인스턴스 유형만 이 플릿에 사용되도록 할 수 있습니다.
14. 선택 사항 이 플릿의 작업자에게 연결할 Amazon Elastic Block Store (Amazon EBS) gp3 볼륨의 크기를 지정할 수 있습니다. 자세한 내용은 EBS 사용 설명서를 참조하십시오.
15. 다음을 선택합니다.
16. 선택 사항 이 플릿의 기능을 정의하는 사용자 지정 작업자 요구 사항을 정의하여 작업 제출 시 지정된 사용자 지정 호스트 요구 사항과 결합할 수 있습니다. 플릿을 자체 라이선스 서버에 연결하려는 경우 특정 라이선스 유형을 예로 들 수 있습니다.
17. 다음을 선택합니다.
18. 선택 사항 플릿을 대기열에 연결하려면 드롭다운에서 대기열을 선택합니다. 대기열이 기본 Conda 대기열 환경으로 설정된 경우 플릿에는 파트너 DCC 애플리케이션 및 렌더러를 지원하는 패키지가 자동으로 제공됩니다. 제공된 패키지 목록은 을 참조하십시오. 기본 Conda 대기열 환경
19. 다음을 선택합니다.
20. 선택 사항 플릿에 태그를 추가하려면 Add new tag를 선택한 다음 해당 태그의 키와 값을 입력합니다.
21. 다음을 선택합니다.
22. 플릿 설정을 검토한 다음 플릿 생성을 선택합니다. 플릿을 생성하고 나면 플릿이 표시됩니다.

VFX Reference Platform 호환성

VFX 업계의 공통 대상 VFX Reference Platform 플랫폼입니다. Amazon Linux 2023을 실행하는 표준 서비스 관리형 플릿 Amazon EC2 인스턴스를 지원하는 VFX Reference Platform 소프트웨어와 함께 사용하려면 서비스 관리형 플릿을 사용할 때 다음 고려 사항을 염두에 두어야 합니다.

매년 업데이트됩니다. VFX Reference Platform 데드라인 클라우드 서비스 관리 플릿을 포함한 AL2023 사용에 대한 이러한 고려 사항은 2022~2024년 참조 플랫폼 달력 연도 (CY) 를 기준으로 합니다. 자세한 정보는 VFX Reference Platform을 참조하세요.

> **Note**
>
> 고객 관리형 플릿에 대한 사용자 지정 Amazon Machine Image (AMI) 을 생성하는 경우 Amazon EC2 인스턴스를 준비할 때 이러한 요구 사항을 추가할 수 있습니다.

AL2023 Amazon EC2 인스턴스에서 VFX Reference Platform 지원되는 소프트웨어를 사용하려면 다음 사항을 고려하십시오.

- AL2023 버전과 함께 설치된 glibc 버전은 런타임 사용에는 호환되지만 VFX Reference Platform CY2024 이전 버전과 호환되는 소프트웨어를 빌드하는 데는 호환되지 않습니다.
- Python 3.9 및 3.11은 서비스 매니지드 플릿과 함께 제공되므로 CY2022 및 CY2024 호환됩니다. VFX Reference Platform Python 3.7과 3.10은 서비스 관리 플릿에서 제공되지 않습니다. 이를 필요로 하는 소프트웨어는 대기열 또는 작업 환경에 Python 설치를 제공해야 합니다.
- 서비스 관리 플릿에서 제공되는 일부 Boost 라이브러리 구성 요소는 버전 1.75이며, 이 버전은 과 호환되지 않습니다. VFX Reference Platform 애플리케이션에서 Boost를 사용하는 경우 호환성을 위해 자체 라이브러리 버전을 제공해야 합니다.
- 인텔 TBB 업데이트 3은 서비스 관리형 제품군에서 제공됩니다. 이 제품은 VFX Reference Platform CY2022, CY2023 및 CY2024 과 호환됩니다.
- 에서 지정한 버전을 가진 다른 라이브러리는 서비스 관리형 플릿에서 제공하지 않습니다. VFX Reference Platform 서비스 관리 플릿에서 사용되는 모든 애플리케이션을 라이브러리에 제공해야 합니다. 라이브러리 목록은 참조 플랫폼을 참조하십시오.

데드라인 클라우드 고객 관리 플릿 관리

이 섹션에서는 Deadline Cloud의 고객 관리형 차량 (CMF) 을 관리하는 방법을 설명합니다.

CMF는 관리하는 작업자 집합입니다. CMF는 AWS 인프라 내부, 온프레미스 또는 코로케이션된 데이터 센터에 위치할 수 있습니다. CMF는 차량에 대한 완전한 통제와 책임을 제공합니다. 여기에는 차량 내 작업자의 공급, 운영, 관리 및 해체 작업이 포함됩니다.

주제

- 고객 관리형 플릿 생성
- 작업자 호스트 설정 및 구성
- Windows 작업 사용자 비밀에 대한 액세스 관리
- 작업에 필요한 소프트웨어 설치 및 구성
- AWS 자격 증명 구성
- Amazon Machine Image 생성
- Amazon EC2 Auto Scaling 그룹으로 플릿 인프라 생성
- 고객 관리 플릿을 라이선스 엔드포인트에 연결

고객 관리형 플릿 생성

고객 관리형 차량 (CMF) 을 만들려면 다음 단계를 완료하세요.

Deadline Cloud console

Deadline Cloud 콘솔을 사용하여 고객 관리형 플릿을 만들려면

1. 데드라인 클라우드 콘솔을 엽니다.
2. 팜을 선택합니다. 사용 가능한 팜 목록이 표시됩니다.
3. 작업하려는 팜의 이름을 선택합니다.
4. 플릿 탭을 선택합니다.
5. 플릿 생성을 선택합니다.
6. 플릿 이름을 입력합니다.
7. (선택 사항) 플릿에 대한 설명을 입력합니다.
8. 플릿 유형에서 고객 관리를 선택합니다.
9. Auto Scaling 유형을 선택합니다. 자세한 내용은 Auto Scaling 이벤트 처리를 위한 사용을 EventBridge 참조하십시오.
    - 스케일링 없음: 온프레미스 플릿을 만들고 있는데 Deadline Cloud Auto Scaling을 옵트아웃하고 싶은 경우
    - 규모 조정 권장 사항: Amazon Elastic Compute Cloud (Amazon EC2) 플릿을 만들고 있습니다.
10. 플릿의 서비스 액세스를 선택합니다.

a. 권한을 더 세부적으로 제어하려면 각 플릿에 대해 새 서비스 역할 생성 및 사용 옵션을 사용하는 것이 좋습니다. 이 옵션은 기본적으로 설정되어 있습니다.

b. 서비스 역할 선택을 선택하여 기존 서비스 역할을 사용할 수도 있습니다.

11. 선택 내용을 검토한 후 다음을 선택합니다.
12. 플릿에 맞는 운영 체제를 선택하세요. 플릿의 모든 작업자는 공통 운영 체제를 사용해야 합니다.
13. 호스트 CPU 아키텍처를 선택합니다.
14. 이 플릿의 작업자 호스트에 대한 다음 하드웨어 요구 사항을 선택합니다.

a. 플릿의 워크로드 수요를 충족하는 최소 및 최대 vCPU 및 메모리 하드웨어 요구 사항을 선택합니다.

b. (선택 사항) GPU 요구 사항을 선택한 다음 최소 및 최대 GPU를 입력합니다.

15. 선택 항목을 검토한 후 다음을 선택합니다.
16. (선택 사항) 사용자 지정 작업자 요구 사항을 정의합니다.
17. 드롭다운을 사용하여 플릿과 연결할 대기열을 하나 이상 선택합니다.

> **Note**
>
> 모두 동일한 신뢰 경계에 있는 대기열에만 플릿을 연결하는 것이 좋습니다. 이렇게 하면 동일한 작업자에서 실행 중인 작업 간에 강력한 보안 경계가 보장됩니다.

18. 대기열 연결을 검토한 후 다음을 선택합니다.
19. (선택 사항) 기본 Conda 대기열 환경의 경우 작업에서 요청한 Conda 패키지를 설치할 대기열 환경을 생성합니다.

> **Note**
>
> Conda 대기열 환경은 작업에서 요청한 Conda 패키지를 설치하는 데 사용됩니다. CMF에는 필수 Conda 명령이 기본적으로 설치되어 있지 않으므로 일반적으로 CMF와 관련된 대기열에서는 Conda 대기열 환경을 선택 취소해야 합니다.

20. (선택 사항) CMF에 태그를 추가합니다. 자세한 내용은 리소스 태그 지정을 참조하십시오. AWS
21. 플릿 구성을 검토하고 변경하십시오.

22. 플릿 생성을 선택합니다.
23. 플릿 탭을 선택한 다음 플릿 ID를 기록해 둡니다.

AWS CLI

를 사용하여 고객 관리형 플릿을 AWS CLI 만들려면

1. 를 엽니다. AWS CLI
2. 편집 `fleet-trust-policy.json`.

    a. ##### 표시된 텍스트를 AWS 계정 ID 및 데드라인 클라우드 팜 ID로 대체하여 다음 IAM 정책을 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "credentials.deadline.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "ACCOUNT_ID"
                },
                "ArnEquals": {
                    "aws:SourceArn":
 "arn:aws:deadline:*:ACCOUNT_ID:farm/FARM_ID"
                }
            }
        }
    ]
}
```

    b. 변경 내용을 저장합니다.
3. 편집 `create-cmf-fleet.json`.

    a. 다음 IAM 정책을 추가하세요.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "deadline:AssumeFleetRoleForWorker",
                "deadline:UpdateWorker",
                "deadline:DeleteWorker",
                "deadline:UpdateWorkerSchedule",
                "deadline:BatchGetJobEntity",
                "deadline:AssumeQueueRoleForWorker"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": "${aws:ResourceAccount}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": "${aws:ResourceAccount}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*:/aws/deadline/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": "${aws:ResourceAccount}"
                }
            }
        }
```

```
    ]
}
```

b. 변경 내용을 저장합니다.

4. 플릿의 작업자가 사용할 IAM 역할을 추가합니다.

```
aws iam create-role --role-name FleetWorkerRoleName --assume-role-policy-
document file://fleet-trust-policy.json
aws iam put-role-policy --role-name FleetWorkerRoleName --policy-name
 FleetWorkerPolicy --policy-document file://fleet-policy.json
```

5. 편집 `create-fleet-request.json`.

a. 기울임꼴로 표시된 텍스트를 CMF 값으로 대체하여 다음 IAM 정책을 추가합니다.

> **Note**
>
> *ROLE_ARN# ## ## # ####.* `create-cmf-fleet.json`
> *OS_FAMILY#* 경우, 또는 중 하나를 선택해야 합니다. `linux` `macos` `windows`

```
{
    "farmId": "FARM_ID",
    "displayName": "FLEET_NAME",
    "description": "FLEET_DESCRIPTION",
    "roleArn": "ROLE_ARN",
    "minWorkerCount": 0,
    "maxWorkerCount": 10,
    "configuration": {
        "customerManaged": {
            "mode": "NO_SCALING",
            "workerCapabilities": {
                "vCpuCount": {
                    "min": 1,
                    "max": 4
                },
                "memoryMiB": {
                    "min": 1024,
                    "max": 4096
                },
                "osFamily": "OS_FAMILY",
                "cpuArchitectureType": "x86_64",
```

```
            },
        },
    }
}
```

b. 변경 내용을 저장합니다.

6. 플릿을 생성하세요.

```
aws deadline create-fleet --cli-input-json file://create-fleet-request.json
```

작업자 호스트 설정 및 구성

작업자 호스트는 Deadline Cloud 작업자를 실행하는 호스트 컴퓨터를 말합니다. 이 섹션에서는 작업자 호스트를 설정하고 특정 요구 사항에 맞게 구성하는 방법을 설명합니다. 각 작업자 호스트는 작업자 에이전트라는 프로그램을 실행합니다. 작업자 에이전트는 다음을 담당합니다.

- 작업자 라이프 사이클 관리.
- 배정된 작업, 진행 상황 및 결과 동기화
- 실행 중인 작업을 모니터링합니다.
- 구성된 대상으로 로그 전달

제공된 Deadline Cloud 작업자 에이전트를 사용하는 것이 좋습니다. 작업자 에이전트는 오픈 소스이므로 기능 요청을 권장하지만 필요에 맞게 개발하고 사용자 지정할 수도 있습니다.

다음 섹션의 작업을 완료하려면 다음이 필요합니다.

Linux

- Linux기반 아마존 Elastic Compute Cloud (아마존 EC2) 인스턴스입니다. 아마존 리눅스 2023을 추천합니다.
- sudo특권.
- 파이썬 3.9 이상

Windows

- Windows기반 아마존 Elastic Compute Cloud (아마존 EC2) 인스턴스입니다. 추천합니다 Windows Server 2022.

- 작업자 호스트에 대한 관리자 액세스
- 모든 사용자를 위해 Python 3.9 이상이 설치되었습니다

Python 가상 환경 생성 및 구성

Python 3.9 이상을 설치하고 설치한 Linux 경우 Python 가상 환경을 만들 수 있습니다PATH.

Python 가상 환경을 만들고 활성화하려면

1. 를 엽니다 AWS CLI.
2. Python 가상 환경을 만들고 활성화합니다.

```
python3 -m venv /opt/deadline/worker
source /opt/deadline/worker/bin/activate
pip install --upgrade pip
```

데드라인 클라우드 워커 에이전트 설치

Python을 설정하고 가상 환경을 만든 후 Deadline Cloud 작업자 에이전트 Python 패키지를 설치합니다. Linux

워커 에이전트 Python 패키지를 설치하려면

1. 터미널을 엽니다.
   a. 켜기Linux, 사용자로 터미널을 `root` 엽니다 (또는 `sudo` /를 사용su).
   b. Windows켜기에서 관리자 명령 프롬프트 또는 PowerShell 터미널을 엽니다.
2. PyPI에서 데드라인 클라우드 워커 에이전트 패키지를 다운로드하여 설치하세요.

> **Note**
>
> 에서 Windows 에이전트 파일은 Python의 글로벌 사이트 패키지 디렉터리에 설치해야 합니다. Python 가상 환경은 현재 지원되지 않습니다.

```
python -m pip install deadline-cloud-worker-agent
```

데드라인 클라우드 워커 에이전트 구성

데드라인 클라우드 워커 에이전트 설정은 세 가지 방법으로 구성할 수 있습니다. 를 통해 설정된 운영 체제를 사용하는 것이 좋습니다`install-deadline-worker`.

명령줄 인수 - 명령줄에서 Deadline Cloud 작업자 에이전트를 실행할 때 인수를 지정할 수 있습니다. 일부 구성 설정은 명령줄 인수를 통해 사용할 수 없습니다. 사용 가능한 모든 명령줄 인수를 `deadline-worker-agent --help` 보려면 를 입력하여 사용 가능한 모든 명령줄 인수를 확인하십시오.

환경 변수 - 로 시작하는 환경 변수를 설정하여 Deadline Cloud 작업자 에이전트를 구성할 수 `DEADLINE_WORKER_` 있습니다. 예를 들어, 작업자 에이전트의 출력을 verbose로 설정하는 `export DEADLINE_WORKER_VERBOSE=true` 데 사용할 수 있습니다. 더 많은 예제와 정보는 `/etc/amazon/deadline/worker.toml.example` on Linux 또는 `C:\ProgramData\Amazon\Deadline\Config\worker.toml.example` on을 참조하십시오. Windows

구성 파일 - 작업자 에이전트를 설치하면 `/etc/amazon/deadline/worker.toml` `C:\ProgramData\Amazon\Deadline\Config\worker.toml` on Linux 또는 on에 구성 파일이 생성됩니다Windows. 작업자 에이전트는 시작할 때 이 구성 파일을 로드합니다. 예제 구성 파일 (`/etc/amazon/deadline/worker.toml.example`on Linux 또는 `C:\ProgramData\Amazon\Deadline\Config\worker.toml.example` onWindows) 을 사용하여 기본 작업자 에이전트 구성 파일을 특정 요구 사항에 맞게 조정할 수 있습니다.

마지막으로 작업자 에이전트에 대해 자동 종료를 활성화하는 것이 좋습니다. 이렇게 하면 필요할 때 워커 플릿을 확장하고 렌더링 작업이 완료되면 워커 플릿을 종료할 수 있습니다. Auto Scaling을 사용하면 필요한 만큼만 리소스를 사용할 수 있습니다.

자동 종료를 활성화하려면

root사용자로서:

- 파라미터를 사용하여 작업자 에이전트를 설치합니다**--allow-shutdown**.

  Linux

  다음을 입력합니다.

```
/opt/deadline/worker/bin/install-deadline-worker \
  --farm-id FARM_ID \
  --fleet-id FLEET_ID \
  --region REGION \
```

```
  --allow-shutdown
```

Windows

입력:

```
install-deadline-worker ^
  --farm-id FARM_ID ^
  --fleet-id FLEET_ID ^
  --region REGION ^
  --allow-shutdown
```

작업 사용자 및 그룹 생성

이 섹션에서는 에이전트 사용자와 큐에 jobRunAsUser 정의된 사용자 간의 필수 사용자 및 그룹 관계에 대해 설명합니다.

Deadline Cloud 작업자 에이전트는 호스트에서 에이전트별 전담 사용자로 실행해야 합니다. 작업자가 특정 운영 체제 사용자 및 그룹으로 대기열 작업을 실행하도록 Deadline Cloud 대기열의 jobRunAsUser 속성을 구성해야 합니다. 즉, 작업에 있는 공유 파일 시스템 권한을 제어할 수 있습니다. 또한 작업과 작업자 에이전트 사용자 사이에 중요한 보안 경계를 제공합니다.

Linux작업 사용자 및 그룹

에이전트-사용자를 설정하려면 다음 요구 사항을 충족해야 합니다. jobRunAsUser

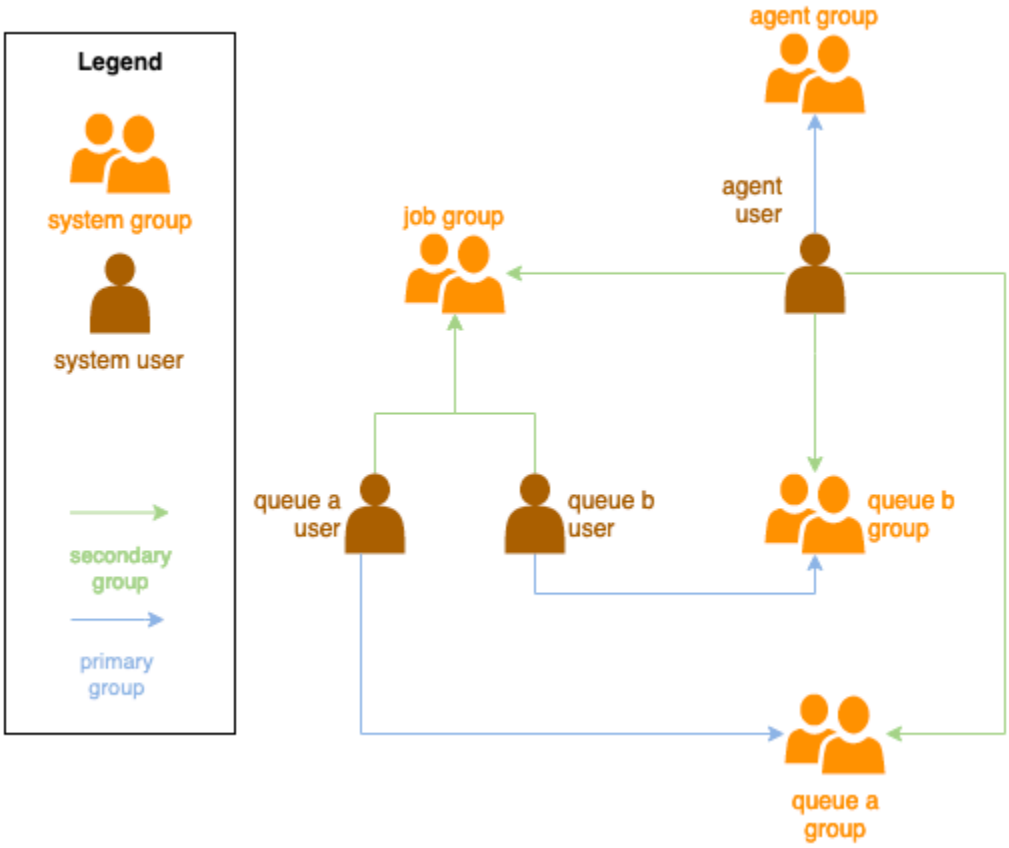
- 각 그룹에는 하나의 그룹이 있으며jobRunAsUser, 이 그룹은 해당 그룹의 기본 그룹입니다. jobRunAsUser
- 에이전트-사용자는 작업자가 작업을 받는 대기열의 기본 그룹에 속합니다. jobRunAsUser 보안 모범 사례를 위해 이 그룹을 에이전트-사용자의 보조 그룹으로 사용하는 것이 좋습니다. 이 공유 그룹을 사용하면 작업자 에이전트가 작업이 실행되는 동안 해당 작업에 파일을 사용할 수 있도록 할 수 있습니다.
- A는 에이전트-사용자의 기본 그룹에 속하지 *jobRunAsUser* 않습니다. 보안 모범 사례:
  - 작업자 에이전트가 작성한 민감한 파일은 에이전트의 기본 그룹이 소유합니다.
  - a가 이 그룹에 jobRunAsUser 속하는 경우 작업자에서 실행 중인 대기열에 제출된 작업은 작업자 에이전트가 쓰는 파일에 액세스할 수 있습니다.
- 기본 AWS 지역은 작업자가 속한 팜의 지역과 일치해야 합니다. 자세한 내용은 구성 및 자격 증명 파일 설정을 참조하십시오.

이는 다음에 적용되어야 합니다.

- 에이전트-사용자
- 워커의 모든 큐 `jobRunAsUser` 계정

- 에이전트-사용자는 로 `sudo` 명령을 실행할 수 있습니다. `jobRunAsUser`

다음 다이어그램은 플릿과 연결된 대기열에 대한 에이전트 `jobRunAsUser` 사용자와 사용자 및 그룹 간의 관계를 보여줍니다.



Windows 사용자

Windows사용자를 로 사용하려면 다음 `jobRunAsUser` 요구 사항을 충족해야 합니다.

- 모든 큐 `jobRunAsUser` 사용자가 존재해야 합니다.
- 암호는 대기열 `JobRunAsUser` 필드에 지정된 암호의 값과 일치해야 합니다. 지침은 의 7단계를 참조하십시오대기열 생성.
- 에이전트-사용자는 해당 사용자로 로그온할 수 있어야 합니다.

Windows 작업 사용자 비밀에 대한 액세스 관리

jobRunAsUserWindows에서 대기열을 구성할 때는 AWS Secrets Manager 암호를 지정해야 합니다. 이 시크릿의 값은 다음 형식의 JSON으로 인코딩된 객체여야 합니다.

```
{
    "password": "JOB_USER_PASSWORD"
}
```

워커가 대기열의 구성에 따라 작업을 실행하려면 플릿의 IAM 역할에 암호 값을 가져올 권한이 있어야 합니다. jobRunAsUser 고객 관리형 KMS 키를 사용하여 암호를 암호화하는 경우 플릿의 IAM 역할에도 KMS 키를 사용하여 암호를 해독할 수 있는 권한이 있어야 합니다.

이러한 비밀에 대한 최소 권한 원칙을 따르는 것이 좋습니다. 즉, 대기열의 jobRunAsUser → windows →의 비밀 값을 가져오기 위한 액세스 권한은 다음과 같아야 합니다. passwordArn

- 플릿과 큐 사이에 큐-플릿 연결이 생성되면 플릿 역할에 부여됩니다.
- 플릿과 큐 간에 큐-플릿 연결이 삭제되면 플릿 역할에서 취소됩니다.

또한 암호가 포함된 AWS Secrets Manager jobRunAsUser 암호는 더 이상 사용되지 않을 때 삭제해야 합니다.

비밀번호 비밀에 대한 액세스 권한을 부여하세요.

Deadline Cloud 플릿은 jobRunAsUser 큐와 플릿이 연결된 경우 큐의 비밀번호 비밀번호에 저장된 비밀번호에 액세스할 수 있어야 합니다. AWS Secrets Manager 리소스 정책을 사용하여 플릿 역할에 대한 액세스 권한을 부여하는 것이 좋습니다. 이 가이드라인을 엄격하게 준수하면 어떤 플릿 역할이 비밀에 액세스할 수 있는지 쉽게 확인할 수 있습니다.

시크릿에 대한 액세스 권한을 부여하려면

1. AWS 시크릿 매니저 콘솔을 열어 시크릿을 찾아보십시오.
2. “리소스 권한” 섹션에 다음 형식의 정책 설명을 추가합니다.

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

```
    {
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "FLEET_ROLE_ARN"
      },
      "Action" : "secretsmanager:GetSecretValue",
      "Resource" : "*"
    }
    // ...
  ]
}
```

비밀번호 비밀에 대한 액세스 권한 취소

플릿에서 더 이상 대기열에 액세스할 필요가 없는 경우 해당 대기열의 암호 비밀번호에 대한 액세스를 제거하세요. `jobRunAsUser` AWS Secrets Manager 리소스 정책을 사용하여 플릿 역할에 대한 액세스 권한을 부여하는 것이 좋습니다. 이 가이드라인을 엄격하게 준수하면 어떤 플릿 역할이 비밀에 액세스할 수 있는지 쉽게 확인할 수 있습니다.

시크릿에 대한 액세스 권한을 취소하려면

1. AWS 시크릿 매니저 콘솔을 열어 시크릿을 찾아보십시오.
2. 리소스 권한 섹션에서 다음 양식의 정책 설명을 제거합니다.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    // ...
    {
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "FLEET_ROLE_ARN"
      },
      "Action" : "secretsmanager:GetSecretValue",
      "Resource" : "*"
    }
    // ...
  ]
}
```

작업에 필요한 소프트웨어 설치 및 구성

Deadline Cloud 작업자 에이전트를 설정한 후에는 작업을 실행하는 데 필요한 모든 소프트웨어를 사용하여 작업자 호스트를 준비할 수 있습니다.

관련 jobRunAsUser 사용자와 함께 대기열에 작업을 제출하면 작업이 해당 사용자 권한으로 실행됩니다. 해당 사용자의 명령에서 모든 명령을 사용할 수 있어야 합니다. PATH

Linux에서는 다음 중 하나로 사용자를 PATH 위해 를 지정할 수 있습니다.

- 해당 ~/.bashrc 또는 ~/.bash_profile
- /etc/profile.d/*및 와 같은 시스템 구성 파일 /etc/profile
- 셸 시작 스크립트:/etc/bashrc.

Windows에서는 다음 중 하나에서 사용자를 PATH 위해 를 지정할 수 있습니다.

- 해당 사용자별 환경 변수
- 시스템 전반의 환경 변수

디지털 콘텐츠 제작 도구 어댑터 설치

Deadline Cloud는 퍼스트 파티 통합 지원과 함께 디지털 콘텐츠 제작 (DCC) 애플리케이션을 제공합니다. 고객 관리형 플릿에서 이러한 통합을 사용하려면 DCC 소프트웨어와 어댑터를 설치해야 합니다.

고객 관리형 플릿에 DCC 어댑터를 설치하려면

1. 터미널을 엽니다.
    a. Linux에서는 터미널을 사용자로 엽니다 (또는 sudo /를 사용su). root
    b. Windows에서는 관리자 명령 프롬프트 또는 PowerShell 터미널을 엽니다.
2. 데드라인 클라우드 어댑터 패키지를 설치합니다.

```
pip install deadline deadline-cloud-for-maya deadline-cloud-for-nuke deadline-
cloud-for-blender
```

AWS 자격 증명 구성

이 섹션에서는 AWS 자격 증명을 구성하는 방법을 설명합니다.

작업자 수명 주기의 초기 단계는 부트스트래핑입니다. 이 단계에서 작업자 에이전트 소프트웨어는 플릿에 작업자를 생성하고 추가 작업을 위해 플릿의 역할로부터 AWS 자격 증명을 얻습니다.

AWS credentials for Amazon EC2

Amazon EC2용 AWS 자격 증명을 구성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 다음 역할 생성을 선택합니다.
3. AWS 서비스를 선택합니다.
4. 서비스 또는 사용 사례로 EC2를 선택한 후 다음을 선택합니다.
5. `AWSDeadlineCloud-WorkerHost` AWS 관리형 정책을 연결합니다.

On-premise AWS credentials

AWS 온프레미스 자격 증명을 구성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 다음 역할 생성을 선택합니다.
3. 선택한 AWS 계정후 다음을 선택합니다.
4. `AWSDeadlineCloud-WorkerHost` AWS 관리형 정책을 연결합니다.
5. AWS IAM 사용자의 IAM 액세스 및 비밀 키 생성:
    a. 어디서나 사용할 수 있는 IAM 역할에 대해서는 어디서든 IAM 역할을 참조하십시오.
    b. 호스트에서 자격 증명을 설정하는 가장 안전한 방법은 AWS Identity 및 Access Management Roles Anywhere에서 임시 보안 자격 증명 획득을 참조하십시오.
    c. CLI를 대체 인증으로 사용할 수도 있습니다. 자세한 내용은 IAM 사용자 자격 증명으로 인증을 참조하십시오.
6. 이러한 키를 작업자 호스트 파일 시스템의 에이전트-사용자 AWS 자격 증명 파일에 저장합니다.
    a. Linux의 경우 이 위치는 다음과 같습니다. `~/.aws/credentials`

b. Windows의 경우 이 위치는 다음과 같습니다. `%USERPROFILE%\.aws\credentials`

> **Note**
>
> 자격 증명은 작업자 에이전트를 설치한 OS 사용자 이름 (`deadline-worker-agent`) 만 액세스할 수 있어야 합니다.

```
# Replace keys below
[default]
aws_access_key_id=ACCESS_KEY_ID
aws_secret_access_key=SECRET_ACCESSS_KEY
```

7. `deadline-worker-agent`소유자 및 권한을 변경하십시오.

> **Note**
>
> 작업자 에이전트를 설치할 때 OS 사용자 (`deadline-worker-agent`) 이름을 변경한 경우 해당 이름을 대신 사용하십시오.

Amazon Machine Image 생성

Amazon Elastic Compute Cloud Amazon Machine Image (Amazon EC2AMI) 의 고객 관리형 플릿 (CMF) 에서 사용할 () 를 생성하려면 이 섹션의 작업을 완료하십시오. 진행하기 전에 Amazon EC2 인스턴스를 생성해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스 시작을 참조하십시오.

> **Important**
>
> AMI생성하면 Amazon EC2 인스턴스에 연결된 볼륨의 스냅샷이 생성됩니다. 인스턴스에 설치된 모든 소프트웨어는 인스턴스가 유지되므로 에서 인스턴스를 시작할 때 해당 인스턴스가 재사용됩니다. AMI 플릿에 적용하기 전에 패치 전략을 채택하고 새 AMI 제품이 있으면 업데이트된 소프트웨어로 정기적으로 업데이트하는 것이 좋습니다.

Amazon EC2 인스턴스 준비

를 AMI 구축하기 전에 작업자 상태를 삭제해야 합니다. 작업자 상태는 작업자 에이전트가 실행될 때에도 지속됩니다. 이 상태가 에서 AMI 지속되면 해당 상태에서 시작된 모든 인스턴스가 동일한 상태를 공유하게 됩니다.

기존 로그 파일도 모두 삭제하는 것이 좋습니다. AMI를 준비할 때 로그 파일은 Amazon EC2 인스턴스에 남아 있을 수 있습니다. 이러한 파일을 삭제하면 AMI를 사용하는 워커 플릿에서 발생할 수 있는 문제를 진단할 때 혼동을 최소화할 수 있습니다.

또한 Amazon EC2가 시작될 때 Deadline Cloud 작업자 에이전트가 시작되도록 작업자 에이전트 시스템 서비스를 활성화해야 합니다.

마지막으로 작업자 에이전트 자동 종료를 활성화하는 것이 좋습니다. 이렇게 하면 필요할 때 워커 플릿을 확장하고 렌더링 작업이 완료되면 워커 플릿을 종료할 수 있습니다. 이 Auto Scaling을 사용하면 필요한 만큼만 리소스를 사용할 수 있습니다.

Amazon EC2 인스턴스를 준비하려면

1. Amazon EC2 콘솔을 엽니다.
2. Amazon EC2 인스턴스 시작 자세한 내용은 인스턴스 시작을 참조하십시오.
3. 호스트를 설정하여 ID 공급자 (IdP) 에 연결한 다음 필요한 공유 파일 시스템을 마운트합니다.
4. 튜토리얼을 따라,, 를 실행해 보세요. 데드라인 클라우드 워커 에이전트 설치 작업자 에이전트 구성 작업 사용자 및 그룹 생성
5. VFX 참조 플랫폼과 호환되는 소프트웨어를 실행하기 위해 Amazon Linux 2023을 AMI 기반으로 준비하려면 몇 가지 요구 사항을 업데이트해야 합니다. 자세한 내용은 VFX Reference Platform 호환성을 참조하세요.
6. 터미널을 엽니다.
   a. Linux에서는 `root` 사용자 권한으로 터미널을 엽니다 (또는 /를 사용sudo). `su`
   b. Windows에서는 관리자 명령 프롬프트 또는 PowerShell 터미널을 엽니다.
7. 작업자 서비스가 실행되고 있지 않고 부팅 시 시작되도록 구성되어 있는지 확인하십시오.
   a. Linux에서는 다음을 실행합니다.

```
systemctl stop deadline-worker
systemctl enable deadline-worker
```

   b. 윈도우에서는 다음을 실행합니다.

```
sc.exe stop DeadlineWorker
sc.exe config DeadlineWorker start= auto
```

8. 작업자 상태를 삭제합니다.

   a. Linux에서는 다음을 실행합니다.

   ```
   rm -rf /var/lib/deadline/*
   ```

   b. 윈도우에서는 다음을 실행합니다.

   ```
   del /Q /S %PROGRAMDATA%\Amazon\Deadline\Cache\*
   ```

9. 로그 파일을 삭제합니다.

   a. Linux에서는 다음을 실행합니다.

   ```
   rm -rf /var/log/amazon/deadline/*
   ```

   b. 윈도우에서는 다음을 실행합니다.

   ```
   del /Q /S %PROGRAMDATA%\Amazon\Deadline\Logs\*
   ```

10. Windows에서는 시작 메뉴에 있는 Amazon EC2Launch 설정 애플리케이션을 실행하여 최종 호스트 준비 및 인스턴스 종료를 완료하는 것이 좋습니다.

> **Note**
>
> Sysprep 없이 종료를 선택해야 하며 Sysprep을 사용한 종료를 선택하지 않아야 합니다. Sysprep으로 시스템을 종료하면 모든 로컬 사용자를 사용할 수 없게 됩니다. 자세한 내용은 Windows 인스턴스용 사용 설명서의 사용자 지정 AMI 생성 항목의 시작하기 전 섹션을 참조하십시오.

다음을 빌드하십시오. AMI

빌드하려면 AMI

1. Amazon EC2 콘솔을 엽니다.

2. 탐색 창에서 인스턴스를 선택한 다음 인스턴스를 선택합니다.
3. 인스턴스 상태를 선택한 다음 인스턴스 중지를 선택합니다.
4. 인스턴스가 중지된 후 Actions를 선택합니다.
5. 이미지 및 템플릿을 선택한 다음 이미지 생성을 선택합니다.
6. 이미지 이름을 입력합니다.
7. (선택 사항) 이미지에 대한 설명을 입력합니다.
8. 이미지 생성을 선택합니다.

Amazon EC2 Auto Scaling 그룹으로 플릿 인프라 생성

이 섹션에서는 Amazon EC2 Auto Scaling 플릿을 생성하는 방법을 설명합니다.

아래 AWS CloudFormation YAML 템플릿을 사용하여 Amazon EC2 Auto Scaling (Auto Scaling) 그룹, 두 개의 서브넷, 인스턴스 프로필, 인스턴스 액세스 역할이 있는 Amazon VPC (가상 사설 클라우드) 를 생성합니다. 이는 서브넷에서 Auto Scaling을 사용하여 인스턴스를 시작하는 데 필요합니다.

렌더링 요구 사항에 맞게 인스턴스 유형 목록을 검토하고 업데이트해야 합니다.

Amazon EC2 Auto Scaling 플릿을 만들려면

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 매개 변수 `Farm ID`Fleet ID, 및 를 사용하여 CloudFormation 템플릿을 생성합니다`AMI ID`.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Amazon Deadline Cloud customer-managed fleet
Parameters:
  FarmId:
    Type: String
    Description: Farm ID
  FleetId:
    Type: String
    Description: Fleet ID
  AMIId:
    Type: String
    Description: AMI ID for launcing Workers
Resources:
  deadlineVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
```

```
      CidrBlock: 100.100.0.0/16
  deadlineWorkerSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: !Join
        - ' '
        - - Security Group created for deadline workers in fleet
          - !Ref FleetId
      GroupName: !Join
        - ''
        - - deadlineWorkerSecurityGroup-
          - !Ref FleetId
      SecurityGroupEgress:
        - CidrIp: 0.0.0.0/0
          IpProtocol: '-1'
      SecurityGroupIngress: []
      VpcId: !Ref deadlineVPC
  deadlineIGW:
    Type: 'AWS::EC2::InternetGateway'
    Properties: {}
  deadlineVPCGatewayAttachment:
    Type: 'AWS::EC2::VPCGatewayAttachment'
    Properties:
      VpcId: !Ref deadlineVPC
      InternetGatewayId: !Ref deadlineIGW
  deadlinePublicRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref deadlineVPC
  deadlinePublicRoute:
    Type: 'AWS::EC2::Route'
    Properties:
      RouteTableId: !Ref deadlinePublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref deadlineIGW
    DependsOn:
      - deadlineIGW
      - deadlineVPCGatewayAttachment
  deadlinePublicSubnet0:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref deadlineVPC
      CidrBlock: 100.100.16.0/22
      AvailabilityZone: !Join
```

```
      - ''
      - - !Ref 'AWS::Region'
        - a
deadlineSubnetRouteTableAssociation0:
  Type: 'AWS::EC2::SubnetRouteTableAssociation'
  Properties:
    RouteTableId: !Ref deadlinePublicRouteTable
    SubnetId: !Ref deadlinePublicSubnet0
deadlinePublicSubnet1:
  Type: 'AWS::EC2::Subnet'
  Properties:
    VpcId: !Ref deadlineVPC
    CidrBlock: 100.100.20.0/22
    AvailabilityZone: !Join
      - ''
      - - !Ref 'AWS::Region'
        - c
deadlineSubnetRouteTableAssociation1:
  Type: 'AWS::EC2::SubnetRouteTableAssociation'
  Properties:
    RouteTableId: !Ref deadlinePublicRouteTable
    SubnetId: !Ref deadlinePublicSubnet1
deadlineInstanceAccessAccessRole:
  Type: 'AWS::IAM::Role'
  Properties:
    RoleName: !Join
      - '-'
      - - deadline
        - InstanceAccess
        - !Ref FleetId
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service: ec2.amazonaws.com
          Action:
            - 'sts:AssumeRole'
    Path: /
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy'
      - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
      - 'arn:aws:iam::aws:policy/AWSDeadlineCloud-WorkerHost'
deadlineInstanceProfile:
  Type: 'AWS::IAM::InstanceProfile'
```

```
    Properties:
      Path: /
      Roles:
        - !Ref deadlineInstanceAccessAccessRole
  deadlineLaunchTemplate:
    Type: 'AWS::EC2::LaunchTemplate'
    Properties:
      LaunchTemplateName: !Join
        - ''
        - - deadline-LT-
          - !Ref FleetId
      LaunchTemplateData:
        NetworkInterfaces:
          - DeviceIndex: 0
            AssociatePublicIpAddress: true
            Groups:
              - !Ref deadlineWorkerSecurityGroup
            DeleteOnTermination: true
        ImageId: !Ref AMIId
        InstanceInitiatedShutdownBehavior: terminate
        IamInstanceProfile:
          Arn: !GetAtt
            - deadlineInstanceProfile
            - Arn
        MetadataOptions:
          HttpTokens: required
          HttpEndpoint: enabled

  deadlineAutoScalingGroup:
    Type: 'AWS::AutoScaling::AutoScalingGroup'
    Properties:
      AutoScalingGroupName: !Join
        - ''
        - - deadline-ASG-autoscalable-
          - !Ref FleetId
      MinSize: 0
      MaxSize: 10
      VPCZoneIdentifier:
        - !Ref deadlinePublicSubnet0
        - !Ref deadlinePublicSubnet1
      NewInstancesProtectedFromScaleIn: true
      MixedInstancesPolicy:
        InstancesDistribution:
          OnDemandBaseCapacity: 0
```

```
          OnDemandPercentageAboveBaseCapacity: 0
          SpotAllocationStrategy: capacity-optimized
          OnDemandAllocationStrategy: lowest-price
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref deadlineLaunchTemplate
            Version: !GetAtt
              - deadlineLaunchTemplate
              - LatestVersionNumber
          Overrides:
            - InstanceType: m5.large
            - InstanceType: m5d.large
            - InstanceType: m5a.large
            - InstanceType: m5ad.large
            - InstanceType: m5n.large
            - InstanceType: m5dn.large
            - InstanceType: m4.large
            - InstanceType: m3.large
            - InstanceType: r5.large
            - InstanceType: r5d.large
            - InstanceType: r5a.large
            - InstanceType: r5ad.large
            - InstanceType: r5n.large
            - InstanceType: r5dn.large
            - InstanceType: r4.large
      MetricsCollection:
        - Granularity: 1Minute
          Metrics:
            - GroupMinSize
            - GroupMaxSize
            - GroupDesiredCapacity
            - GroupInServiceInstances
            - GroupTotalInstances
            - GroupInServiceCapacity
            - GroupTotalCapacity
```

3. IAM 역할을 생성한 후에는 다음 사항을 확인해야 합니다.

   - 작업자의 Amazon EC2 인스턴스에 연결된 IAM 역할의 자격 증명은 작업을 포함하여 해당 작업자에서 실행 중인 모든 프로세스에서 사용할 수 있습니다. 작업자는 다음과 같은 작업을 수행할 수 있는 최소한의 권한을 가져야 합니다. `deadline:CreateWorker` `deadline:AssumeFleetRoleForWorker`.

- 작업자 에이전트는 큐 역할에 대한 자격 증명을 획득하여 작업을 실행하는 데 사용할 수 있도록 구성합니다. Amazon EC2 인스턴스 프로필 역할에는 작업에 필요한 권한이 포함되어서는 안 됩니다.

데드라인 클라우드 스케일 권장 기능을 사용하여 Amazon EC2 플릿을 자동 확장하십시오.

데드라인 클라우드는 Amazon EC2 Auto Scaling (Auto Scaling) 그룹을 활용하여 Amazon EC2 고객 관리형 플릿 (CMF) 을 자동으로 확장합니다. 플릿을 자동 확장하려면 플릿 모드를 구성하고 계정에 필요한 인프라를 배포해야 합니다. 배포한 인프라는 모든 플릿에서 작동하므로 한 번만 설정하면 됩니다.

기본 워크플로는 플릿 모드를 자동 확장하도록 구성하면 Deadline Cloud에서 권장 플릿 크기가 변경될 때마다 해당 플릿에 대한 EventBridge 이벤트를 전송하는 것입니다 (한 이벤트에는 플릿 ID, 권장 플릿 크기 및 기타 메타데이터가 포함됨). 관련 이벤트를 필터링하고 Lambda가 해당 이벤트를 사용하도록 하는 EventBridge 규칙이 있습니다. Lambda는 Amazon EC2 Auto Scaling과 통합되어 Amazon EC2 플릿을 자동으로 `AutoScalingGroup` 확장할 것입니다.

플릿 모드를 다음으로 설정합니다. **`EVENT_BASED_AUTO_SCALING`**

플릿 모드를 로 구성하십시오`EVENT_BASED_AUTO_SCALING`. 콘솔을 사용하여 이 작업을 수행하거나를 사용하여 `CreateFleet` 또는 `UpdateFleet` API를 직접 AWS CLI 호출할 수 있습니다. 모드가 구성된 후에는 권장 플릿 크기가 변경될 때마다 Deadline Cloud에서 EventBridge 이벤트 전송을 시작합니다.

- `UpdateFleet`명령 예시:

```
aws deadline update-fleet \
  --farm-id FARM_ID \
  --fleet-id FLEET_ID \
  --configuration file://configuration.json
```

- `CreateFleet`명령 예시:

```
aws deadline create-fleet \
  --farm-id FARM_ID \
  --display-name "Fleet name" \
  --max-worker-count 10 \
  --configuration file://configuration.json
```

다음은 위 () `--configuration file://configuration.json` CLI 명령에서 `configuration.json` 사용된 예제입니다.

- 플릿에서 Auto Scaling을 활성화하려면 모드를 로 설정해야 `EVENT_BASED_AUTO_SCALING` 합니다.
- CMF를 생성할 때 CMF에 할당된 기본값입니다. `workerCapabilities` CMF에 사용할 수 있는 리소스를 늘려야 하는 경우 이 값을 변경할 수 있습니다.

플릿 모드를 구성한 후 Deadline Cloud는 해당 플릿에 대한 플릿 크기 권장 이벤트를 생성하기 시작합니다.

```
{
    "customerManaged": {
        "mode": "EVENT_BASED_AUTO_SCALING",
        "workerCapabilities": {
            "vCpuCount": {
                "min": 1,
                "max": 4
            },
            "memoryMiB": {
                "min": 1024,
                "max": 4096
            },
            "osFamily": "linux",
            "cpuArchitectureType": "x86_64",
        }
    }
}
```

AWS CloudFormation 템플릿을 사용하여 Auto Scaling 스택을 배포합니다.

이벤트를 필터링하는 EventBridge 규칙, 이벤트를 사용하고 Auto Scaling을 제어하는 Lambda, 처리되지 않은 이벤트를 저장하는 SQS 대기열을 설정할 수 있습니다. 다음 AWS CloudFormation 템플릿을 사용하여 스택에 모든 것을 배포하십시오. 리소스를 성공적으로 배포한 후 작업을 제출하면 플릿이 자동으로 확장됩니다.

```
Resources:
  AutoScalingLambda:
    Type: 'AWS::Lambda::Function'
    Properties:
      Code:
```

```
        ZipFile: |-
          """
          This lambda is configured to handle "Fleet Size Recommendation Change"
          messages. It will handle all such events, and requires
          that the ASG is named based on the fleet id. It will scale up/down the fleet
          based on the recommended fleet size in the message.

          Example EventBridge message:
          {
              "version": "0",
              "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
              "detail-type": "Fleet Size Recommendation Change",
              "source": "aws.deadline",
              "account": "111122223333",
              "time": "2017-12-22T18:43:48Z",
              "region": "us-west-1",
              "resources": [],
              "detail": {
                  "farmId": "farm-12345678900000000000000000000000",
                  "fleetId": "fleet-12345678900000000000000000000000",
                  "oldFleetSize": 1,
                  "newFleetSize": 5,
              }
          }
          """

          import json
          import boto3
          import logging

          logger = logging.getLogger()
          logger.setLevel(logging.INFO)

          auto_scaling_client = boto3.client("autoscaling")

          def lambda_handler(event, context):
              logger.info(event)
              event_detail = event["detail"]
              fleet_id = event_detail["fleetId"]
              desired_capacity = event_detail["newFleetSize"]

              asg_name = f"deadline-ASG-autoscalable-{fleet_id}"
              auto_scaling_client.set_desired_capacity(
                  AutoScalingGroupName=asg_name,
```

```
                DesiredCapacity=desired_capacity,
                HonorCooldown=False,
            )

            return {
                'statusCode': 200,
                'body': json.dumps(f'Successfully set desired_capacity for {asg_name}
to {desired_capacity}')
            }
    Handler: index.lambda_handler
    Role: !GetAtt
      - AutoScalingLambdaServiceRole
      - Arn
    Runtime: python3.11
  DependsOn:
    - AutoScalingLambdaServiceRoleDefaultPolicy
    - AutoScalingLambdaServiceRole
 AutoScalingEventRule:
  Type: 'AWS::Events::Rule'
  Properties:
    EventPattern:
      source:
        - aws.deadline
      detail-type:
        - Fleet Size Recommendation Change
    State: ENABLED
    Targets:
      - Arn: !GetAtt
          - AutoScalingLambda
          - Arn
        DeadLetterConfig:
          Arn: !GetAtt
            - UnprocessedAutoScalingEventQueue
            - Arn
        Id: Target0
        RetryPolicy:
          MaximumRetryAttempts: 15
 AutoScalingEventRuleTargetPermission:
  Type: 'AWS::Lambda::Permission'
  Properties:
    Action: 'lambda:InvokeFunction'
    FunctionName: !GetAtt
      - AutoScalingLambda
      - Arn
```

```
      Principal: events.amazonaws.com
      SourceArn: !GetAtt
        - AutoScalingEventRule
        - Arn
  AutoScalingLambdaServiceRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Action: 'sts:AssumeRole'
            Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
        Version: 2012-10-17
      ManagedPolicyArns:
        - !Join
          - ''
          - - 'arn:'
            - !Ref 'AWS::Partition'
            - ':iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
  AutoScalingLambdaServiceRoleDefaultPolicy:
    Type: 'AWS::IAM::Policy'
    Properties:
      PolicyDocument:
        Statement:
          - Action: 'autoscaling:SetDesiredCapacity'
            Effect: Allow
            Resource: '*'
        Version: 2012-10-17
      PolicyName: AutoScalingLambdaServiceRoleDefaultPolicy
      Roles:
        - !Ref AutoScalingLambdaServiceRole
  UnprocessedAutoScalingEventQueue:
    Type: 'AWS::SQS::Queue'
    Properties:
      QueueName: deadline-unprocessed-autoscaling-events
    UpdateReplacePolicy: Delete
    DeletionPolicy: Delete
  UnprocessedAutoScalingEventQueuePolicy:
    Type: 'AWS::SQS::QueuePolicy'
    Properties:
      PolicyDocument:
        Statement:
          - Action: 'sqs:SendMessage'
```

```
            Condition:
              ArnEquals:
                'aws:SourceArn': !GetAtt
                  - AutoScalingEventRule
                  - Arn
            Effect: Allow
            Principal:
              Service: events.amazonaws.com
            Resource: !GetAtt
              - UnprocessedAutoScalingEventQueue
              - Arn
        Version: 2012-10-17
      Queues:
        - !Ref UnprocessedAutoScalingEventQueue
```

고객 관리 플릿을 라이선스 엔드포인트에 연결

AWS 데드라인 클라우드 (Deadline Cloud) 사용량 기반 라이선스 서버는 일부 타사 제품에 대한 온디맨드 라이선스를 제공합니다. 이를 통해 사용한 만큼 비용을 지불할 수 있습니다. 사용한 시간 동안만 변경 가능합니다.

Deadline Cloud 사용량 기반 라이선스 서버는 Deadline Cloud 작업자가 라이선스 서버와 통신할 수 있는 한 모든 플릿 유형에 사용할 수 있습니다. 이는 서비스 관리 플릿에 자동으로 설정됩니다. 이 설정은 고객 관리 플릿에만 필요합니다.

라이선스 서버를 만들려면 다음이 필요합니다.

- 타사 라이선스에 대한 트래픽을 허용하는 팜 VPC의 보안 그룹입니다.
- Deadline Cloud 라이선스 엔드포인트 작업에 대한 액세스를 허용하는 정책이 첨부된 AWS Identity and Access Management (IAM) 역할입니다.

주제

- 1단계: 보안 그룹 생성
- 2단계: 라이선스 엔드포인트 설정
- 3단계: 렌더링 애플리케이션을 엔드포인트에 연결

1단계: 보안 그룹 생성

Amazon VPC 콘솔 (https://console.aws.amazon.com/vpc/) 을 사용하여 팜의 VPC를 위한 보안 그룹을 생성할 수 있습니다. 다음 인바운드 규칙을 허용하도록 보안 그룹을 구성하십시오.

- 오토데스크 마야 및 아놀드 — 2701 - 2702, TCP, IPv4
- 오토데스크 3ds 맥스 — 2704, TCP, IPv4
- 파운드리 나이크 — 6101, TCP, IPv4
- 사이드폭스 후디니, 만트라, 카르마 — 1715 - 1717, TCP, IPv4

각 인바운드 규칙의 출처는 플릿의 작업자 보안 그룹입니다.

보안 그룹을 생성하는 방법에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 보안 그룹 생성을 참조하십시오.

2단계: 라이선스 엔드포인트 설정

라이선스 엔드포인트는 타사 제품의 라이선스 서버에 대한 액세스를 제공합니다. 라이선스 요청은 라이선스 엔드포인트로 전송됩니다. 엔드포인트는 해당 라이선스 서버로 요청을 라우팅합니다. 라이선스 서버는 사용 제한 및 권한을 추적합니다. 생성한 각 라이선스 엔드포인트에 대해 요금이 부과됩니다. 자세한 내용은 Amazon VPC 요금을 참조하세요.

적절한 권한을 사용하여 에서 라이선스 엔드포인트를 생성할 수 있습니다. AWS Command Line Interface 라이선스 엔드포인트를 생성하는 데 필요한 정책은 라이선스 엔드포인트 생성을 허용하는 정책을 참조하십시오.

AWS CloudShell (https://console.aws.amazon.com/cloudshell/) 또는 기타 AWS CLI 환경을 사용하여 다음 AWS Command Line Interface 명령을 사용하여 라이선스 엔드포인트를 구성할 수 있습니다.

1. 라이선스 엔드포인트를 생성합니다. 보안 그룹 ID, 서브넷 ID, VPC ID를 이전에 생성한 값으로 교체합니다. 서브넷을 여러 개 사용하는 경우 공백으로 구분하십시오.

   ```
   aws deadline create-license-endpoint \
     --security-group-id SECURITY_GROUP_ID \
     --subnet-ids SUBNET_ID1 SUBNET_ID2 \
     --vpc-id VPC_ID
   ```

2. 다음 명령을 사용하여 엔드포인트가 성공적으로 생성되었는지 확인합니다. VPC 엔드포인트의 DNS 이름을 기억하십시오.

```
aws deadline get-license-endpoint \
  --license-endpoint-id LICENSE_ENDPOINT_ID
```

3. 사용 가능한 미터링 제품 목록 보기:

```
aws deadline list-available-metered-products
```

4. 다음 명령을 사용하여 사용량 제한 제품을 라이선스 엔드포인트에 추가합니다.

```
aws deadline put-metered-product \
--license-endpoint-id LICENSE_ENDPOINT_ID \
--product-id PRODUCT_ID
```

다음 `remove-metered-product` 명령을 사용하여 라이선스 엔드포인트에서 제품을 제거할 수 있습니다.

```
aws deadline remove-metered-product \
  --license-endpoint-id LICENSE_ENDPOINT_ID \
  --productId PRODUCT_ID
```

다음 `delete-license-endpoint` 명령을 사용하여 라이선스 엔드포인트를 삭제할 수 있습니다.

```
aws deadline delete-license-endpoint \
 --license-endpoint-id LICENSE_ENDPOINT_ID
```

3단계: 렌더링 애플리케이션을 엔드포인트에 연결

라이선스 엔드포인트가 설정되면 애플리케이션은 타사 라이선스 서버를 사용하는 것과 동일하게 이를 사용합니다. 일반적으로 Microsoft Windows 레지스트리 키와 같은 환경 변수나 기타 시스템 설정을 라이선스 서버 포트 및 주소로 설정하여 응용 프로그램의 라이선스 서버를 구성합니다.

라이선스 엔드포인트 DNS 이름을 가져오려면 다음 AWS CLI 명령을 사용합니다.

```
aws deadline get-license-endpoint
```

또는 Amazon VPC 콘솔 (https://console.aws.amazon.com/vpc/) 을 사용하여 이전 단계에서 데드라인 클라우드 API로 생성한 VPC 엔드포인트를 식별할 수 있습니다.

구성 예제

Example — 오토데스크 마야와 아놀드

환경 변수를 다음과 같이 설정합니다. ADSKFLEX_LICENSE_FILE

```
2702@VPC_Endpoint_DNS_Name:2701@VPC_Endpoint_DNS_Name
```

> **Note**
> Windows워커의 경우 콜론 (:) 대신 세미콜론 (;) 을 사용하여 엔드포인트를 구분합니다.

Example — 오토데스크 3ds Max

환경 변수를 ADSKFLEX_LICENSE_FILE 다음과 같이 설정합니다.

```
2704@VPC_Endpoint_DNS_Name
```

Example — 파운드리 뉴크

환경 변수를 다음과 같이 설정합니다. 라이센스가 제대로 작동하는지 foundry_LICENSE 6101@*VPC_Endpoint_DNS_Name* 테스트하려면 터미널에서 Nuke를 실행할 수 있습니다.

```
~/nuke/Nuke14.0v5/Nuke14.0 -x
```

Example — SideFX, 후디니, 만트라, 카르마

다음 명령을 실행합니다:

```
/opt/hfs19.5.640/bin/hserver -S
 "http://VPC_Endpoint_DNS_Name:1715;http://VPC_Endpoint_DNS_Name:1716;http://
VPC_Endpoint_DNS_Name:1717;"
```

라이선스가 제대로 작동하는지 테스트하려면 다음 명령을 사용하여 Houdini 씬을 렌더링할 수 있습니다.

```
/opt/hfs19.5.640/bin/hython ~/forpentest.hip -c "hou.node('/out/mantra1').render()"
```

데드라인 클라우드에서의 사용자 관리

AWS 데드라인 클라우드는 사용자와 AWS IAM Identity Center 그룹을 관리하는 데 사용합니다. IAM Identity Center는 엔터프라이즈 싱글 사인온 (SSO) 공급자와 통합할 수 있는 클라우드 기반 싱글 사인온 서비스입니다. 통합을 통해 사용자는 회사 계정으로 로그인할 수 있습니다.

데드라인 클라우드는 기본적으로 IAM Identity Center를 활성화하며, 데드라인 클라우드를 설정하고 사용하려면 데드라인 클라우드가 필요합니다. 자세한 내용은 ID 소스 관리를 참조하십시오.

Deadline Cloud 모니터에 액세스할 수 있는 사용자와 그룹을 관리할 AWS Organizations 책임은 조직 소유자에게 있습니다. IAM ID 센터 또는 Deadline Cloud 콘솔을 사용하여 이러한 사용자 및 그룹을 생성하고 관리할 수 있습니다. 자세한 내용은 AWS Organizations란 무엇인가요?를 참조하세요.

Deadline Cloud 콘솔을 사용하여 모니터를 사용하여 팜, 큐, 플릿을 관리할 수 있는 사용자 및 그룹을 생성하고 제거할 수 있습니다. Deadline Cloud에 사용자를 추가할 때는 IAM Identity Center를 사용하여 비밀번호를 재설정해야 액세스할 수 있습니다.

주제

- 모니터의 사용자 및 그룹을 관리합니다.
- 팜, 큐, 플릿의 사용자 및 그룹을 관리합니다.

모니터의 사용자 및 그룹을 관리합니다.

Organizations 소유자는 Deadline Cloud 콘솔을 사용하여 Deadline Cloud 모니터에 액세스할 수 있는 사용자 및 그룹을 관리할 수 있습니다. 기존 IAM Identity Center 사용자 및 그룹 중에서 선택하거나 콘솔에서 새 사용자 및 그룹을 추가할 수 있습니다.

1. 에 AWS Management Console 로그인하고 Deadline Cloud 콘솔을 엽니다. 메인 페이지의 시작하기 섹션에서 데드라인 클라우드 설정 또는 대시보드로 이동을 선택합니다.
2. 왼쪽 탐색 창에서 사용자 관리를 선택합니다. 기본적으로 그룹 탭이 선택됩니다.

수행할 작업에 따라 그룹 탭 또는 사용자 탭을 선택합니다.

Monitor groups

그룹을 생성하려면

1. 그룹 생성을 선택합니다.

2. 그룹 이름을 입력합니다. 이름은 IAM ID 센터 조직 내 그룹 간에 고유해야 합니다.

그룹을 제거하려면

1. 제거할 그룹을 선택합니다.
2. 제거를 선택합니다.
3. 확인 대화 상자에서 그룹 제거를 선택합니다.

> **Note**
>
> IAM ID 센터에서 그룹을 제거하고 있습니다. 그룹 구성원은 더 이상 Deadline Cloud에 로그인하거나 팜 리소스에 액세스할 수 없습니다.

Monitor users

사용자를 추가하려면

1. 사용자 탭을 선택합니다.
2. 사용자 추가를 선택합니다.
3. 새 사용자의 이름, 이메일 주소, 사용자 이름을 입력합니다.
4. 원하는 경우 새 사용자를 추가할 IAM Identity Center 그룹을 하나 이상 선택합니다.
5. 초대 보내기를 선택하여 새 사용자에게 IAM Identity Center 조직 가입 지침이 포함된 이메일을 보내십시오.

사용자를 제거하려면

1. 모니터에서 제거할 사용자를 선택합니다.
2. 제거를 선택합니다.
3. 확인 대화 상자에서 사용자 제거를 선택합니다.

> **Note**
>
> IAM ID 센터에서 사용자를 제거하고 있습니다. 사용자는 더 이상 Deadline Cloud 모니터에 로그인하거나 팜 리소스에 액세스할 수 없습니다.

팜, 큐, 플릿의 사용자 및 그룹을 관리합니다.

1. 아직 로그인하지 않았다면 Deadline Cloud 콘솔에 AWS Management Console 로그인하여 여십시오.
2. 왼쪽 탐색 창에서 팜 및 기타 리소스를 선택합니다.
3. 관리할 팜을 선택합니다. 팜 이름을 선택하여 세부 정보 페이지를 엽니다. 검색 창을 사용하여 팜을 검색할 수 있습니다.
4. 대기열 또는 플릿을 관리하려면 Queues 또는 Fleets 탭을 선택한 다음 관리할 대기열 또는 플릿을 선택합니다.
5. 액세스 관리 탭을 선택합니다. 기본적으로 그룹 탭이 선택됩니다. 사용자를 관리하려면 토글을 사용자로 이동하십시오.

수행할 작업에 따라 그룹 탭 또는 사용자 탭을 선택합니다.

액세스 수준 정의는 권한을 참조하십시오.

Groups

그룹을 추가하려면

1. 그룹 토글을 선택합니다.
2. 그룹 추가를 선택합니다.
3. 드롭다운에서 추가할 그룹을 선택합니다.
4. 그룹 액세스 수준의 경우 다음 옵션 중 하나를 선택합니다.
   - 최종 사용자
   - 기고자
   - 매니저
   - 소유자
5. 추가를 선택합니다.

그룹을 제거하려면

1. 제거할 그룹을 선택합니다.
2. 제거를 선택합니다.

3. 확인 대화 상자에서 제거를 선택합니다.

Users

사용자를 추가하려면

1. 사용자를 추가하려면 [사용자 추가] 를 선택합니다.
2. 드롭다운에서 팜에 추가할 사용자를 선택합니다.
3. 사용자 액세스 수준에 대해 다음 옵션 중 하나를 선택합니다.
   - 최종 사용자
   - 기고자
   - 매니저
   - 소유자
4. 추가를 선택합니다. 사용자가 팜에 추가됩니다.

사용자를 제거하려면

1. 제거할 사용자를 선택합니다.
2. 제거 확인 대화 상자에서 제거를 선택합니다. 그러면 선택한 팜에서 사용자가 제거됩니다.

또한 https://console.aws.amazon.com/singlesignon/ 에서 IAM Identity Center 콘솔을 사용하여 사용자 및 그룹에 대한 팜 권한을 추가하거나 제거할 수 있습니다.

데드라인 클라우드 작업

작업은 AWS Deadline Cloud에서 작업 일정을 잡고 가용 작업자에 대한 작업을 실행하는 데 사용하는 일련의 지침입니다. 작업을 생성할 때는 작업을 전송할 팜과 대기열을 선택합니다. 작업자가 처리할 지침을 제공하는 JSON 또는 YAML 파일도 제공합니다. 데드라인 클라우드는 작업 설명을 위한 공개 작업 설명 (OpenJD) 사양을 따르는 작업 템플릿을 허용합니다. 자세한 내용은 GitHub 웹 사이트의 Open Job Description 설명서를 참조하십시오.

직무는 다음과 같이 구성됩니다.

- 단계 — 작업자에서 실행할 스크립트를 정의합니다. 단계에는 최소 작업자 메모리 또는 먼저 완료해야 하는 기타 단계와 같은 요구 사항이 있을 수 있습니다. 각 단계에는 하나 이상의 작업이 있습니다.
- 태스크 — 작업을 수행하기 위해 작업자에게 보내는 작업 단위입니다. 작업은 단계 스크립트와 스크립트에서 사용되는 프레임 번호와 같은 매개 변수의 조합입니다. 모든 단계에 대한 모든 작업이 완료되면 작업이 완료됩니다.
- 환경 — 여러 단계 또는 작업에서 공유하는 지침을 설정하고 해제할 수 있습니다.

다음과 같은 방법으로 작업을 생성할 수 있습니다.

- 데드라인 클라우드 제출자를 사용하세요.
- 작업 번들을 생성하고 데드라인 클라우드 명령줄 인터페이스 (데드라인 클라우드 CLI) 를 사용합니다.
- AWS SDK를 사용하세요.
- AWS Command Line Interface (AWS CLI) 를 사용하세요.

제출자는 DCC 소프트웨어 인터페이스에서 작업 생성을 관리하는 디지털 콘텐츠 제작 (DCC) 소프트웨어용 플러그인입니다. 작업을 생성한 후에는 제출자를 사용하여 Deadline Cloud로 전송하여 처리합니다. 제출자는 백그라운드에서 작업을 설명하는 OpenJD 작업 템플릿을 만듭니다. 동시에 자산 파일을 Amazon Simple Storage Service (Amazon S3) 버킷에 업로드합니다. 파일을 전송하는 데 걸리는 시간을 줄이기 위해 파일을 마지막으로 업로드한 이후 변경된 파일만 Amazon S3로 전송됩니다.

Deadline Cloud에 작업을 제출하기 위한 스크립트와 파이프라인을 직접 만들려면 Deadline Cloud CLI, AWS SDK 또는 AWS CLI to call 작업을 사용하여 작업을 만들고, 가져오고, 보고, 나열할 수 있습니다. 다음 항목에서는 데드라인 클라우드 CLI를 사용하는 방법을 설명합니다.

데드라인 클라우드 CLI는 데드라인 클라우드 제출자와 함께 설치됩니다. 자세한 정보는 데드라인 클라우드 제출자를 설정하세요.을 참조하세요.

주제

- 데드라인 클라우드 CLI를 사용하여 작업 제출
- 데드라인 클라우드에서 작업 스케줄링
- 데드라인 클라우드 CLI의 작업 상태
- 데드라인 클라우드에서 작업 수정
- 데드라인 클라우드가 작업을 처리하는 방법
- 데드라인 클라우드 작업 문제 해결

데드라인 클라우드 CLI를 사용하여 작업 제출

데드라인 클라우드 명령줄 인터페이스 (데드라인 클라우드 CLI) 를 사용하여 작업을 제출하려면 명령을 사용합니다. `deadline bundle submit`

작업이 대기열에 제출됩니다. 팜과 대기열을 아직 설정하지 않은 경우 Deadline Cloud 콘솔 (https://console.aws.amazon.com/https://console.aws.amazon.com/deadlinecloud/home) 을 사용하여 팜과 대기열을 설정하고 팜 및 대기열 ID를 확인하십시오. 자세한 내용은 팜 세부 정보 정의 및 큐 세부 정보 정의를 참조하십시오.

Deadline Cloud CLI의 기본 팜 및 대기열을 설정하려면 다음 명령을 사용합니다. 기본값을 설정하면 팜이나 대기열을 지정하지 않고도 Deadline Cloud CLI 명령을 사용할 수 있습니다. 다음 예제에서는 *farmId* 및 를 사용자 고유의 *queueId* 정보로 바꾸십시오.

```
deadline config set defaults.farm_id farmId
deadline config set defaults.queue_id queueId
```

작업의 단계와 작업을 지정하려면 OpenJD 작업 템플릿을 만드십시오. 자세한 내용은 Open Job Description 사양 리포지토리의 템플릿 스키마 [버전: 2023-09] 를 참조하십시오. GitHub

다음 예는 YAML 작업 템플릿입니다. 두 단계와 단계당 다섯 개의 작업으로 구성된 작업을 정의합니다.

```
name: Sample Job
specificationVersion: jobtemplate-2023-09
steps:
```

```
- name: Sample Step 1
  parameterSpace:
    taskParameterDefinitions:
    - name: var
      range: 1-5
      type: INT
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
- name: Sample Step 2
  parameterSpace:
    taskParameterDefinitions:
    - name: var
      range: 1-5
      type: INT
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
```

작업을 생성하려면 이라는 sample_job 새 폴더를 만든 다음 새 폴더에 템플릿 파일을 로 저장합니다template.yaml. 다음 데드라인 클라우드 CLI 명령을 사용하여 작업을 제출합니다.

```
deadline bundle submit path/to/sample_job
```

명령의 응답에는 작업의 식별자가 포함됩니다. 나중에 작업 상태를 확인할 수 있도록 ID를 기억해 두십시오.

```
Submitting to Queue: test-queue
Waiting for Job to be created...
Submitted job bundle:
   sample_job
Job creation completed successfully
jobId
```

작업을 제출할 때 사용할 수 있는 추가 옵션이 있습니다. 자세한 정보는 데드라인 클라우드 CLI를 사용하여 작업을 제출하기 위한 추가 옵션을 참조하세요.

데드라인 클라우드 CLI를 사용하여 작업을 제출하기 위한 추가 옵션

`deadline bundle submit`Deadline Cloud CLI 명령은 작업에 대한 추가 정보를 지정하는 데 사용할 수 있는 옵션을 제공합니다. 다음 예제에서는 다음과 같은 작업을 하는 방법을 보여줍니다.

- 작업 템플릿을 처리할 때 사용되는 매개변수를 지정합니다.
- 공유 환경의 파일 및 폴더를 작업에 첨부합니다.
- 작업이 취소되기 전까지 최대 작업 실패 횟수를 설정합니다.
- 작업의 최대 재시도 횟수를 설정합니다.

작업 파라미터

이 `parameters` 옵션은 작업을 생성할 때 작업 매개 변수 값을 설정합니다. 작업 템플릿은 필드를 정의하고 `parameters` 옵션은 값을 설정합니다. 매개변수는 기본값을 가질 수 있습니다. 매개 변수 값이 지정된 경우 지정된 값이 기본값보다 우선 적용됩니다.

다음 작업 템플릿은 필드를 정의합니다. `TestParameter`

```
name: Sample Job With Job Parameter
parameterDefinitions:
- default: test
  name: TestParameter
  type: STRING
specificationVersion: jobtemplate-2023-09
steps:
- description: step description
  name: MyStep
  parameterSpace:
    taskParameterDefinitions:
    - name: var
      range: 1-5
      type: INT
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
```

다음 명령은 의 `TestParameter` 값을 “Hello AWS”로 설정합니다.

```
deadline bundle submit sample_job --parameter "TestParameter=Hello AWS"
```

스토리지 프로필

저장소 프로필은 운영 체제가 다른 작업자 간에 파일을 공유하는 데 도움이 됩니다. Deadline Cloud 콘솔을 사용하여 스토리지 프로필을 생성합니다. 그런 다음 `storage-profile-id` 파라미터를 사용하여 스토리지 프로필을 사용합니다. 자세한 정보는 데드라인 클라우드의 공유 스토리지을 참조하세요.

Deadline Cloud CLI를 사용하여 작업 제출을 위한 스토리지 프로필을 설정하려면 다음 명령을 사용하여 구성 매개변수를 설정합니다`storage-profile-id`.

```
deadline config set settings.storage_profile_id storageProfileId
```

실패한 최대 작업 수

이 `max-failed-tasks-count` 옵션은 전체 작업이 실패하고 나머지 모든 작업이 표시되기 전에 실패할 수 있는 최대 작업 수를 설정합니다`CANCELED`. 기본 값은 100입니다.

```
deadline bundle submit sample_job --max-failed-tasks-count 10
```

최대 실패 작업 재시도 횟수

이 `max-retries-per-task` 옵션은 작업이 실패하기 전에 다시 시도할 수 있는 최대 횟수를 설정합니다. 작업을 다시 시도하면 해당 상태가 됩니다. `READY` 기본값은 5입니다.

```
deadline bundle submit sample_job --max-retries-per-task 10
```

데드라인 클라우드에서 작업 스케줄링

AWS Deadline Cloud는 작업이 생성된 후 대기열과 연결된 하나 이상의 플릿에서 해당 작업이 처리되도록 예약합니다. 특정 작업을 처리하는 플릿은 플릿에 구성된 기능과 특정 단계의 호스트 요구 사항을 기반으로 선택됩니다.

작업은 최선 작업 우선 순위 (가장 높은 순서에서 가장 낮은 순) 에 따라 스케줄링됩니다. 두 작업의 우선 순위가 같으면 가장 오래된 작업이 먼저 예약됩니다.

다음 섹션에서는 작업 예약 프로세스에 대한 세부 정보를 제공합니다.

플릿 호환성을 결정하세요.

작업이 생성된 후 Deadline Cloud는 작업이 제출된 대기열과 연결된 플릿의 기능과 비교하여 작업의 각 단계에 대한 호스트 요구 사항을 확인합니다. 플릿이 호스트 요구 사항을 충족하면 작업이 READY 상태로 전환됩니다.

대기열과 연결된 플릿이 충족할 수 없는 요구 사항이 작업 단계에 있는 경우 단계 상태는 로 설정됩니다NOT_COMPATIBLE. 또한 작업의 나머지 단계도 취소됩니다.

플릿의 기능은 플릿 수준에서 설정됩니다. 플릿에 속한 작업자가 작업 요구 사항을 충족하더라도 해당 작업단이 작업 요구 사항을 충족하지 않으면 해당 작업자의 작업이 배정되지 않습니다.

다음 작업 템플릿에는 해당 단계에 대한 호스트 요구 사항을 지정하는 단계가 있습니다.

```
name: Sample Job With Host Requirements
specificationVersion: jobtemplate-2023-09
steps:
- name: Step 1
  script:
    actions:
      onRun:
        args:
        - '1'
        command: /usr/bin/sleep
  hostRequirements:
    amounts:
    # Capabilities starting with "amount." are amount capabilities. If they start with
 "amount.worker.",
    # they are defined by the OpenJD specification. Other names are free for custom
 usage.
    - name: amount.worker.vcpu
      min: 4
      max: 8
    attributes:
    - name: attr.worker.os.family
      anyOf:
      - linux
```

다음 기능을 갖춘 플릿으로 이 작업을 예약할 수 있습니다.

```
{
```

```
    "vCpuCount": {"min": 4, "max": 8},
    "memoryMiB": {"min": 1024},
    "osFamily": "linux",
    "cpuArchitectureType": "x86_64"
}
```

다음 기능을 갖춘 플릿에는 이 작업을 예약할 수 없습니다.

```
{
    "vCpuCount": {"min": 4},
    "memoryMiB": {"min": 1024},
    "osFamily": "linux",
    "cpuArchitectureType": "x86_64"
}
    The vCpuCount has no maximum, so it exceeds the maximum vCPU host requirement.

{
    "vCpuCount": {"max": 8},
    "memoryMiB": {"min": 1024},
    "osFamily": "linux",
    "cpuArchitectureType": "x86_64"
}
    The vCpuCount has no minimum, so it doesn't satisfy the minimum vCPU host
 requirement.

{
    "vCpuCount": {"min": 4, "max": 8},
    "memoryMiB": {"min": 1024},
    "osFamily": "windows",
    "cpuArchitectureType": "x86_64"
}
    The osFamily doesn't match.
```

플릿 스케일링

호환되는 서비스 관리 플릿에 작업이 할당되면 플릿은 자동 스케일링됩니다. 플릿의 작업자 수는 플릿을 실행하는 데 사용할 수 있는 작업 수에 따라 달라집니다.

작업이 고객 관리형 플릿에 할당되면 작업자가 이미 존재하거나 이벤트 기반 Auto Scaling을 사용하여 생성될 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 이벤트 처리에 사용을 EventBridge 참조하십시오.

세션

작업의 작업은 하나 이상의 세션으로 나누어집니다. 작업자는 세션을 실행하여 환경을 설정하고 작업을 실행한 다음 환경을 분해합니다. 각 세션은 작업자가 취해야 하는 하나 이상의 작업으로 구성됩니다.

작업자가 섹션 작업을 완료하면 작업자에게 추가 세션 작업을 보낼 수 있습니다. 작업자는 세션의 기존 환경과 작업 첨부 파일을 재사용하여 작업을 더 효율적으로 완료합니다.

작업 첨부 파일은 Deadline Cloud CLI 작업 번들의 일부로 사용하는 제출자가 생성합니다. 명령의 `--attachments` 옵션을 사용하여 작업 첨부 파일을 생성할 수도 있습니다. `create-job` AWS CLI 환경은 특정 대기열에 연결된 대기열 환경과 작업 템플릿에 정의된 작업 단계 환경의 두 위치에서 정의됩니다.

세션 작업 유형에는 네 가지가 있습니다.

- `syncInputJobAttachments`— 입력 작업 첨부 파일을 작업자에게 다운로드합니다.
- `envEnter`— 환경에 대한 `onEnter` 작업을 수행합니다.
- `taskRun`— `onRun` 작업에 대한 작업을 수행합니다.
- `envExit`— 환경에 대한 `onExit` 작업을 수행합니다.

다음 작업 템플릿에는 단계 환경이 있습니다. 여기에는 단계 환경을 설정하는 `onEnter` 정의, 실행할 작업을 정의하는 `onRun` 정의, 단계 환경을 분해하는 `onExit` 정의가 있습니다. 이 작업을 위해 생성되는 세션에는 작업 하나, 하나 이상의 `taskRun` 작업, 그리고 한 개의 `envExit` 동작이 포함됩니다. `envEnter`

```
name: Sample Job with Maya Environment
specificationVersion: jobtemplate-2023-09
steps:
- name: Maya Step
  stepEnvironments:
  - name: Maya
    description: Runs Maya in the background.
    script:
      embeddedFiles:
      - name: initData
        filename: init-data.yaml
        type: TEXT
        data: |
```

```
          scene_file: MyAwesomeSceneFile
          renderer: arnold
          camera: persp
      actions:
        onEnter:
          command: MayaAdaptor
          args:
          - daemon
          - start
          - --init-data
          - file://{{Env.File.initData}}
        onExit:
          command: MayaAdaptor
          args:
          - daemon
          - stop
  parameterSpace:
    taskParameterDefinitions:
    - name: Frame
      range: 1-5
      type: INT
  script:
    embeddedFiles:
    - name: runData
      filename: run-data.yaml
      type: TEXT
      data: |
        frame: {{Task.Param.Frame}}
    actions:
      onRun:
        command: MayaAdaptor
        args:
        - daemon
        - run
        - --run-data
        - file://{{ Task.File.runData }}
```

단계 종속성

Deadline Cloud는 한 단계가 다른 단계가 완료될 때까지 기다렸다가 시작하도록 단계 간 종속성 정의를 지원합니다. 한 단계에 대해 둘 이상의 종속성을 정의할 수 있습니다. 종속성이 있는 단계는 모든 종속성이 완료될 때까지 예약되지 않습니다.

작업 템플릿이 순환 종속성을 정의하면 작업이 거부되고 작업 상태가 로 설정됩니다.
CREATE_FAILED

다음 작업 템플릿은 두 단계로 작업을 생성합니다. StepB에 따라 다릅니다StepA. StepB성공적으로 StepA 완료된 후에만 실행됩니다.

작업이 생성된 후에는 StepA READY 상태가 되고 PENDING 상태가 StepB 됩니다. StepA완료 후 READY 상태로 StepB 이동합니다. StepA실패하거나 취소된 StepA 경우 해당 CANCELED 상태로 StepB 이동합니다.

여러 단계에 대한 종속성을 설정할 수 있습니다. 예를 들어, StepC If는 두 가지 모두에 StepA 종속되며 StepB 다른 두 단계가 완료될 때까지 StepC 시작되지 않습니다.

```
name: Step-Step Dependency Test
specificationVersion: 'jobtemplate-2023-09'
steps:
- name: A
  script:
    actions:
      onRun:
        command: bash
        args: ['{{ Task.File.run }}']
    embeddedFiles:
      - name: run
        type: TEXT
        data: |
          #!/bin/env bash

          set -euo pipefail

          sleep 1
          echo Task A Done!
- name: B
  dependencies:
  - dependsOn: A # This means Step B depends on Step A
  script:
    actions:
      onRun:
        command: bash
        args: ['{{ Task.File.run }}']
    embeddedFiles:
      - name: run
        type: TEXT
```

```
        data: |
          #!/bin/env bash

          set -euo pipefail

          sleep 1
          echo Task B Done!
```

데드라인 클라우드 CLI의 작업 상태

이 항목에서는 AWS Deadline Cloud 명령줄 인터페이스 (Deadline Cloud CLI) 를 사용하여 작업 또는 단계의 상태를 보는 방법을 설명합니다. Deadline Cloud 모니터를 사용하여 작업 또는 단계의 상태를 보려면 을 참조하십시오데드라인 클라우드에서 작업, 단계 및 작업을 보고 관리합니다..

`deadline job get --job-id`데드라인 클라우드 CLI 명령을 사용하여 작업 상태를 확인할 수 있습니다. 명령에 대한 응답에는 작업 또는 단계의 상태와 각 처리 상태의 작업 수가 포함됩니다.

작업을 처음 제출하면 상태는 입니다`CREATE_IN_PROGRESS`. 작업이 검증 검사를 통과하면 작업 상태가 로 바뀝니다`CREATE_COMPLETE`. 그렇지 않으면 상태가 로 변경됩니다`CREATE_FAILED`.

작업이 검증 검사에 실패할 수 있는 몇 가지 가능한 원인은 다음과 같습니다.

- 작업 템플릿은 OpenJD 사양을 따르지 않습니다.
- 작업에 너무 많은 단계가 포함되어 있습니다.
- 작업에 총 작업 수가 너무 많습니다.

작업의 최대 단계 및 작업 수에 대한 할당량을 보려면 Service Quotas 콘솔을 사용하세요. 자세한 정보는 에 대한 할당량 Deadline Cloud을 참조하세요.

내부 서비스 오류로 인해 작업이 생성되지 않을 수도 있습니다. 이 경우 작업의 상태 코드는 `INTERNAL_ERROR` 이며 상태 메시지 필드는 보다 자세한 설명을 제공합니다.

다음 Deadline Cloud CLI 명령을 사용하여 작업에 대한 세부 정보를 볼 수 있습니다. 다음 예시에서는 자체 *jobID*정보로 바꾸십시오.

```
deadline job get --job-id jobId
```

`deadline job get`명령의 응답은 다음과 같습니다.

```
jobId: jobId
name: Sample Job
lifecycleStatus: CREATE_COMPLETE
lifecycleStatusMessage: Job creation completed successfully
priority: 50
createdAt: 2024-03-26 18:11:19.065000+00:00
createdBy: Test User
startedAt: 2024-03-26 18:12:50.710000+00:00
taskRunStatus: STARTING
taskRunStatusCounts:
  PENDING: 0
  READY: 5
  RUNNING: 0
  ASSIGNED: 0
  STARTING: 0
  SCHEDULED: 0
  INTERRUPTING: 0
  SUSPENDED: 0
  CANCELED: 0
  FAILED: 0
  SUCCEEDED: 0
  NOT_COMPATIBLE: 0
maxFailedTasksCount: 100
maxRetriesPerTask: 5
```

작업 또는 단계의 각 작업에는 상태가 있습니다. 작업 상태를 조합하여 작업 및 단계의 전체 상태를 제공합니다. 각 상태의 작업 수는 응답 `taskRunStatusCounts` 필드에 보고됩니다.

작업 또는 단계의 상태는 해당 작업의 상태에 따라 달라집니다. 상태는 이러한 상태의 작업에 따라 순서대로 결정됩니다. 단계 상태는 작업 상태와 동일하게 결정됩니다.

다음 목록은 상태를 설명합니다.

`NOT_COMPATIBLE`

작업의 작업 중 하나를 완료할 수 있는 플릿이 없기 때문에 작업이 팜과 호환되지 않습니다.

`RUNNING`

한 명 이상의 작업자가 작업에서 작업을 실행하고 있습니다. 실행 중인 작업이 하나 이상 있으면 작업이 표시됩니다`RUNNING`.

ASSIGNED

한 명 이상의 작업자에게 다음 작업으로 해당 작업의 작업이 배정됩니다. 환경 (있는 경우) 이 설정됩니다.

STARTING

한 명 이상의 작업자가 작업을 실행하기 위한 환경을 설정하고 있습니다.

SCHEDULED

해당 작업에 대한 작업은 작업자의 다음 작업으로 한 명 이상의 작업자에게 예약됩니다.

READY

해당 작업에 대한 하나 이상의 작업을 처리할 준비가 되었습니다.

INTERRUPTING

작업 중 하나 이상의 작업이 중단되었습니다. 작업 상태를 수동으로 업데이트하면 중단이 발생할 수 있습니다. 또한 Amazon Elastic Compute Cloud (Amazon EC2) 스팟 가격 변동으로 인한 중단에 대한 대응으로 발생할 수도 있습니다.

FAILED

작업 중 하나 이상의 작업이 성공적으로 완료되지 않았습니다.

CANCELED

작업에 있는 하나 이상의 작업이 취소되었습니다.

SUSPENDED

작업 중 하나 이상의 작업이 일시 중단되었습니다.

PENDING

작업의 작업이 다른 리소스를 사용할 수 있을 때까지 대기 중입니다.

SUCCEEDED

작업의 모든 작업이 성공적으로 처리되었습니다.

데드라인 클라우드에서 작업 수정

다음 AWS Command Line Interface (AWS CLI) `update` 명령을 사용하여 작업 구성을 수정하거나 작업, 단계 또는 작업의 대상 상태를 설정할 수 있습니다.

- `aws deadline update-job`
- `aws deadline update-step`
- `aws deadline update-task`

다음 명령 예제에서는 각 `update` 명령을 사용자 고유의 *`user input placeholder`*정보로 바꾸십시오.

Deadline Cloud 모니터를 사용하여 작업 구성을 수정할 수도 있습니다. 자세한 정보는 데드라인 클라우드에서 작업, 단계 및 작업을 보고 관리합니다.을 참조하세요.

Example — 작업 재개

단계 종속성이 없는 한 작업의 모든 작업이 해당 `READY` 상태로 전환됩니다. 종속성이 있는 단계는 복원되는 `PENDING` 대로 `READY` 또는 복원되는 대로 전환됩니다.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status PENDING
```

Example — 작업 취소

상태가 `SUCCEEDED` 없거나 표시되지 `FAILED` 않은 작업의 모든 작업`CANCELED`.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status CANCELED
```

Example — 작업을 실패로 표시

`SUCCEEDED`현재 상태인 작업의 모든 작업은 변경되지 않습니다. 다른 모든 작업은 표시됩니다`FAILED`.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
```

```
--job-id jobID \
--target-task-run-status FAILED
```

Example — 작업을 성공으로 표시

작업의 모든 작업이 SUCCEEDED 상태로 이동합니다.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status SUCCEEDED
```

Example — 작업 일시 중지

SUCCEEDEDCANCELED, 또는 FAILED 상태의 작업에 있는 작업은 변경되지 않습니다. 다른 모든 작업은 표시됩니다SUSPENDED.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--target-task-run-status SUSPENDED
```

Example — 작업 우선 순위 변경

작업의 우선 순위를 업데이트하여 예약된 순서를 변경합니다. 일반적으로 우선 순위가 높은 작업이 먼저 예약됩니다.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--priority 100
```

Example — 허용된 실패한 작업 수 변경

나머지 작업이 취소되기 전에 작업에 발생할 수 있는 최대 실패 작업 수를 업데이트합니다.

```
aws deadline update-job \
--farm-id farmID \
```

```
--queue-id queueID \
--job-id jobID \
--max-failed-tasks-count 200
```

Example — 허용되는 작업 재시도 횟수 변경

작업이 실패하기 전에 작업을 재시도할 수 있는 최대 횟수를 업데이트합니다. 최대 재시도 횟수에 도달한 작업은 이 값을 늘릴 때까지 대기열에 추가할 수 없습니다.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--max-retries-per-task 10
```

Example — 작업 아카이브

작업의 수명 주기 상태를 로 ARCHIVED 업데이트합니다. 보관된 작업은 예약하거나 수정할 수 없습니다. FAILED, CANCELEDSUCCEEDED, 또는 SUSPENDED 상태에 있는 작업만 보관할 수 있습니다.

```
aws deadline update-job \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--lifecycle-status ARCHIVED
```

Example — 단계 재개

단계 종속성이 없는 한 단계의 모든 작업이 READY 상태로 전환됩니다. 종속성이 있는 단계의 작업은 READY 또는 PENDING 중 하나로 전환되고 작업이 복원됩니다.

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status PENDING
```

Example — 단계 취소

단계에 있는 모든 작업 중 상태가 SUCCEEDED 없거나 FAILED 표시되지 않은 모든 작업CANCELED.

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status CANCELED
```

Example — 단계를 실패로 표시

해당 단계의 모든 작업 중 상태가 SUCCEEDED 변경되지 않은 상태로 유지됩니다. 다른 모든 작업은 표시됩니다FAILED.

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status FAILED
```

Example — 단계를 성공으로 표시

단계의 모든 작업이 표시됩니다SUCCEEDED.

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--target-task-run-status SUCCEEDED
```

Example — 단계 일시 중지

SUCCEEDEDCANCELED, 또는 FAILED 상태의 단계에 있는 작업은 변경되지 않습니다. 다른 모든 작업은 표시됩니다SUSPENDED.

```
aws deadline update-step \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
```

```
--target-task-run-status SUSPENDED
```

Example — 작업 상태 변경

update-taskDeadline Cloud CLI 명령을 사용하면 작업이 지정된 상태로 전환됩니다.

```
aws deadline update-task \
--farm-id farmID \
--queue-id queueID \
--job-id jobID \
--step-id stepID \
--task-id taskID \
--target-task-run-status SUCCEEDED | SUSPENDED | CANCELED | FAILED | PENDING
```

데드라인 클라우드가 작업을 처리하는 방법

AWS 데드라인 클라우드는 작업을 처리하기 위해 오픈 잡 디스크립션 (OpenJD) 작업 템플릿을 사용하여 필요한 리소스를 결정합니다. Deadline Cloud는 대기열과 관련된 플릿 중에서 작업에 적합한 작업자를 선택합니다. 선택한 작업자는 단계에 필요한 모든 역량 특성을 충족합니다.

그런 다음 Deadline Cloud는 작업자에게 지침을 보내 단계에 대한 세션을 설정합니다. 작업을 실행하려면 단계에 필요한 소프트웨어를 작업자 인스턴스에서 사용할 수 있어야 합니다. 플릿의 조정 설정에 용량이 있는 경우 서비스는 여러 작업자의 세션을 열 수 있습니다.

Amazon Machine Image(AMI) 에서 소프트웨어를 설정하거나 작업자가 저장소 또는 패키지 관리자에서 런타임에 소프트웨어를 로드할 수 있습니다. 큐, 작업 또는 단계 환경을 사용하여 원하는 소프트웨어를 배포할 수 있습니다.

Deadline Cloud 서비스는 OpenJD 템플릿을 사용하여 작업에 필요한 단계와 각 단계에 필요한 작업을 결정합니다. 일부 단계는 다른 단계에 종속되므로 Deadline Cloud는 단계를 완료하는 순서를 결정합니다. 그런 다음 Deadline Cloud는 각 단계의 작업을 작업자에게 보내 처리하도록 합니다. 작업이 완료되면 서비스에서 동일한 세션에 다른 작업을 전송하거나 작업자가 새 세션을 시작할 수 있습니다.

데드라인 클라우드 모니터, 데드라인 클라우드 명령줄 인터페이스 (데드라인 클라우드 CLI) 또는 에서 작업 진행 상황을 추적할 수 있습니다. AWS CLI모니터 사용에 대한 자세한 내용은 을 참조하십시오데드라인 클라우드 모니터 사용. 데드라인 클라우드 CLI 사용에 대한 자세한 내용은 을 참조하십시오. 데드라인 클라우드 CLI의 작업 상태

각 단계의 모든 작업이 완료되면 작업이 완료되고 출력을 워크스테이션에 다운로드할 수 있습니다. 작업이 완료되지 않았더라도 완료된 각 단계 및 작업의 출력을 다운로드할 수 있습니다.

데드라인 클라우드는 제출된 지 120일이 지난 작업을 삭제합니다. 작업이 제거되면 해당 작업과 관련된 모든 단계 및 작업도 제거됩니다. 작업을 다시 실행해야 하는 경우 해당 작업에 대한 OpenJD 템플릿을 다시 제출하십시오.

데드라인 클라우드 작업 문제 해결

AWS Deadline Cloud의 작업과 관련된 일반적인 문제에 대한 자세한 내용은 다음 항목을 참조하십시오.

주제

- 작업 생성이 실패한 이유는 무엇인가요?
- 내 작업이 호환되지 않는 이유는 무엇인가요?
- 작업이 준비되지 않은 이유는 무엇인가요?
- 작업이 실패한 이유는 무엇인가요?
- 내 단계가 보류 중인 이유는 무엇인가요?

작업 생성이 실패한 이유는 무엇인가요?

작업이 검증 검사에 실패할 수 있는 몇 가지 원인은 다음과 같습니다.

- 작업 템플릿은 OpenJD 사양을 따르지 않습니다.
- 작업에 너무 많은 단계가 포함되어 있습니다.
- 작업에 총 작업 수가 너무 많습니다.
- 내부 서비스 오류로 인해 작업이 생성되지 않았습니다.

작업의 최대 단계 및 작업 수에 대한 할당량을 보려면 Service Quotas 콘솔을 사용하세요. 자세한 정보는 에 대한 할당량 Deadline Cloud을 참조하세요.

내 작업이 호환되지 않는 이유는 무엇인가요?

작업이 대기열과 호환되지 않는 일반적인 이유는 다음과 같습니다.

- 작업이 제출된 대기열에 연결된 플릿이 없습니다. Deadline Cloud 모니터를 열고 대기열에 연결된 플릿이 있는지 확인합니다. 대기열을 보는 방법에 대한 자세한 내용은 을 참조하십시오. 데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다.

- 이 작업의 호스트 요구 사항은 해당 대기열과 연결된 플릿이 충족하지 못합니다. 확인하려면 작업 템플릿의 `hostRequirements` 항목을 팜의 플릿 구성과 비교하십시오. 플릿 중 하나가 호스트 요구 사항을 충족하는지 확인하십시오. 플릿 호환성에 대한 자세한 내용은 을 참조하십시오. 플릿 호환성을 결정하세요. 플릿 구성을 보려면 을 참조하십시오데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다..

작업이 준비되지 않은 이유는 무엇인가요?

작업이 현재 `READY` 상태에서 중단된 것처럼 보이는 이유는 다음과 같습니다.

- 대기열과 연결된 플릿의 최대 작업자 수는 0으로 설정되어 있습니다. 확인하려면 을 참조하십시오데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다..
- 대기열에 우선 순위가 더 높은 작업이 있습니다. 확인하려면 을 참조하십시오데드라인 클라우드에서 대기열 및 차량 세부 정보를 볼 수 있습니다..
- 고객 관리 플릿의 경우 Auto Scaling 구성을 확인하십시오. 자세한 정보는 데드라인 클라우드 스케일 권장 기능을 사용하여 Amazon EC2 플릿을 자동 확장하십시오.을 참조하세요.

작업이 실패한 이유는 무엇인가요?

작업은 여러 가지 이유로 실패할 수 있습니다. 문제를 검색하려면 Deadline Cloud 모니터를 열고 실패한 작업을 선택합니다. 실패한 작업을 선택한 다음 해당 작업의 로그를 확인하세요. 지침은 데드라인 클라우드에서 로그 보기 섹션을 참조하세요.

- 라이선스 오류가 나타나거나 소프트웨어에 유효한 라이선스가 없어서 워터마크가 나타나는 경우 작업자가 필요한 라이선스 서버에 연결할 수 있는지 확인하세요. 자세한 정보는 고객 관리 플릿을 라이선스 엔드포인트에 연결을 참조하세요.

내 단계가 보류 중인 이유는 무엇인가요?

하나 이상의 종속성이 완료되지 않아도 단계가 `PENDING` 상태를 유지할 수 있습니다. Deadline Cloud 모니터를 사용하여 종속성 상태를 확인할 수 있습니다. 지침은 데드라인 클라우드에서 단계 보기 섹션을 참조하세요.

데드라인 클라우드용 파일 스토리지

작업자는 작업을 처리하는 데 필요한 입력 파일이 들어 있는 저장소 위치와 결과물을 저장하는 위치에 액세스할 수 있어야 합니다. AWS Deadline Cloud는 저장 위치에 대한 두 가지 옵션을 제공합니다.

- Deadline Cloud는 작업 첨부 파일을 통해 워크스테이션과 Deadline Cloud 작업자 간에 작업에 대한 입력 및 출력 파일을 주고 받습니다. 데드라인 클라우드는 파일 전송을 활성화하기 위해 사용자의 Amazon Simple Storage Service (Amazon S3) 버킷을 사용합니다. AWS 계정

  서비스 관리형 플릿에서 작업 첨부 파일을 사용하는 경우 가상 사설망 (VPN) 에 가상 파일 시스템 (VFS) 을 설정할 수 있습니다. 그러면 작업자는 필요할 때만 파일을 로드할 수 있습니다.
- 공유 스토리지를 사용하면 운영 체제와 파일을 공유하여 파일에 대한 액세스를 제공할 수 있습니다.

  플랫폼 간 공유 스토리지를 사용하는 경우 작업자가 서로 다른 두 운영 체제 간에 파일 경로를 매핑할 수 있도록 스토리지 프로필을 생성할 수 있습니다.

주제

- 데드라인 클라우드의 Job 첨부
- 데드라인 클라우드의 공유 스토리지

데드라인 클라우드의 Job 첨부

작업 첨부를 사용하면 워크스테이션과 AWS Deadline Cloud 간에 파일을 주고 받을 수 있습니다. 작업 첨부를 사용하면 파일에 대한 Amazon S3 버킷을 수동으로 설정할 필요가 없습니다. 대신 Deadline Cloud 콘솔로 대기열을 생성할 때 작업 첨부를 위한 버킷을 선택합니다.

데드라인 클라우드에 작업을 처음 제출하면 해당 작업에 대한 모든 파일이 데드라인 클라우드로 전송됩니다. 후속 제출의 경우 변경된 파일만 전송되므로 시간과 대역폭이 모두 절약됩니다.

처리가 완료되면 작업 세부 정보 페이지에서 또는 Deadline Cloud CLI `deadline job download-output` 명령을 사용하여 결과를 다운로드할 수 있습니다.

여러 대기열에 동일한 S3 버킷을 사용할 수 있습니다. 각 대기열에 다른 루트 접두사를 설정하여 버킷의 첨부 파일을 정리합니다.

콘솔로 대기열을 생성할 때 기존 AWS Identity and Access Management (IAM) 역할을 선택하거나 콘솔에서 새 역할을 생성하도록 할 수 있습니다. 콘솔이 역할을 생성하면 해당 대기열에 지정된 버킷에

액세스할 수 있는 권한이 설정됩니다. 기존 역할을 선택하는 경우 역할에 S3 버킷에 액세스할 수 있는 권한을 부여해야 합니다.

작업 첨부 S3 버킷의 암호화

Job 첨부 파일은 기본적으로 S3 버킷에서 자동으로 암호화됩니다. 이 접근 방식은 무단 액세스로부터 정보를 보호하는 데 도움이 됩니다. Deadline Cloud에서 제공하는 키로 파일을 암호화하기 위해 별도의 조치를 취하지 않아도 됩니다. 자세한 내용은 Amazon S3 사용 설명서의 Amazon S3가 이제 모든 새 객체를 자동으로 암호화함을 참조하십시오.

자체 고객 관리 AWS Key Management Service 키를 사용하여 작업 첨부 파일이 포함된 S3 버킷을 암호화할 수 있습니다. 이렇게 하려면 버킷과 연결된 대기열의 IAM 역할을 수정하여 에 대한 액세스를 허용해야 합니다. AWS KMS key

대기열 역할의 IAM 정책 편집기를 열려면

1. 데드라인 클라우드 콘솔에 AWS Management Console 로그인하고 엽니다. 메인 페이지의 시작하기 섹션에서 팜 보기를 선택합니다.
2. 팜 목록에서 수정할 큐가 있는 팜을 선택합니다.
3. 큐 목록에서 수정할 큐를 선택합니다.
4. 큐 세부 정보 섹션에서 서비스 역할을 선택하여 서비스 역할에 대한 IAM 콘솔을 엽니다.

그런 다음 다음 절차를 완료하십시오.

에 대한 권한으로 역할 정책을 업데이트하려면 AWS KMS

1. 권한 정책 목록에서 역할에 대한 정책을 선택합니다.
2. 이 정책 섹션에 정의된 권한에서 편집을 선택합니다.
3. 새 문 추가를 선택합니다.
4. 다음 정책을 복사하여 편집기에 붙여넣습니다. *Region*, *accountID*, *keyID*를 원하는 값으로 변경하십시오.

```
{
   "Effect": "Allow",
   "Action": [
       "kms:Decrypt",
       "kms:DescribeKey",
       "kms:GenerateDataKey"
```

```
    ],
    "Resource": [
        "arn:aws:kms:Region:accountID:key/keyID"
    ]
}
```

5. 다음을 선택합니다.
6. 정책 변경 내용을 검토한 다음 만족하면 변경 내용 저장을 선택합니다.

S3 버킷의 작업 첨부 파일 관리

Deadline Cloud는 작업에 필요한 작업 첨부 파일을 S3 버킷에 저장합니다. 이러한 파일은 시간이 지남에 따라 누적되므로 Amazon S3 비용이 증가합니다. 비용을 줄이기 위해 S3 수명 주기 구성을 S3 버킷에 적용할 수 있습니다. 이 구성은 버킷의 파일을 자동으로 삭제할 수 있습니다. S3 버킷이 계정에 있으므로 언제든지 S3 수명 주기 구성을 수정하거나 제거하도록 선택할 수 있습니다. 자세한 내용은 Amazon S3 사용 설명서의 S3 수명 주기 구성 예를 참조하십시오.

보다 세분화된 S3 버킷 관리 솔루션을 AWS 계정 위해 마지막으로 액세스한 시간을 기준으로 S3 버킷의 객체가 만료되도록 설정할 수 있습니다. 자세한 내용은 AWS 아키텍처 블로그에서 비용 절감을 위한 마지막 액세스 날짜를 기준으로 Amazon S3 객체 만료를 참조하십시오.

데드라인: 클라우드 가상 파일 시스템

AWS Deadline Cloud의 작업 첨부 파일에 대한 가상 파일 시스템 지원을 통해 작업자의 클라이언트 소프트웨어가 Amazon Simple Storage Service와 직접 통신할 수 있습니다. 작업자는 처리 전에 모든 파일을 다운로드하는 대신 필요할 때만 파일을 로드할 수 있습니다. 파일은 로컬에 저장됩니다. 이 방법을 사용하면 두 번 이상 사용된 자산을 여러 번 다운로드하지 않아도 됩니다. 작업이 완료되면 모든 파일이 제거됩니다.

- 가상 파일 시스템은 특정 직무 프로필의 성능을 크게 향상시킵니다. 일반적으로 전체 파일 중 하위 집합이 작고 작업자 수가 많을수록 효과가 가장 큽니다. 작업자 수가 적은 소수의 파일이라도 처리 시간은 거의 같습니다.
- 가상 파일 시스템 지원은 서비스 관리 플릿의 Linux 작업자만 사용할 수 있습니다.
- Deadline Cloud 가상 파일 시스템은 다음 작업을 지원하지만 POSIX와 호환되지는 않습니다.
  - 파일`create`,`delete`,`open`,`close`,`read`,`write`,`append`,`truncate`,`rename`,`move`,`copy`,`stat`, 및 `fsync` `falloc`

  - 디렉터리 `createdelete`,`rename`,`move`,`copy`, 및 `stat`
- 가상 파일 시스템은 작업에서 대규모 데이터 세트의 일부만 액세스하는 경우 데이터 전송을 줄이고 성능을 개선하도록 설계되었으며 모든 워크로드에 최적화되어 있지는 않습니다. 프로덕션 작업을 실행하기 전에 워크로드를 테스트해야 합니다.

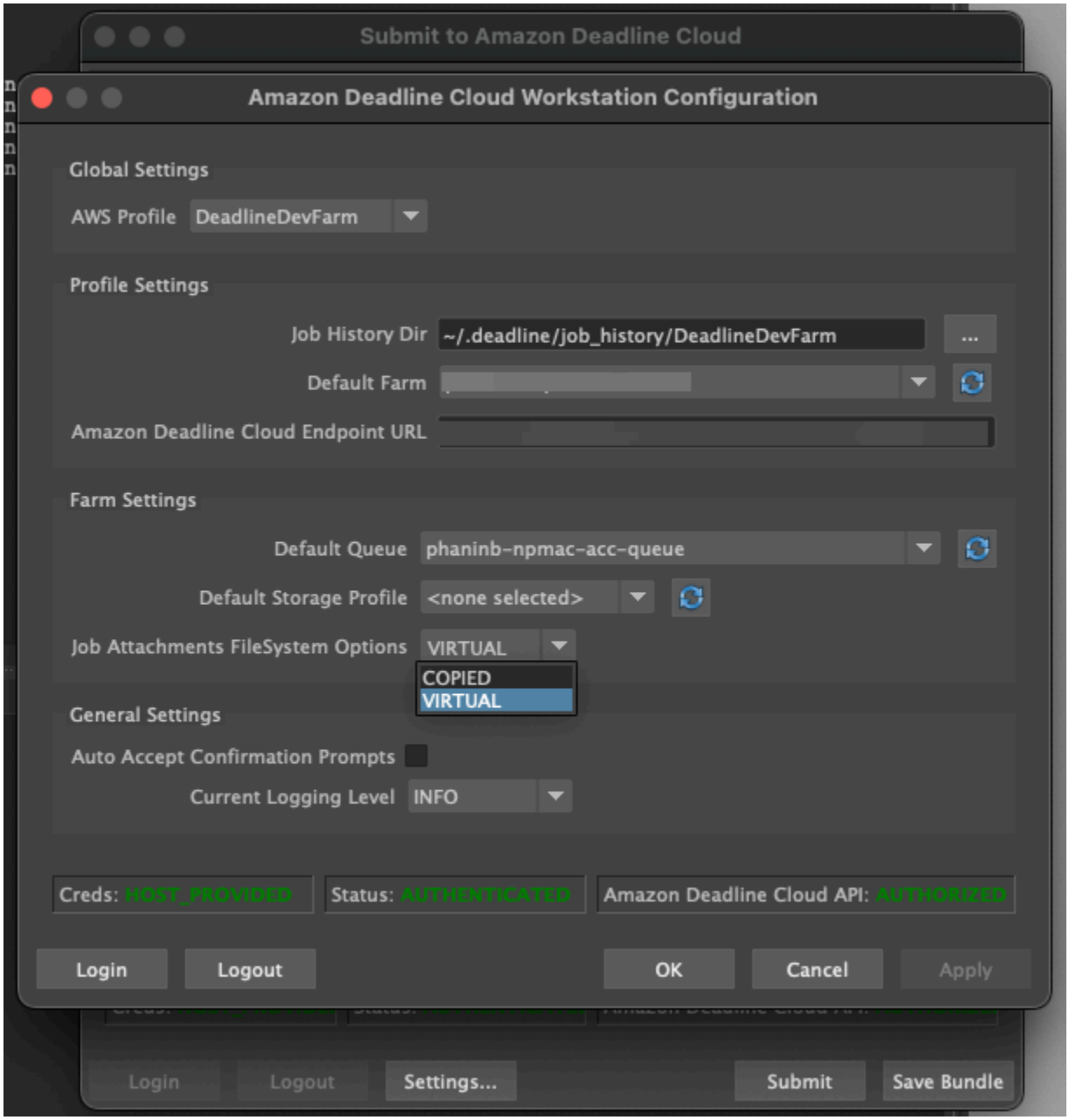
VFS 지원을 활성화합니다.

각 작업에 대해 VFS (가상 파일 시스템 지원) 가 활성화됩니다. 다음과 같은 경우 작업은 기본 작업 첨부 프레임워크로 폴백됩니다.

- 작업자 인스턴스 프로필은 가상 파일 시스템을 지원하지 않습니다.
- 문제가 있으면 가상 파일 시스템 프로세스를 시작할 수 없습니다.
- 가상 파일 시스템을 마운트할 수 없습니다.

제출자를 사용하여 가상 파일 시스템 지원을 활성화하려면

1. 작업을 제출할 때 설정 버튼을 선택하여 AWS Deadline Cloud 워크스테이션 구성 패널을 엽니다.
2. [작업 첨부 파일 시스템 옵션] 드롭다운에서 [VIRTUAL] 을 선택합니다.



3. 변경 내용을 저장하려면 확인을 선택합니다.

를 사용하여 가상 파일 시스템 지원을 활성화하려면 AWS CLI

- 저장된 작업을 제출할 때는 다음 명령을 사용하십시오.

```
deadline bundle submit-job --job-attachments-file-system VIRTUAL
```

특정 작업에 대해 가상 파일 시스템이 성공적으로 시작되었는지 확인하려면 Amazon CloudWatch Logs에서 로그를 검토하십시오. 다음 메시지를 찾아보십시오.

```
Using mount_point mount_point
Launching vfs with command command
Launched vfs as pid PID number
```

로그에 다음 메시지가 포함된 경우 가상 파일 시스템 지원이 비활성화됩니다.

```
Virtual File System not found, falling back to COPIED for JobAttachmentsFileSystem.
```

가상 파일 시스템 지원 문제 해결

데드라인 클라우드 모니터를 사용하여 가상 파일 시스템의 로그를 볼 수 있습니다. 지침은 데드라인 클라우드에서 로그 보기 섹션을 참조하세요.

또한 가상 파일 시스템 로그는 작업자 에이전트 출력과 공유된 대기열과 연결된 CloudWatch 로그 그룹에도 전송됩니다.

데드라인 클라우드의 공유 스토리지

공유 스토리지를 사용하기 위해 작업자는 운영 체제 파일 공유 시스템을 사용하여 작업 입력 및 출력을 위한 공유 스토리지 공간에 액세스합니다.

파일을 공유하는 데 사용하는 실제 방법은 운영 체제 및 네트워크에 공유 저장소를 구현하는 방식에 따라 다릅니다. 파일 공유를 구성하고 요구 사항을 충족하는지 확인하는 방법은 사용자의 책임입니다.

시스템 간 파일 공유 솔루션을 사용하는 경우 저장소 프로필을 사용하여 파일 시스템 간 Linux 파일 위치를 매핑할 수 있습니다. Windows

데드라인 클라우드의 스토리지 프로필

스토리지 프로필을 사용하면 플랫폼 간 공유 스토리지를 사용하여 팜을 설정할 수 있습니다. 스토리지 프로필은 제출된 워크스테이션과 다른 운영 체제를 사용하는 작업자에서 처리된 작업의 운영 체제 간 경로를 매핑합니다.

워크스테이션과 작업자 간에 운영 체제가 혼합되어 있는 고객 관리형 플릿을 사용할 때는 스토리지 프로파일이 필요합니다. 서비스 관리 플릿에서는 스토리지 프로파일이 지원되지 않습니다.

스토리지 프로필을 생성한 후에는 해당 프로필을 사용하는 대기열과 플릿에 대한 액세스 권한을 부여해야 합니다.

스토리지 프로필을 만들려면

1. 데드라인 클라우드 콘솔을 엽니다.
2. 시작하기에서 데드라인 클라우드 대시보드로 이동을 선택합니다.
3. 팜을 선택한 다음 스토리지 프로필 탭을 선택합니다.
4. 스토리지 프로필 생성을 선택합니다.
5. 드롭다운에서 운영 체제를 선택합니다.
6. 프로필 이름을 입력합니다. 명확한 이름을 지정하면 작업을 제출할 때 사용할 스토리지 프로필을 선택하는 데 도움이 됩니다.
7. 경로 이름에는 작업을 제출하는 워크스테이션에 있는 작업 데이터의 루트 위치를 입력합니다.
8. 스토리지 유형 선택:
   - 로컬은 작업자와 워크스테이션 간에 공유되지 않는 파일 위치를 말합니다. 작업 첨부 파일로 업로드됩니다.
   - 공유란 작업자와 워크스테이션 간에 공유되는 스토리지를 말합니다. 공유 스토리지의 파일은 작업 첨부 파일로 업로드되지 않습니다.
9. 파일 시스템 위치 경로를 제공하십시오. 작업 데이터의 루트 디렉터리입니다.
10. 생성을 선택합니다.

스토리지 프로필을 생성한 후에는 새 프로필을 사용하도록 대기열과 고객 관리형 플릿을 수정해야 합니다. 스토리지 프로필에 대한 액세스를 허용하려면 이전 절차를 완료한 후 다음 절차를 사용하십시오.

대기열 및 고객 관리 플릿이 스토리지 프로필을 사용할 수 있도록 하려면

1. 큐 또는 플릿 탭을 선택합니다.
2. 수정할 대기열 또는 플릿을 선택합니다.
3. 스토리지 프로필 수정을 선택합니다.
4. 허용할 스토리지 프로필을 선택하고 해당 프로필에서 파일 시스템 위치를 선택합니다.
5. 변경 사항 저장를 선택합니다.

데드라인 클라우드의 예산 및 사용량 관리

AWS Deadline Cloud 예산 관리자 및 사용량 탐색기는 비용 변수에 대한 사용 가능한 정보를 기반으로 Deadline Cloud를 사용하는 데 드는 대략적인 비용을 제공하는 비용 관리 도구입니다. 비용 관리 도구는 Deadline Cloud 및 기타 AWS 서비스를 실제로 사용하는 데 필요한 금액을 보장하지 않습니다.

데드라인 클라우드의 비용을 관리하는 데 도움이 되도록 다음 기능을 사용할 수 있습니다.

- 예산 관리자 - Deadline Cloud 예산 관리자를 사용하면 프로젝트 비용 관리에 도움이 되는 예산을 만들고 편집할 수 있습니다.
- 사용량 탐색기 - Deadline Cloud 사용량 탐색기를 사용하면 사용된 리소스 수와 해당 AWS 리소스의 예상 비용을 확인할 수 있습니다.

비용 가정

Deadline Cloud 비용 관리 도구에서 사용하는 기본 계산은 다음과 같습니다.

```
Cost per job =
    (CMF run time x CMF compute rate) +
    (SMF run time x SMF compute rate) +
    (License run time x license rate)
```

- 실행 시간은 시작 시간부터 종료 시간까지 작업에 있는 모든 작업의 합계입니다.
- 계산 속도는 서비스 관리 플릿의 AWS Deadline Cloud 요금에 따라 결정됩니다. 고객 관리형 플릿의 경우 컴퓨팅 요금은 작업자 시간당 1 USD로 추정됩니다.
- 라이선스 요금은 Deadline Cloud 기본 라이선스 가격에 따라 결정됩니다. 추가 등급은 포함되지 않습니다. 라이선스 가격에 대한 자세한 내용은 AWS Deadline Cloud 가격을 참조하십시오.

Deadline Cloud 비용 관리 도구의 예상 비용은 여러 가지 이유로 실제 비용과 다를 수 있습니다. 일반적인 이유는 다음과 같습니다.

- 고객 소유 리소스 및 가격. 자체 리소스를 온프레미스 또는 다른 클라우드 제공업체로부터 AWS 가져오거나 외부에서 가져올 수 있습니다. 이러한 리소스의 실제 비용은 계산되지 않습니다.
- 유휴 근로자 비용. 최소 인스턴스 수가 0보다 큰 플릿의 경우 유휴 작업자는 계산에 포함되지 않습니다.

- 프로모션 크레딧, 할인 및 맞춤형 가격 계약. 비용 관리 도구에는 프로모션 크레딧, 비공개 가격 계약 또는 기타 할인은 고려되지 않습니다. 추정치에 포함되지 않은 다른 할인을 받을 수 있습니다.
- 자산 보관. 자산 스토리지는 예상 비용 및 사용량에 포함되지 않습니다.
- 가격 변동. AWS 대부분의 서비스에 대한 pay-as-you-go 가격을 제공합니다. 시간이 지남에 따라 가격이 변경될 수 있습니다. 비용 관리 도구는 공개된 up-to-date 가격을 가장 많이 사용하지만 변경 후에는 지연이 발생할 수 있습니다.
- 세금. 비용 관리 도구에는 서비스 구매에 적용되는 세금이 포함되지 않습니다.
- 반올림. 비용 관리 도구는 가격 데이터를 수학적으로 반올림합니다.
- 통화. 예상 비용은 미국 달러로 산정됩니다. 글로벌 환율은 시간에 따라 달라집니다. 추정치를 현재 환율의 다른 통화로 환산하는 경우 환율 변동이 추정치에 영향을 미칩니다.
- 외부 라이선스. 사전 구매한 라이선스 (자체 라이선스 지참) 를 사용하기로 선택한 경우 Deadline Cloud 비용 관리 도구에서 이 비용을 계산할 수 없습니다.

데드라인 클라우드 예산 관리자 사용

Deadline Cloud 예산 관리자를 사용하면 대기열, 플릿 또는 팜과 같은 특정 리소스에 대한 지출을 관리할 수 있습니다. 예산 금액 및 한도를 생성하고, 예산 대비 추가 지출을 줄이거나 중단하는 데 도움이 되는 자동 조치를 설정할 수 있습니다.

다음 섹션에서는 Deadline Cloud 예산 관리자를 사용하는 단계를 설명합니다.

주제

- 전제 조건
- 액세스 예산 관리자
- 예산 생성
- 예산 보기
- 예산 편집
- 예산 비활성화하기

전제 조건

Deadline Cloud 예산 관리자를 사용하려면 OWNER 액세스 수준이 있어야 합니다. OWNER권한을 부여하려면 다음 단계를 따르세요데드라인 클라우드에서의 사용자 관리.

액세스 예산 관리자

Deadline Cloud 예산 관리자에 액세스하려면 다음 절차를 사용하십시오.

1. 데드라인 클라우드 콘솔에 AWS Management Console 로그인하고 엽니다.
2. [팜 보기] 를 선택합니다.
3. 정보를 얻으려는 팜을 찾은 다음 작업 관리를 선택합니다. 데드라인 클라우드 모니터가 새 탭에서 열립니다.
4. 데드라인 클라우드 모니터의 왼쪽 탐색 창에서 예산을 선택합니다.

예산 관리자 요약 페이지에는 활성 및 비활성 예산 목록이 모두 표시됩니다.

- 활성 예산은 선택한 리소스 (대기열) 를 기준으로 추적됩니다.
- 비활성 예산이 만료되었거나 사용자에 의해 취소되었으므로 더 이상 이 예산 한도를 기준으로 비용을 추적하지 않습니다.

예산을 선택하면 예산 요약 페이지에 예산에 대한 기본 정보가 포함됩니다. 제공되는 정보에는 예산 이름, 상태, 자원, 잔여 비율, 잔여 금액, 총 예산, 시작 날짜 및 종료 날짜가 포함됩니다.

예산 생성

예산을 생성하려면 다음 절차를 사용하십시오.

1. 아직 로그인하지 않았다면 에 로그인하여 Deadline Cloud 콘솔을 열고 팜을 선택한 다음 작업 관리를 선택합니다. AWS Management Console
2. 예산 관리자 페이지에서 예산 생성을 선택합니다.
3. 세부 정보 섹션에서 예산의 예산 이름을 입력합니다.
4. (선택 사항) 설명 필드에 예산에 대한 명확하고 간략한 설명을 입력합니다.
5. 리소스에서 대기열 드롭다운을 선택하여 예산을 생성하려는 대기열을 찾아 선택합니다.
6. 기간의 경우 다음 단계를 완료하여 예산 시작 및 종료 날짜를 설정합니다.

    a. 시작 날짜에 YYYY/MM/DD 형식으로 예산 추적의 첫 번째 날짜를 입력하거나 달력 아이콘을 선택하고 날짜를 선택합니다.

    기본 시작 날짜는 예산이 생성된 날짜입니다.

b. 종료 날짜의 경우 예산 추적의 마지막 날짜를 YYYY/MM/DD 형식으로 입력하거나 달력 아이콘을 선택하고 날짜를 선택합니다.

   기본 종료 날짜는 시작 날짜로부터 120일입니다.

7. 예산 금액에는 예산의 달러 금액을 입력합니다.
8. (선택 사항) 한도 알림을 생성하는 것이 좋습니다. 제한 조치 섹션에서는 예산에 특정 금액이 남아 있을 때 발생하는 자동 작업을 구현할 수 있습니다. 이렇게 하려면 다음 단계를 완료하세요.

   a. 새 작업 추가를 선택합니다.

   b. 잔액에 액션을 시작하려는 달러 금액을 입력합니다.

   c. 액션 드롭다운에서 원하는 액션을 선택합니다. 액션에는 다음이 포함됩니다.

      - 현재 작업을 완료한 후 중지 — 임계값에 도달했을 때 현재 실행 중인 모든 작업은 완료될 때까지 계속 실행되며 비용이 발생합니다.
      - 작업 즉시 중지 — 임계량에 도달하면 모든 작업이 즉시 취소됩니다.

   d. 추가 한도 알림을 생성하려면 새 작업 추가를 선택하고 이전 두 단계를 반복합니다.

9. 예산 생성을 선택합니다. 예산 관리자 페이지가 나타납니다. 새로 생성된 예산은 활성 예산 탭에 표시됩니다.

예산 보기

예산을 생성한 후에는 예산 관리자 페이지에서 예산을 볼 수 있습니다. 여기에서 예산의 총액과 특정 예산에 할당된 전체 비용을 볼 수 있습니다.

예산을 보려면 다음 절차를 사용하십시오.

1. 아직 로그인하지 않았다면 에 로그인하고 Deadline Cloud 콘솔을 열고 팜을 선택한 다음 작업 관리를 선택합니다. AWS Management Console
2. 왼쪽 탐색 창에서 예산을 선택합니다. 예산 관리자 페이지가 나타납니다.
3. 활성 예산을 보려면 활성 예산 탭을 선택하고 보려는 예산명을 선택합니다. 예산 세부 정보 페이지가 나타납니다.
4. 만료된 예산의 예산 세부 정보를 보려면 비활성 예산 탭을 선택합니다. 그런 다음 보려는 예산의 이름을 선택합니다. 예산 세부 정보 페이지가 나타납니다.

예산 편집

모든 활성 예산을 편집할 수 있습니다. 활성 예산을 편집하려면 다음 절차를 사용하십시오.

1. 아직 로그인하지 않았다면 에 로그인하여 Deadline Cloud 콘솔을 열고 팜을 선택한 다음 작업 관리를 선택합니다. AWS Management Console
2. 예산 관리자 페이지의 활성 예산 탭에서 편집하려는 예산 옆의 버튼을 선택합니다.
3. 오른쪽 상단의 조치 드롭다운 메뉴에서 예산 편집을 선택합니다.
4. 원하는 대로 변경한 다음 예산 업데이트를 선택합니다.

예산 비활성화하기

모든 활성 예산을 비활성화할 수 있습니다. 예산을 비활성화하면 예산의 상태가 활성에서 비활성으로 변경됩니다. 예산이 비활성화되면 더 이상 해당 예산 금액까지 자원을 추적하지 않습니다.

예산을 비활성화하려면 다음 절차를 사용하십시오.

1. 아직 로그인하지 않았다면 에 로그인하여 Deadline Cloud 콘솔을 열고 팜을 선택한 다음 작업 관리를 선택합니다. AWS Management Console
2. 예산 관리자 페이지의 활성 예산 탭에서 비활성화하려는 예산 옆의 버튼을 선택합니다.
3. 오른쪽 상단의 조치 드롭다운 메뉴에서 예산 비활성화를 선택합니다. 잠시 후 선택한 예산이 활성에서 비활성으로 변경되고 활성 예산 탭에서 비활성 예산 탭으로 이동합니다.

데드라인 클라우드 사용량 탐색기 사용

Deadline Cloud 사용량 탐색기를 사용하면 각 팜에서 발생하는 활동에 대한 실시간 지표를 확인할 수 있습니다. 대기열, 작업, 라이선스 제품 또는 인스턴스 유형과 같은 다양한 변수별로 팜의 비용을 확인할 수 있습니다. 다양한 기간을 선택하여 특정 기간의 사용량을 확인하고 시간 경과에 따른 사용 추세를 살펴보세요. 또한 선택한 데이터 포인트의 세부 분석을 볼 수 있으므로 지표를 더 자세히 살펴볼 수 있습니다. 사용량은 시간 (분 및 시간) 또는 비용 (USD) 별로 표시할 수 있습니다.

다음 섹션에서는 Deadline Cloud 사용량 탐색기에 액세스하고 사용하는 단계를 보여줍니다.

주제

- 전제 조건
- 사용량 탐색기를 엽니다.

- 사용 현황 탐색기를 사용하세요.

전제 조건

Deadline Cloud 사용 탐색기를 사용하려면 둘 중 하나 MANAGER 또는 OWNER 팜 권한이 있어야 합니다. 자세한 정보는 팜, 큐, 플릿의 사용자 및 그룹을 관리합니다.을 참조하세요.

사용량 탐색기를 엽니다.

Deadline Cloud 사용 탐색기를 열려면 다음 절차를 사용하십시오.

1. 데드라인 클라우드 콘솔에 AWS Management Console 로그인하고 엽니다.
2. 사용 가능한 팜을 모두 보려면 팜 보기를 선택합니다.
3. 정보를 얻으려는 팜을 찾은 다음 작업 관리를 선택합니다. 데드라인 클라우드 모니터가 새 탭에서 열립니다.
4. 데드라인 클라우드 모니터의 왼쪽 메뉴에서 사용량 탐색기를 선택합니다.

사용 현황 탐색기를 사용하세요.

사용 탐색기 페이지에서 데이터를 표시할 수 있는 특정 매개변수를 선택할 수 있습니다. 기본적으로 지난 7일 동안의 총 사용량을 시간 (시간 및 분) 으로 볼 수 있습니다. 이러한 매개변수를 변경할 수 있으며 표시되는 정보는 매개변수 설정에 따라 동적으로 변경됩니다.

대기열, 작업, 컴퓨팅 사용량, 인스턴스 유형 또는 라이선스 제품을 기준으로 결과를 그룹화할 수 있습니다. 라이선스 제품을 선택하면 특정 라이선스에 대한 비용이 계산됩니다. 다른 모든 그룹의 경우 각 작업을 실행하는 데 걸린 시간을 합산하여 시간을 계산합니다.

사용량 탐색기는 사용자가 설정한 필터 기준에 따라 100개의 결과만 반환합니다. 결과는 만든 날짜의 타임스탬프를 기준으로 내림차순으로 나열됩니다. 결과가 100개를 넘으면 오류 메시지가 표시됩니다. 쿼리를 구체화하여 결과 수를 줄일 수 있습니다.

- 더 작은 시간 범위를 선택하세요.
- 더 적은 수의 대기열을 선택하세요
- 다른 그룹화 선택 (예: 작업 대신 큐별로 그룹화)

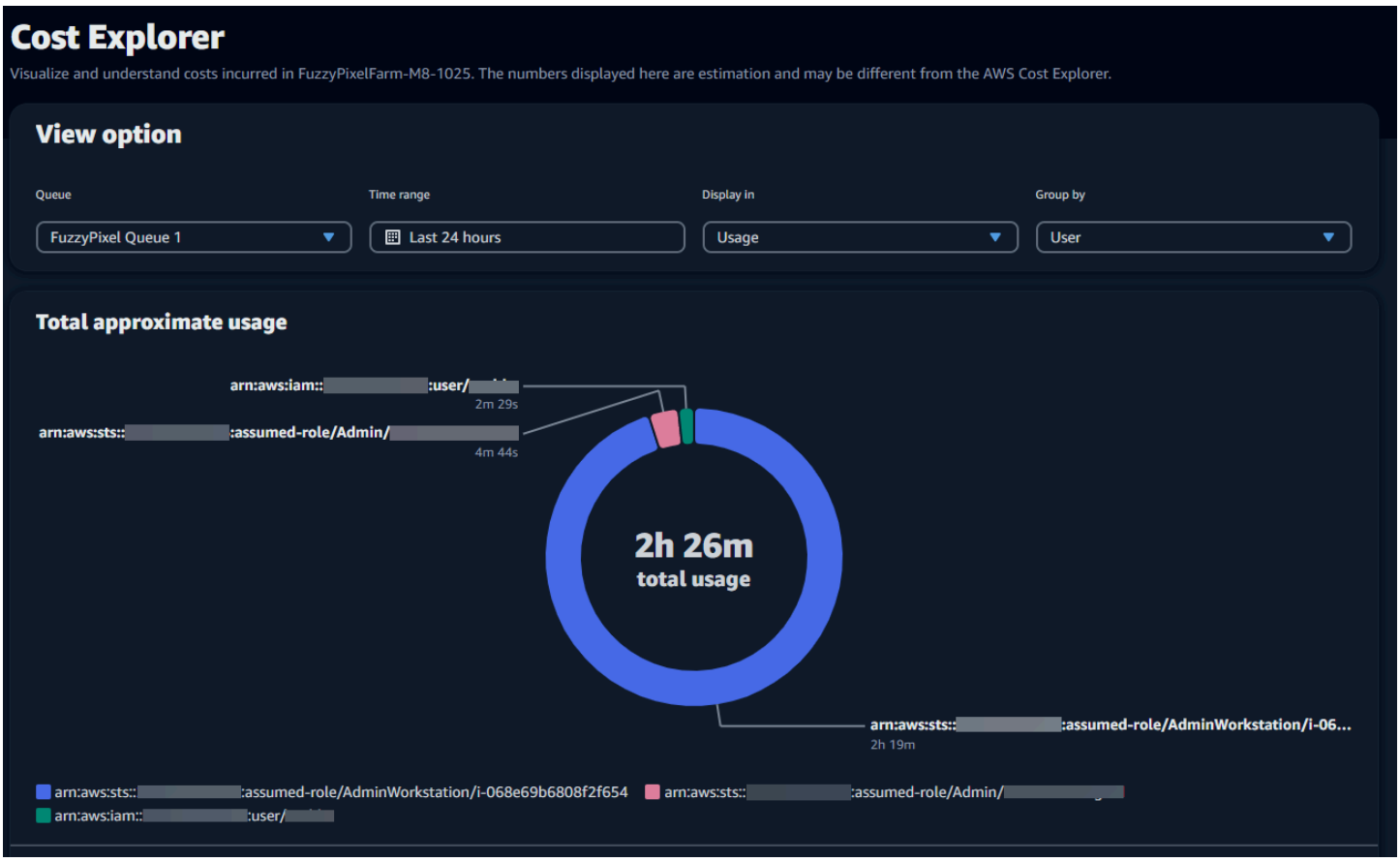
주제

- 시각적 그래프를 사용하여 데이터를 검토하세요.
- 지표 분석 보기
- 대략적인 대기열 실행 시간 보기

시각적 그래프를 사용하여 데이터를 검토하세요.

데이터를 시각적 형식으로 검토하여 추세 및 추가 분석이나 주의가 필요할 수 있는 잠재적 영역을 식별할 수 있습니다. 사용량 탐색기는 총계를 더 작은 소계로 그룹화할 수 있는 옵션과 함께 전체 사용량과 비용을 표시하는 원형 차트를 제공합니다.

> **Note**
>
> 차트에는 상위 5개 결과만 표시되며 다른 결과는 '기타' 섹션에 합쳐집니다. 차트 아래의 분류 섹션에서 모든 결과를 볼 수 있습니다.



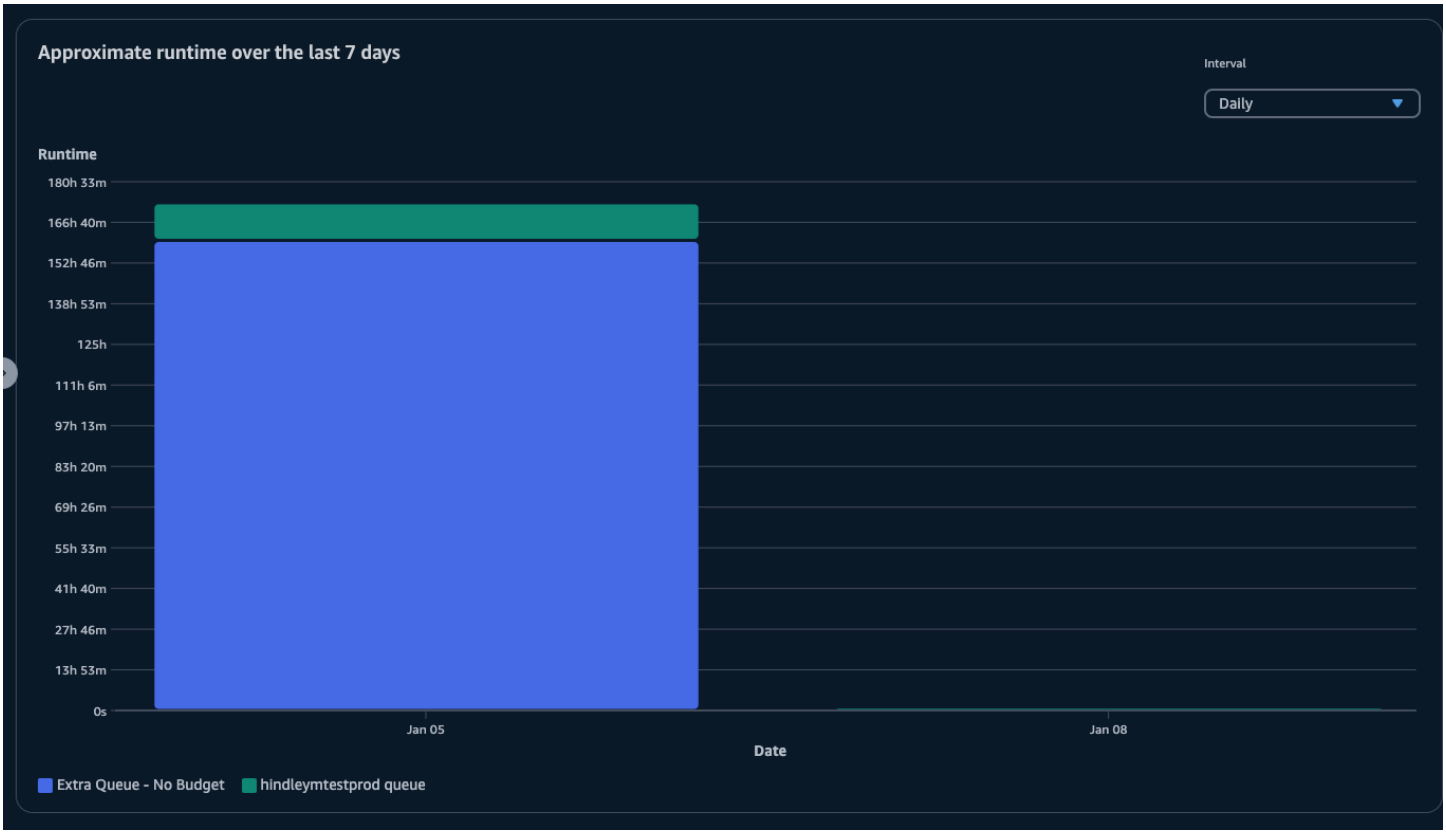
지표 분석 보기

원형 차트 아래에는 사용 현황 탐색기가 특정 지표에 대한 보다 자세한 분석을 제공하며, 이러한 지표는 매개변수 변경에 따라 변경됩니다. 기본적으로 사용량 탐색기에는 5개의 결과가 표시됩니다. 분류 섹션의 페이지 매김 화살표를 사용하여 결과를 스크롤할 수 있습니다.

분류는 기본적으로 최소화됩니다. 결과를 확장하여 표시하려면 전체 분류 보기 화살표를 선택합니다. 분류를 다운로드하려면 데이터 다운로드를 선택합니다.

대략적인 대기열 실행 시간 보기

또한 지정한 다양한 간격을 기준으로 대기열의 대략적인 실행 시간을 볼 수 있습니다. 간격 옵션은 시간별, 일별, 주별, 월별입니다. 간격을 선택하면 그래프에 대기열의 대략적인 실행 시간이 표시됩니다.



비용 관리

AWS Deadline Cloud는 작업 비용을 관리하고 시각화하는 데 도움이 되는 예산 및 사용량 탐색기를 제공합니다. 하지만 데드라인 클라우드는 Amazon S3와 같은 다른 AWS 서비스를 사용합니다. 이러한 서비스에 대한 비용은 Deadline Cloud 예산 또는 사용량 탐색기에 반영되지 않으며 사용량에 따라 별

도로 청구됩니다. Deadline Cloud를 구성하는 방법에 따라 다음 AWS 서비스 및 기타 서비스를 사용할 수 있습니다.

Service	가격 페이지
아마존 CloudWatch 로그	아마존 CloudWatch 로그 요금
Amazon Elastic Compute Cloud	Amazon 엘라스틱 컴퓨트 클라우드 요금
AWS Key Management Service	AWS Key Management Service 요금
AWS PrivateLink	AWS PrivateLink 요금
Amazon Simple Storage Service(S3)	Amazon Simple Storage Service 요금
Amazon Virtual Private Cloud	Amazon Virtual Private 클라우드 요금

비용 관리 모범 사례

다음 모범 사례를 사용하면 Deadline Cloud를 사용할 때의 비용과 비용과 효율성 사이에서 취할 수 있는 절충점을 이해하고 관리하는 데 도움이 될 수 있습니다.

> **Note**
>
> Deadline Cloud를 사용하는 데 드는 최종 비용은 여러 AWS 서비스 간의 상호 작용, 처리하는 작업량, 작업 실행 AWS 리전 위치에 따라 달라집니다. 다음 모범 사례는 지침이므로 비용을 크게 절감하지 못할 수 있습니다.

CloudWatch 로그 모범 사례

데드라인 클라우드는 작업자 및 작업 로그를 로그로 CloudWatch 전송합니다. 이러한 로그를 수집, 저장 및 분석하는 데 비용이 부과됩니다. 작업을 모니터링하는 데 필요한 최소한의 데이터만 기록하여 비용을 절감할 수 있습니다.

대기열 또는 플릿을 생성하면 Deadline Cloud는 다음 이름을 가진 CloudWatch 로그 로그 그룹을 생성합니다.

- `aws/deadline/<FARM_ID>/<FLEET_ID>`

- aws/deadline/*<FARM_ID>*/*<QUEUE_ID>*

기본적으로 이러한 로그는 만료되지 않습니다. 로그 그룹의 보존 정책을 조정하여 오래된 로그를 제거하고 스토리지 비용을 줄일 수 있습니다. 또한 로그를 Amazon S3로 내보낼 수 있습니다. Amazon S3 스토리지 비용은 스토리지 비용보다 저렴합니다 CloudWatch. 자세한 내용은 Amazon S3에 로그 데이터 내보내기를 참조하세요.

Amazon EC2 모범 사례

Amazon EC2 인스턴스를 서비스 관리형 플릿과 고객 관리형 플릿 모두에 사용할 수 있습니다. 세 가지 고려 사항이 있습니다.

- 서비스 관리형 플릿의 경우 플릿의 최소 작업자 수를 설정하여 항상 하나 이상의 인스턴스를 사용할 수 있도록 선택할 수 있습니다. 최소 작업자 수를 0보다 높게 설정하면 플릿에서 항상 이 만큼의 작업자를 운영하고 있습니다. 이렇게 하면 Deadline Cloud에서 작업 처리를 시작하는 데 걸리는 시간을 줄일 수 있지만 인스턴스의 유휴 시간에 대해서는 요금이 부과됩니다.
- 서비스 관리형 플릿의 경우 플릿의 최대 크기를 설정합니다. 이로 인해 플릿이 자동 확장할 수 있는 인스턴스 수가 제한됩니다. 처리 대기 중인 작업이 더 많아도 플릿은 이 크기를 넘지 않을 것입니다.
- 서비스 관리형 플릿과 고객 관리형 플릿 모두에 대해 플릿의 Amazon EC2 인스턴스 유형을 지정할 수 있습니다. 더 작은 인스턴스를 사용하면 분당 비용이 더 적게 들지만 작업을 완료하는 데 더 오래 걸릴 수 있습니다. 반대로 인스턴스가 클수록 분당 비용이 더 많이 들지만 작업 완료 시간을 줄일 수 있습니다. 인스턴스에 대한 작업 수요를 이해하면 비용을 줄이는 데 도움이 될 수 있습니다.
- 가능하면 플릿에 사용할 Amazon EC2 스팟 인스턴스를 선택하십시오. 스팟 인스턴스는 할인된 가격으로 사용할 수 있지만 온디맨드 요청으로 인해 중단될 수 있습니다. 온디맨드 인스턴스는 초 단위로 요금이 부과되며 중단되지 않습니다.

베스트 프랙티스 AWS KMS

기본적으로 Deadline Cloud는 AWS 소유 키로 데이터를 암호화합니다. 이 키에는 요금이 부과되지 않습니다.

고객 관리 키를 사용하여 데이터를 암호화하도록 선택할 수 있습니다. 자체 키를 사용하는 경우 키 사용 방식에 따라 요금이 부과됩니다. 기존 키를 사용하는 경우 추가 사용에 따른 추가 비용이 부과됩니다.

에 대한 모범 사례 AWS PrivateLink

를 사용하여 인터페이스 AWS PrivateLink 엔드포인트를 사용하여 VPC와 Deadline Cloud 간의 연결을 생성할 수 있습니다. 연결을 생성하면 데드라인 클라우드 API 작업을 모두 호출할 수 있습니다. 생성한 각 엔드포인트에 대해 시간당 요금이 부과됩니다. 를 사용하는 PrivateLink 경우 엔드포인트를 3개 이상 생성해야 하며, 구성에 따라 최대 5개까지 필요할 수 있습니다.

Amazon S3 모범 사례

Deadline Cloud는 Amazon S3를 사용하여 처리, 작업 첨부, 출력 및 로그에 필요한 자산을 저장합니다. Amazon S3와 관련된 비용을 줄이려면 저장하는 데이터의 양을 줄이십시오. 몇 가지 제안 사항:

- 현재 사용 중이거나 곧 사용할 자산만 저장하십시오.
- S3 수명 주기 구성을 사용하여 S3 버킷에서 사용하지 않는 파일을 자동으로 삭제합니다.

아마존 VPC 모범 사례

고객 관리형 플릿에 사용량 기반 라이선스를 사용하는 경우 Deadline Cloud 라이선스 엔드포인트를 생성합니다. Deadline Cloud 라이선스 엔드포인트는 계정에 생성된 Amazon VPC 엔드포인트입니다. 이 엔드포인트에는 시간당 요금이 부과됩니다. 비용을 줄이려면 사용량 기반 라이선스를 사용하지 않을 때는 엔드포인트를 제거하세요.

보안: Deadline Cloud

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처를 활용할 수 있습니다.

보안은 기업과 기업 간의 공동 책임입니다. AWS 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 — AWS AWS 서비스 클라우드에서 실행되는 인프라를 보호하는 역할을 합니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 범위AWS 서비스 내 규정 준수 프로그램의AWS 서비스 참조하십시오. AWS Deadline Cloud
- 클라우드에서의 보안 — AWS 서비스 사용하는 항목에 따라 책임이 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 공동 책임 모델을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 Deadline Cloud됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 Deadline Cloud 충족하도록 구성하는 방법을 보여줍니다. 또한 Deadline Cloud 리소스를 모니터링하고 보호하는 데 도움이 AWS 서비스 되는 기타 도구를 사용하는 방법도 알아봅니다.

주제

- 데이터 보호: Deadline Cloud
- 데드라인 클라우드의 ID 및 액세스 관리
- 규정 준수 검증: Deadline Cloud
- 의 레질리언스 Deadline Cloud
- 데드라인 클라우드의 인프라 보안
- 데드라인 클라우드의 구성 및 취약성 분석
- 교차 서비스 혼동된 대리인 방지
- 인터페이스 엔드포인트를 AWS Deadline Cloud 사용한 액세스 (AWS PrivateLink)
- 데드라인 클라우드의 보안 모범 사례

데이터 보호: Deadline Cloud

AWS 공동 책임 모델 의 데이터 보호에 적용됩니다 AWS Deadline Cloud. 이 모델에 설명된 대로 AWS 는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하십시오.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center OR AWS Identity and Access Management (IAM) 을 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신할 수 있습니다. AWS TLS 1.2는 필수이며 TLS 1.3를 권장합니다.
- 를 사용하여 API 및 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 API를 AWS 통해 액세스할 때 FIPS 140-2로 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용하십시오. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조하십시오.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API Deadline Cloud 또는 AWS 서비스 SDK를 사용하거나 다른 방법으로 작업하는 경우가 포함됩니다. AWS CLI AWS 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

주제

- 저장된 데이터 암호화
- 전송 중 암호화
- 키 관리

- 인터네트워크 트래픽 개인 정보 보호
- 옵트아웃

저장된 데이터 암호화

AWS Deadline Cloud AWS Key Management Service ()AWS KMS 에 저장된 암호화 키를 사용하여 저장된 데이터를 암호화하여 민감한 데이터를 보호합니다. 저장 중 암호화는 가능한 모든 AWS 리전 곳에서 Deadline Cloud 사용할 수 있습니다.

데이터를 암호화하면 디스크에 저장된 민감한 데이터는 유효한 키가 없으면 사용자나 애플리케이션이 읽을 수 없습니다. 유효한 관리 키를 가진 당사자만 데이터를 해독할 수 있습니다.

저장된 데이터를 암호화하는 데 Deadline Cloud 사용하는 방법에 AWS KMS 대한 자세한 내용은 을 참조하십시오. 키 관리

전송 중 암호화

전송 중인 데이터의 경우 전송 계층 보안 (TLS) 1.2 또는 1.3을 AWS Deadline Cloud 사용하여 서비스와 작업자 간에 전송되는 데이터를 암호화합니다. TLS 1.2는 필수이며 TLS 1.3를 권장합니다. 또한 가상 사설 클라우드 (VPC) 를 사용하는 경우 VPC와 (과) 사이에 사설 연결을 설정하는 AWS PrivateLink 데 사용할 수 있습니다. Deadline Cloud

키 관리

새 팜을 만들 때 다음 키 중 하나를 선택하여 팜 데이터를 암호화할 수 있습니다.

- AWS 소유한 KMS 키 - 팜을 만들 때 키를 지정하지 않는 경우의 기본 암호화 유형입니다. KMS 키는 이 (가) 소유합니다. AWS Deadline Cloud AWS 소유한 키는 보거나 관리하거나 사용할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 별도의 조치를 취할 필요는 없습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의AWS 소유 키를 참조하십시오.
- 고객 관리형 KMS 키 - 팜을 생성할 때 고객 관리 키를 지정합니다. 팜 내의 모든 콘텐츠는 KMS 키로 암호화됩니다. 키는 계정에 저장되며 사용자가 생성, 소유, 관리하며 AWS KMS 요금이 부과됩니다. KMS 키를 완전히 제어할 수 있습니다. 다음과 같은 작업을 수행할 수 있습니다.
  - 주요 정책 수립 및 유지
  - IAM 정책 및 권한 부여 수립 및 유지
  - 키 정책 활성화 및 비활성화
  - 태그 추가

- 키 별칭 생성

Deadline Cloud 팜에서 사용하는 고객 소유 키는 수동으로 교체할 수 없습니다. 키 자동 교체가 지원됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 소유 키를 참조하십시오.

고객 관리 키를 만들려면 AWS Key Management Service 개발자 가이드의 대칭 고객 관리 키 만들기 단계를 따르세요.

지원금 Deadline Cloud 사용 AWS KMS 방법

Deadline Cloud 고객 관리 키를 사용하려면 허가가 필요합니다. 고객 관리 키로 암호화된 팜을 생성하면 지정한 KMS 키에 대한 액세스 권한을 얻으라는 `CreateGrant` 요청을 보내 사용자 대신 권한 부여를 Deadline Cloud 생성합니다. AWS KMS

Deadline Cloud 여러 권한 부여를 사용합니다. 각 권한 부여는 데이터를 암호화하거나 복호화하는 데 필요한 다른 부분에서 Deadline Cloud 사용됩니다. Deadline Cloud 또한 권한을 사용하여 사용자를 대신하여 데이터를 저장하는 데 사용되는 다른 AWS 서비스 (예: Amazon Simple Storage Service, Amazon Elastic Block Store 또는) 에 대한 액세스를 허용합니다 OpenSearch.

서비스 관리형 플릿에서 머신을 관리할 수 Deadline Cloud 있는 부여에는 서비스 주체 `GranteePrincipal` 대신 Deadline Cloud 계정 번호와 역할이 포함됩니다. 일반적이지는 않지만 팜에 지정된 고객 관리형 KMS 키를 사용하여 서비스 관리 플릿의 작업자를 위해 Amazon EBS 볼륨을 암호화하는 데 필요합니다.

고객 관리형 키 정책

키 정책은 고객 관리형 키에 대한 액세스를 제어합니다. 각 키에는 누가 키를 사용할 수 있고 어떻게 사용할 수 있는지를 결정하는 명령문을 포함하는 정확히 하나의 키 정책이 있어야 합니다. 고객 관리 키를 생성할 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키에 대한 액세스 관리를 참조하십시오.

에 대한 최소 IAM 정책 CreateFarm

콘솔 또는 `CreateFarm` API 작업을 사용하여 고객 관리 키를 사용하여 팜을 생성하려면 다음 AWS KMS API 작업을 허용해야 합니다.

- kms:CreateGrant - 고객 관리형 키에 권한 부여를 추가합니다. 지정된 AWS KMS 키에 대한 콘솔 액세스 권한을 부여합니다. 자세한 내용은 AWS Key Management Service 개발자 가이드의 권한 사용을 참조하십시오.
- kms:Decrypt— Deadline Cloud 팜의 데이터를 해독할 수 있습니다.
- kms:DescribeKey— 키를 검증할 수 Deadline Cloud 있도록 고객 관리 키 세부 정보를 제공합니다.
- kms:GenerateDataKey— 고유한 데이터 키를 사용하여 데이터를 암호화할 수 있습니다. Deadline Cloud

다음 정책 설명은 CreateFarm 작업에 필요한 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineCreateGrants",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws::kms:us-west-2:111122223333:key/1234567890abcdef0",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

읽기 전용 작업에 대한 최소 IAM 정책

팜, 큐, 플릿에 대한 정보를 가져오는 것과 같은 읽기 전용 Deadline Cloud 작업에 고객 관리 키를 사용하기 위함입니다. 다음과 같은 AWS KMS API 작업이 허용되어야 합니다.

- kms:Decrypt— Deadline Cloud 팜의 데이터를 해독할 수 있습니다.

- kms:DescribeKey— 키를 검증할 수 Deadline Cloud 있도록 고객 관리 키 세부 정보를 제공합니다.

다음 정책 설명은 읽기 전용 작업에 필요한 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineReadOnly",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

읽기-쓰기 작업에 대한 최소 IAM 정책

팜, 큐, 플릿 생성 및 업데이트와 같은 읽기-쓰기 Deadline Cloud 작업에 고객 관리 키를 사용하는 것. 다음 AWS KMS API 작업이 허용되어야 합니다.

- kms:Decrypt— Deadline Cloud 팜의 데이터를 해독할 수 있습니다.
- kms:DescribeKey— 키를 검증할 수 Deadline Cloud 있도록 고객 관리 키 세부 정보를 제공합니다.
- kms:GenerateDataKey— 고유한 데이터 키를 사용하여 데이터를 암호화할 수 있습니다. Deadline Cloud

다음 정책 설명은 CreateFarm 작업에 필요한 권한을 부여합니다.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineReadWrite",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey",
            ],
            "Resource": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

암호화 키 모니터링

Deadline Cloud 팜에서 AWS KMS 고객 관리 키를 사용하는 경우 Amazon CloudWatch Logs를 사용하여 AWS CloudTrail로 Deadline Cloud 보내는 요청을 추적할 수 AWS KMS있습니다.

CloudTrail 보조금 지급 이벤트

다음 예제 CloudTrail 이벤트는 권한 부여가 생성될 때, 일반적으로 `CreateFarmCreateMonitor`, 또는 `CreateFleet` 작업을 호출할 때 발생합니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/Admin/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
```

```
                "arn": "arn:aws::iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T02:05:26Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T02:05:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "operations": [
            "CreateGrant",
            "Decrypt",
            "DescribeKey",
            "Encrypt",
            "GenerateDataKey"
        ],
        "constraints": {
            "encryptionContextSubset": {
                "aws:deadline:farmId": "farm-abcdef1234567890098765432 1fedcba",
                "aws:deadline:accountId": "111122223333"
            }
        },
        "granteePrincipal": "deadline.amazonaws.com",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
        "retiringPrincipal": "deadline.amazonaws.com"
    },
    "responseElements": {
        "grantId": "6bbe819394822a400fe5e3a75d0e9ef16c1733143fff0c1fc00dc7ac282a18a0",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
```

```
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE44444"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CloudTrail 복호화 이벤트

다음 예제 CloudTrail 이벤트는 고객 관리형 KMS 키를 사용하여 값을 해독할 때 발생합니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/SampleRole/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/SampleRole",
                "accountId": "111122223333",
                "userName": "SampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T18:46:51Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T18:51:44Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:deadline:farmId": "farm-abcdef12345678900987654321fedcba",
            "aws:deadline:accountId": "111122223333",
            "aws-crypto-public-key": "AotL+SAMPLEVALUEiOMEXAMPLEaaqNOTREALaGTESTONLY
+p/5H+EuKd4Q=="
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    },
    "responseElements": null,
    "requestID": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeffffff",
    "eventID": "ffffffff-eeee-dddd-cccc-bbbbbbaaaaaa",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


CloudTrail 암호화를 위한 이벤트입니다.

다음 예제 CloudTrail 이벤트는 고객 관리형 KMS 키를 사용하여 값을 암호화할 때 발생합니다.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/SampleRole/SampleUser01",
```

```
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/SampleRole",
                "accountId": "111122223333",
                "userName": "SampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T18:46:51Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T18:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "numberOfBytes": 32,
        "encryptionContext": {
            "aws:deadline:farmId": "farm-abcdef1234567890098765432 1fedcba",
            "aws:deadline:accountId": "111122223333",
            "aws-crypto-public-key": "AotL+SAMPLEVALUEiOMEXAMPLEaaqNOTREALaGTESTONLY
+p/5H+EuKd4Q=="
        },
        "keyId": "arn:aws::kms:us-
west-2:111122223333:key/abcdef12-3456-7890-0987-654321fedcba"
    },
    "responseElements": null,
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


고객 관리형 KMS 키 삭제

AWS Key Management Service (AWS KMS) 에서 고객 관리형 KMS 키를 삭제하는 것은 파괴적이며 잠재적으로 위험할 수 있습니다. 이렇게 하면 키와 연결된 키 구성 요소와 모든 메타데이터가 되돌릴 수 없는 방식으로 삭제됩니다. 고객 관리형 KMS 키가 삭제된 후에는 해당 키로 암호화된 데이터를 더 이상 복호화할 수 없습니다. 즉, 데이터를 복구할 수 없게 됩니다.

이것이 바로 고객이 AWS KMS KMS 키를 삭제하기 전에 최대 30일의 대기 시간을 주는 이유입니다. 기본 대기 기간은 30일입니다.

대기 기간에 대해

고객 관리형 KMS 키를 삭제하는 것은 파괴적이고 잠재적으로 위험할 수 있으므로 대기 기간을 7~30일로 설정해야 합니다. 기본 대기 기간은 30일입니다.

하지만 실제 대기 기간은 예약한 기간보다 최대 24시간까지 길 수 있습니다. 키가 삭제될 실제 날짜와 시간을 확인하려면 DescribeKey작업을 사용하십시오. AWS KMS 콘솔의 키 세부 정보 페이지에 있는 일반 구성 섹션에서 예약된 삭제 날짜를 확인할 수도 있습니다. 시간대를 확인하세요.

대기 기간 동안 고객 관리형 키의 상태 및 키 상태는 삭제 대기 중입니다.

- 삭제 대기 중인 고객 관리형 KMS 키는 어떠한 암호화 작업에도 사용할 수 없습니다.
- AWS KMS 삭제 보류 중인 고객 관리형 KMS 키의 백업 키는 교체하지 않습니다.

고객 관리형 KMS 키 삭제에 대한 자세한 내용은 개발자 안내서의 AWS Key Management Service 고객 마스터 키 삭제를 참조하십시오.

인터네트워크 트래픽 개인 정보 보호

AWS Deadline Cloud Amazon VPC (가상 사설 클라우드) 를 지원하여 연결을 보호합니다. Amazon VPC는 Virtual Private Cloud(VPC)의 보안을 강화하고 모니터링하기 위해 사용할 수 있는 여러 가지 기능을 제공합니다.

VPC 내에서 실행되는 Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스를 사용하여 고객 관리형 플릿 (CMF) 을 설정할 수 있습니다. AWS PrivateLink사용할 Amazon VPC 엔드포인트를 배포하면 CMF의 작업자와 엔드포인트 간의 Deadline Cloud 트래픽이 VPC 내에 유지됩니다. 또한 인스턴스에 대한 인터넷 액세스를 제한하도록 VPC를 구성할 수 있습니다.

서비스 관리형 플릿에서 작업자는 인터넷을 통해 연락할 수 없지만 인터넷에 액세스할 수 있고 인터넷을 통해 서비스에 연결할 수 있습니다. Deadline Cloud

옵트아웃

AWS Deadline Cloud 개발 및 개선에 도움이 되는 특정 운영 정보를 수집합니다 Deadline Cloud. 수집된 데이터에는 AWS 계정 ID 및 사용자 ID와 같은 정보가 포함되므로 문제가 있는 경우 사용자를 정확하게 식별할 수 Deadline Cloud있습니다. 또한 리소스 ID (해당하는 경우 FarmID 또는 QueueID), 제품 이름 (예:, 등), 제품 버전과 같은 Deadline Cloud 특정 정보도 수집합니다. JobAttachments WorkerAgent

애플리케이션 구성을 사용하여 이 데이터 수집을 거부할 수 있습니다. 클라이언트 워크스테이션과 차량 작업자 모두와 상호 작용하는 Deadline Cloud각 컴퓨터는 개별적으로 옵트아웃해야 합니다.

Deadline Cloud 모니터 - 데스크톱

Deadline Cloud 모니터 - 데스크톱은 충돌 발생 시점, 애플리케이션 실행 시점 등의 운영 정보를 수집하여 애플리케이션에 문제가 발생한 시점을 파악할 수 있도록 합니다. 운영 정보 수집을 거부하려면 설정 페이지로 이동하여 Deadline Cloud Monitor의 성능 측정을 위한 데이터 수집 켜기를 선택 해제하십시오.

옵트아웃하면 데스크톱 모니터는 더 이상 운영 데이터를 전송하지 않습니다. 이전에 수집된 모든 데이터는 보관되며 서비스 개선을 위해 계속 사용될 수 있습니다. 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요.

AWS Deadline Cloud CLI 및 툴

AWS Deadline Cloud CLI, 제출자 및 작업자 에이전트는 모두 충돌 발생 시점 및 작업 제출 시기와 같은 운영 정보를 수집하여 이러한 애플리케이션에 문제가 발생할 때 이를 파악할 수 있도록 합니다. 이 운영 정보 수집을 거부하려면 다음 방법 중 하나를 사용하십시오.

- 터미널에 를 입력합니다**`deadline config set telemetry.opt_out true`**.

  이렇게 하면 현재 사용자로 실행할 때 CLI, 제출자 및 작업자 에이전트가 옵트아웃됩니다.
- Deadline Cloud 작업자 에이전트를 설치할 때 **`--telemetry-opt-out`** 명령줄 인수를 추가하십시오. 예를 들어 **`./install.sh --farm-id $FARM_ID --fleet-id $FLEET_ID --telemetry-opt-out`**입니다.
- 작업자 에이전트, CLI 또는 제출자를 실행하기 전에 환경 변수를 설정합니다. **`DEADLINE_CLOUD_TELEMETRY_OPT_OUT=true`**

옵트아웃한 후에는 Deadline Cloud 도구가 더 이상 운영 데이터를 전송하지 않습니다. 이전에 수집된 모든 데이터는 보관되며 서비스를 개선하는 데 계속 사용될 수 있습니다. 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요.

데드라인 클라우드의 ID 및 액세스 관리

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 AWS 도와줍니다. IAM 관리자는 Deadline Cloud 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 관리합니다. IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

- 고객
- ID를 통한 인증
- 정책을 사용한 액세스 관리
- 데드라인 클라우드가 IAM과 함께 작동하는 방식
- 데드라인 클라우드의 ID 기반 정책 예제
- AWS 데드라인 클라우드의 관리형 정책
- AWS 데드라인 클라우드 ID 및 액세스 문제 해결

고객

데드라인 클라우드에서 수행하는 작업에 따라 AWS Identity and Access Management (IAM) 사용 방식이 다릅니다.

서비스 사용자 — Deadline Cloud 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. Deadline Cloud 기능을 더 많이 사용하여 업무를 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. 데드라인 클라우드의 기능에 액세스할 수 없는 경우 을 참조하십시오AWS 데드라인 클라우드 ID 및 액세스 문제 해결.

서비스 관리자 — 회사에서 데드라인 클라우드 리소스를 담당하고 있다면 데드라인 클라우드에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 사용자가 액세스해야 하는 Deadline Cloud 기능 및 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사에서 Deadline Cloud와 함께 IAM을 사용하는 방법에 대한 자세한 내용은 을 참조하십시오데드라인 클라우드가 IAM과 함께 작동하는 방식.

IAM 관리자 — IAM 관리자인 경우 Deadline Cloud에 대한 액세스를 관리하기 위한 정책을 작성하는 방법에 대해 자세히 알아보는 것이 좋습니다. IAM에서 사용할 수 있는 Deadline Cloud ID 기반 정책의 예를 보려면 을 참조하십시오. 데드라인 클라우드의 ID 기반 정책 예제

ID를 통한 인증

인증은 ID 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK) 와 명령줄 인터페이스 (CLI) 를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하십시오.

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하십시오.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS Directory Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. AWS 계정 IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하십시오.

IAM 사용자 및 그룹

IAM 사용자는 단일 사용자 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증이 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하십시오.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지

고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하십시오.

IAM 역할

IAM 역할은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 역할 사용을 참조하십시오.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 서드 파티 ID 공급자의 역할 생성 단원을 참조하십시오. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하십시오.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하십시오.
- 서비스 간 액세스 — 일부는 다른 기능을 사용합니다. AWS 서비스 AWS 서비스예를 들어 서비스에서 직접적 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.
  - 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.
- 서비스 연결 역할 — 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하십시오.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하십시오.

정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요를 참조하십시오.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 가이드의 ACL(액세스 제어 목록) 개요를 참조하십시오.

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 보안 인증 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하십시오.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU) 에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책 (SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함) 에 대한 권한을 제한합니다. AWS 계정 루트 사용자조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하십시오.
- 세션 정책 - 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하십시오.

여러 정책 타입

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련된 경우 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

데드라인 클라우드가 IAM과 함께 작동하는 방식

IAM을 사용하여 데드라인 클라우드에 대한 액세스를 관리하기 전에 데드라인 클라우드에서 사용할 수 있는 IAM 기능에 대해 알아보세요.

데드라인 클라우드와 함께 사용할 수 있는 IAM 기능 AWS

IAM 특성	데드라인 클라우드 지원
ID 기반 정책	예
리소스 기반 정책	아니요

IAM 특성	데드라인 클라우드 지원
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACLs	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
전달 액세스 세션(FAS)	예
서비스 역할	예
서비스 연결 역할	아니요

Deadline Cloud 및 기타 제품이 대부분의 IAM 기능과 어떻게 AWS 서비스 작동하는지 자세히 알아보려면 IAM 사용 설명서의 IAM과 함께 작동하는AWS 서비스를 참조하십시오.

데드라인 클라우드의 ID 기반 정책

보안 인증 기반 정책 지원	예

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하십시오.

데드라인 클라우드의 ID 기반 정책 예제

데드라인 클라우드 ID 기반 정책의 예를 보려면 을 참조하십시오. 데드라인 클라우드의 ID 기반 정책 예제

데드라인 클라우드 내의 리소스 기반 정책

리소스 기반 정책 지원	아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 AWS 계정경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 개체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하십시오.

데드라인 클라우드의 정책 조치

정책 작업 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

데드라인 클라우드 작업 목록을 보려면 서비스 권한 부여 참조의 AWS Deadline Cloud에서 정의한 작업을 참조하십시오.

데드라인 클라우드의 정책 조치는 조치 앞에 다음 접두사를 사용합니다.

```
deadline
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

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

데드라인 클라우드 ID 기반 정책의 예를 보려면 을 참조하십시오. 데드라인 클라우드의 ID 기반 정책 예제

데드라인 클라우드의 정책 리소스

정책 리소스 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 문장에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

데드라인 클라우드 리소스 유형 및 해당 ARN 목록을 보려면 서비스 인증 참조의 AWS Deadline Cloud에서 정의한 리소스를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업에 대해 알아보려면 AWS Deadline Cloud에서 정의한 작업을 참조하십시오.

데드라인 클라우드 ID 기반 정책의 예를 보려면 을 참조하십시오. 데드라인 클라우드의 ID 기반 정책 예제

데드라인 클라우드의 정책 조건 키

서비스별 정책 조건 키 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 `OR` 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예컨대, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하십시오.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 AWS 설명서의 글로벌 조건 컨텍스트 키를 참조하십시오.

데드라인 클라우드 조건 키 목록을 보려면 서비스 인증 참조의 AWS 데드라인 클라우드의 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업 및 리소스에 대해 알아보려면 AWS Deadline Cloud에서 정의한 작업을 참조하십시오.

데드라인 클라우드 ID 기반 정책의 예를 보려면 을 참조하십시오. 데드라인 클라우드의 ID 기반 정책 예제

데드라인 클라우드의 ACL

ACL 지원	아니요

ACL(액세스 통제 목록)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

데드라인 클라우드가 포함된 ABAC

ABAC 지원(정책의 태그)	예

ABAC(속성 기반 액세스 제어)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 개체 (사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 정보는 IAM 사용 설명서의 ABAC란 무엇입니까?를 참조하십시오. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하십시오.

데드라인 클라우드에서 임시 자격 증명 사용

임시 보안 인증 지원	예

임시 자격 증명을 사용하여 로그인하면 일부 자격 증명이 AWS 서비스 작동하지 않습니다. 임시 자격 증명을 사용하는 방법을 AWS 서비스 비롯한 추가 정보는 IAM 사용 설명서의 IAM과AWS 서비스 연동되는 내용을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하면 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 SSO (Single Sign-On) 링크를 AWS 사용하여 액세스하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다

음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 역할로 전환(콘솔)을 참조하십시오.

또는 API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS CLI . AWS 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하십시오.

데드라인 클라우드의 포워드 액세스 세션

전달 액세스 세션(FAS) 지원	예

IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

데드라인 클라우드의 서비스 역할

서비스 역할 지원	예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.

> **Warning**
>
> 서비스 역할의 권한을 변경하면 Deadline Cloud 기능이 작동하지 않을 수 있습니다. Deadline Cloud에서 관련 지침을 제공하는 경우에만 서비스 역할을 수정하세요.

데드라인 클라우드의 서비스 연결 역할

서비스 연결 역할 지원	아니요

서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스를 참조하십시오. 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

데드라인 클라우드의 ID 기반 정책 예제

기본적으로 사용자와 역할에는 Deadline Cloud 리소스를 만들거나 수정할 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

각 리소스 유형의 ARN 형식을 포함하여 Deadline Cloud에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 인증 참조의 AWS Deadline Cloud의 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- 데드라인 클라우드 콘솔 사용
- 대기열에 작업을 제출하기 위한 정책
- 라이선스 엔드포인트 생성을 허용하는 정책
- 특정 팜 대기열을 모니터링할 수 있도록 허용하는 정책

정책 모범 사례

ID 기반 정책에 따라 계정에서 Deadline Cloud 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하고 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서

사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 직무에 대한AWS 관리형 정책을 참조하십시오.

- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM의 정책 및 권한을 참조하십시오.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하십시오.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하십시오.
- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA 보호 API 액세스 구성을 참조하십시오.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하십시오.

데드라인 클라우드 콘솔 사용

AWS 데드라인 클라우드 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 내 Deadline Cloud 리소스에 대한 세부 정보를 나열하고 볼 수 있어야 AWS 계정합니다. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 Deadline Cloud 콘솔을 계속 사용할 수 있도록 하려면 Deadline Cloud *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 엔티티에 연결하십시오. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하십시오.

대기열에 작업을 제출하기 위한 정책

이 예시에서는 특정 팜의 특정 대기열에 작업을 제출할 권한을 부여하는 범위가 축소된 정책을 만듭니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SubmitJobsFarmAndQueue",
            "Effect": "Allow",
            "Action": "deadline:CreateJob",
            "Resource": "arn:aws:deadline:REGION:ACCOUNT_ID:farm/FARM_A/queue/QUEUE_B/
job/*"
        }
    ]
}
```

라이선스 엔드포인트 생성을 허용하는 정책

이 예시에서는 라이선스 엔드포인트를 생성하고 관리하는 데 필요한 권한을 부여하는 범위가 축소된 정책을 생성합니다. 이 정책을 사용하여 팜과 연결된 VPC에 대한 라이선스 엔드포인트를 만들 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "SID": "CreateLicenseEndpoint",
        "Effect": "Allow",
        "Action": [
            "deadline:CreateLicenseEndpoint",
            "deadline:DeleteLicenseEndpoint",
            "deadline:GetLicenseEndpoint",
            "deadline:UpdateLicenseEndpoint",
            "deadline:ListLicenseEndpoints",
            "deadline:PutMeteredProduct",
            "deadline:DeleteMeteredProduct",
            "deadline:ListMeteredProducts",
            "deadline:ListAvailableMeteredProducts",
            "ec2:CreateVpcEndpoint",
            "ec2:DescribeVpcEndpoints",
            "ec2:DeleteVpcEndpoints"
```

```
        ],
        "Resource": "*"
    }]
}
```

특정 팜 대기열을 모니터링할 수 있도록 허용하는 정책

이 예시에서는 특정 팜의 특정 대기열에 있는 작업을 모니터링할 권한을 부여하는 범위가 축소된 정책을 만듭니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "MonitorJobsFarmAndQueue",
        "Effect": "Allow",
        "Action": [
            "deadline:SearchJobs",
            "deadline:ListJobs",
            "deadline:GetJob",
            "deadline:SearchSteps",
            "deadline:ListSteps",
            "deadline:ListStepConsumers",
            "deadline:ListStepDependencies",
            "deadline:GetStep",
            "deadline:SearchTasks",
            "deadline:ListTasks",
            "deadline:GetTask",
            "deadline:ListSessions",
            "deadline:GetSession",
            "deadline:ListSessionActions",
            "deadline:GetSessionAction"
        ],
        "Resource": [
            "arn:aws:deadline:REGION:123456789012:farm/FARM_A/queue/QUEUE_B",
            "arn:aws:deadline:REGION:123456789012:farm/FARM_A/queue/QUEUE_B/*"
        ]
    }]
}
```

AWS 데드라인 클라우드의 관리형 정책

AWS 관리형 정책은 에서 생성하고 관리하는 독립형 정책입니다. AWS AWS 관리형 정책은 많은 일반 사용 사례에 대한 권한을 제공하도록 설계되었으므로 사용자, 그룹 및 역할에 권한을 할당하기 시작할 수 있습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한 권한을 부여하지 않을 수도 있다는 점에 유의하세요. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

관리형 정책에 정의된 권한은 변경할 수 없습니다. AWS AWS 관리형 정책에 정의된 권한을 업데이트하는 경우 AWS 해당 업데이트는 정책이 연결된 모든 주체 ID (사용자, 그룹, 역할) 에 영향을 미칩니다. AWS 새 API 작업이 시작되거나 기존 서비스에 새 AWS 서비스 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 가장 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하십시오.

AWS 관리형 정책: AWSDeadlineCloud-FleetWorker

`AWSDeadlineCloud-FleetWorker`정책을 AWS Identity and Access Management (IAM) ID에 연결할 수 있습니다.

이 정책은 이 플릿의 작업자에게 서비스에 연결하고 서비스로부터 작업을 받는 데 필요한 권한을 부여합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `deadline`— 교장이 여러 대의 근로자를 관리할 수 있도록 허용합니다.

정책 세부 정보의 JSON 목록은 AWS 관리형 정책 참조 안내서의 AWSDeadlineCloud FleetWorker -를 참조하십시오.

AWS 관리형 정책: AWSDeadlineCloud-WorkerHost

`AWSDeadlineCloud-WorkerHost` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책은 서비스에 처음 연결하는 데 필요한 권한을 부여합니다. Amazon Elastic Compute Cloud (Amazon EC2) 인스턴스 프로필로 사용할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `deadline`— 보안 담당자가 작업자를 생성할 수 있습니다.

정책 세부 정보의 JSON 목록은 AWS 관리형 정책 참조 안내서의 AWSDeadlineCloud WorkerHost -를 참조하십시오.

AWS 관리형 정책: AWSDeadlineCloud-UserAccessFarms

`AWSDeadlineCloud-UserAccessFarms` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책을 통해 사용자는 자신이 속한 팜과 멤버십 수준에 따라 팜 데이터에 액세스할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `deadline`— 사용자가 팜 데이터에 액세스할 수 있도록 허용합니다.
- `ec2`— 사용자가 Amazon EC2 인스턴스 유형에 대한 세부 정보를 볼 수 있습니다.
- `identitystore`— 사용자가 사용자 및 그룹 이름을 볼 수 있습니다.

정책 세부 정보의 JSON 목록은 AWS 관리형 정책 참조 안내서의 UserAccess 팜을 참조하십시오 AWSDeadlineCloud.

AWS 관리형 정책: AWSDeadlineCloud-UserAccessFleets

`AWSDeadlineCloud-UserAccessFleets` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책을 통해 사용자는 자신이 속한 농장과 회원 등급에 따라 플릿 데이터에 액세스할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `deadline`— 사용자가 팜 데이터에 액세스할 수 있도록 허용합니다.
- `ec2`— 사용자가 Amazon EC2 인스턴스 유형에 대한 세부 정보를 볼 수 있습니다.

- `identitystore`— 사용자가 사용자 및 그룹 이름을 볼 수 있습니다.

정책 세부 정보의 JSON 목록은 AWS 관리형 정책 참조 안내서의 AWSDeadlineCloud- UserAccess 플릿을 참조하십시오.

AWS 관리형 정책: AWSDeadlineCloud-UserAccessJobs

`AWSDeadlineCloud-UserAccessJobs` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책을 통해 사용자는 자신이 속한 팜과 구성원 수준에 따라 작업 데이터에 액세스할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `deadline`— 사용자가 팜 데이터에 액세스할 수 있습니다.
- `ec2`— 사용자가 Amazon EC2 인스턴스 유형에 대한 세부 정보를 볼 수 있습니다.
- `identitystore`— 사용자가 사용자 및 그룹 이름을 볼 수 있습니다.

정책 세부 정보의 JSON 목록은 AWS 관리형 정책 참조 안내서의 AWSDeadlineCloud - UserAccess 작업을 참조하십시오.

AWS 관리형 정책: AWSDeadlineCloud-UserAccessQueues

`AWSDeadlineCloud-UserAccessQueues` 정책을 IAM 보안 인증에 연결할 수 있습니다.

이 정책을 통해 사용자는 자신이 속한 팜과 구성원 수준에 따라 큐 데이터에 액세스할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `deadline`— 사용자가 팜 데이터에 액세스할 수 있도록 허용합니다.
- `ec2`— 사용자가 Amazon EC2 인스턴스 유형에 대한 세부 정보를 볼 수 있습니다.
- `identitystore`— 사용자가 사용자 및 그룹 이름을 볼 수 있습니다.

정책 세부 정보의 JSON 목록은 AWS 관리형 정책 참조 안내서의 AWSDeadlineCloud-UserAccessQueues 참조하십시오.

관리형 정책에 대한 마감일 클라우드 업데이트 AWS

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후 Deadline Cloud의 AWS 관리형 정책 업데이트에 대한 세부 정보를 확인하세요. 이 페이지의 변경 사항에 대한 자동 알림을 보려면 Deadline Cloud 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
데드라인 클라우드는 변경 사항을 추적하기 시작했습니다.	데드라인 클라우드는 AWS 관리형 정책의 변경 사항을 추적하기 시작했습니다.	2024년 4월 2일

AWS 데드라인 클라우드 ID 및 액세스 문제 해결

다음 정보를 사용하면 Deadline Cloud와 IAM을 사용할 때 발생할 수 있는 일반적인 문제를 진단하고 해결하는 데 도움이 됩니다.

주제

- 데드라인 클라우드에서 작업을 수행할 권한이 없습니다.
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 제 외부 사용자가 Deadline Cloud AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

데드라인 클라우드에서 작업을 수행할 권한이 없습니다.

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 `mateojackson`IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`*리소스에 대한 세부 정보를 보려고 하지만 가상 `deadline:`*`GetWidget`*권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 deadline:GetWidget on resource: my-example-widget
```

이 경우 `deadline:`*`GetWidget`*작업을 사용하여 *`my-example-widget`*리소스에 액세스할 수 있도록 `mateojackson`사용자 정책을 업데이트해야 합니다.

도움이 필요하면 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

작업을 수행할 권한이 없다는 오류가 발생하는 경우 Deadline Cloud에 역할을 넘길 수 있도록 정책을 업데이트해야 합니다. `iam:PassRole`

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 이라는 IAM 사용자가 Deadline Cloud에서 콘솔을 사용하여 작업을 `marymajor` 수행하려고 할 때 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

제 외부 사용자가 Deadline Cloud AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하십시오.

- Deadline Cloud가 이러한 기능을 지원하는지 알아보려면 을 참조하십시오데드라인 클라우드가 IAM과 함께 작동하는 방식.
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 IAM 사용 설명서에서 소유하고 AWS 계정 있는 다른 IAM 사용자에게 액세스 권한 제공을 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정AWS 계정 소유에 대한 액세스 제공을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 페더레이션)을 참조하십시오.

- 교차 계정 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하십시오.

규정 준수 검증: Deadline Cloud

특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면AWS 서비스 규정 준수 프로그램의AWS 서비스 범위별, 규정 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 AWS 규정 준수 프로그램AWS 보증 프로그램 규정AWS 참조하십시오.

를 사용하여 AWS Artifact타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 퀵 스타트 가이드 - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 — 이 백서에서는 기업이 HIPAA 적격 애플리케이션을 만드는 AWS 데 사용할 수 있는 방법을 설명합니다.

> **Note**
>
> 모든 AWS 서비스 사람이 HIPAA 자격을 갖춘 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조를 참조하십시오.

- AWS 규정 준수 리소스AWS — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (미국 표준 기술 연구소 (NIST), 결제 카드 산업 보안 표준 위원회 (PCI), 국제 표준화기구 (ISO) 등) 에서 보안 제어에 대한 지침을 매핑합니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub— 이를 AWS 서비스 통해 내부 AWS보안 상태를 포괄적으로 파악할 수 있습니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에

대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하십시오.

- Amazon GuardDuty — 환경에 의심스럽고 악의적인 활동이 있는지 AWS 계정모니터링하여 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 수 있습니다.
- AWS Audit Manager— 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

의 레질리언스 Deadline Cloud

AWS 글로벌 인프라는 가용 영역을 중심으로 구축됩니다 AWS 리전 . AWS 리전 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다. 이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

가용 영역에 대한 AWS 리전 자세한 내용은 글로벌 인프라를 참조하십시오AWS .

AWS Deadline Cloud 작업 첨부 파일 S3 버킷에 저장된 데이터는 백업하지 않습니다. 표준 Amazon S3 백업 메커니즘 (예: S3 버전 관리 또는 AWS Backup) 을 사용하여 작업 첨부 데이터 백업을 활성화할 수 있습니다.

데드라인 클라우드의 인프라 보안

AWS 데드라인 클라우드는 관리형 서비스로서 AWS 글로벌 네트워크 보안의 보호를 받습니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 Deadline Cloud에 액세스할 수 있습니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

데드라인 클라우드는 AWS PrivateLink 가상 사설 클라우드 (VPC) 엔드포인트 정책 사용을 지원하지 않습니다. 엔드포인트에 대한 전체 액세스 권한을 부여하는 AWS PrivateLink 기본 정책을 사용합니다. 자세한 내용은 AWS PrivateLink 사용 설명서의 기본 엔드포인트 정책을 참조하십시오.

데드라인 클라우드의 구성 및 취약성 분석

AWS 게스트 운영 체제 (OS) 및 데이터베이스 패치, 방화벽 구성, 재해 복구와 같은 기본 보안 작업을 처리합니다. 적합한 제3자가 이 절차를 검토하고 인증하였습니다. 자세한 내용은 다음 리소스를 참조하세요.

- 공동 책임 모델
- Amazon Web Services: 보안 프로세스의 개요(백서)

AWS Deadline Cloud는 서비스 관리 플릿 또는 고객 관리 플릿의 작업을 관리합니다.

- 서비스 관리 플릿의 경우 Deadline Cloud는 게스트 운영 체제를 관리합니다.
- 고객 관리 플릿의 경우 운영 체제를 관리할 책임은 사용자에게 있습니다.

Deadline Cloud의 구성 및 취약성 분석에 대한 AWS 추가 정보는 다음을 참조하십시오.

- 데드라인 클라우드의 보안 모범 사례

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS서비스 간 사칭으로 인해 대리인 문제가 혼동될 수 있습니다. 교차 서비스 가장은 한 서비스(호출하는 서비스)가 다른 서비스(호출되는 서비스)를 호출할 때 발생할 수 있습니다. 직접적으로 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS 에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

리소스 정책에 aws:SourceArn및 aws:SourceAccount글로벌 조건 컨텍스트 키를 사용하여 리소스에 다른 서비스에 AWS Deadline Cloud 부여하는 권한을 제한하는 것이 좋습니다. 하나의 리소스만 교차 서비스 액세스와 연결되도록 허용하려는 경우 aws:SourceArn를 사용하십시오. 해당 계정의 모든 리소스가 교차 서비스 사용과 연결되도록 허용하려는 경우 aws:SourceAccount을 사용하세요.

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 Amazon 리소스 이름(ARN)이 포함된 aws:SourceArn 전역 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모르거나 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드 문자(*)를 포함한 aws:SourceArn 글로벌 조건 컨텍스트 키를 사용합니다. 예를 들어 arn:aws:deadline:*:*123456789012*:*입니다.

만약 aws:SourceArn 값에 Amazon S3 버킷 ARN과 같은 계정 ID가 포함되어 있지 않은 경우, 권한을 제한하려면 두 글로벌 조건 컨텍스트 키를 모두 사용해야 합니다.

다음 예제는 aws:SourceArn 및 aws:SourceAccount 글로벌 조건 컨텍스트 키를 사용하여 혼동되는 부정 문제를 방지하는 Deadline Cloud 방법을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "deadline.amazonaws.com"
    },
    "Action": "deadline:ActionName",
    "Resource": [
      "*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:deadline:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

인터페이스 엔드포인트를 AWS Deadline Cloud 사용한 액세스 (AWS PrivateLink)

를 AWS PrivateLink 사용하여 VPC와 (과) 사이에 프라이빗 연결을 생성할 수 있습니다. AWS Deadline Cloud인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 연결을 사용하지 않고도 VPC에 있는 Deadline Cloud 것처럼 액세스할 수 있습니다. AWS Direct Connect VPC의 인스턴스에서 Deadline Cloud API에 액세스하는 데는 퍼블릭 IP 주소가 필요하지 않습니다.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 Deadline Cloud로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은AWS PrivateLink 가이드의 AWS PrivateLink를 통해 AWS 서비스 에 액세스를 참조하세요.

고려 사항 Deadline Cloud

인터페이스 엔드포인트를 설정하기 전에 Deadline Cloud가이드의 인터페이스 VPC 엔드포인트를 사용하여 AWS 서비스에 액세스를 참조하십시오.AWS PrivateLink

Deadline Cloud 인터페이스 엔드포인트를 통해 모든 API 작업에 대한 호출을 지원합니다.

기본적으로 인터페이스 엔드포인트를 통해 전체 Deadline Cloud 액세스가 허용됩니다. 또는 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 인터페이스 엔드포인트를 Deadline Cloud 통해 들어오는 트래픽을 제어할 수 있습니다.

Deadline Cloud VPC 엔드포인트 정책을 지원하지 않습니다. 자세한 정보는 AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 VPC 엔드포인트에 대한 액세스 제어를 참조하세요.

Deadline Cloud 엔드포인트

Deadline Cloud 를 사용하여 서비스에 액세스하는 데 두 개의 엔드포인트를 사용합니다. AWS PrivateLink

작업자는 `com.amazonaws.region.deadline.scheduling` 엔드포인트를 사용하여 대기열에서 작업을 가져오고, 진행 상황을 보고하고 Deadline Cloud, 작업 결과를 다시 보냅니다. 고객 관리형 플릿을 사용하는 경우, 관리 작업을 사용하지 않는 한 스케줄링 엔드포인트는 생성해야 하는 유일한 엔드포

인트입니다. 예를 들어 작업에서 더 많은 작업이 생성되는 경우 관리 엔드포인트가 작업을 호출하도록 설정해야 합니다. `CreateJob`

Deadline Cloud 모니터는 `com.amazonaws.region.deadline.management` 를 사용하여 대기열 및 집합을 만들고 수정하거나 작업, 단계 및 작업 목록을 가져오는 등 팜의 리소스를 관리합니다.

Deadline Cloud 또한 다음 서비스 엔드포인트에 대한 엔드포인트가 필요합니다. AWS

- Deadline Cloud 작업자가 AWS STS 작업 자산에 액세스할 수 있도록 인증하는 데 사용합니다. 에 대한 AWS STS자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM의 임시 보안 자격 증명을 참조하십시오.
- 인터넷에 연결되지 않은 서브넷에서 고객 관리형 플릿을 설정하는 경우 작업자가 로그를 작성할 수 있도록 CloudWatch Amazon Logs용 VPC 엔드포인트를 생성해야 합니다. 자세한 내용은 모니터링을 참조하십시오. CloudWatch
- 작업 첨부 파일을 사용하는 경우 작업자가 첨부 파일에 액세스할 수 있도록 Amazon Simple Storage Service (Amazon S3) 용 VPC 엔드포인트를 생성해야 합니다. 자세한 내용은 의 Job 첨부 파일을 참조하십시오 Deadline Cloud.

엔드포인트 생성: Deadline Cloud

Amazon VPC 콘솔 또는 () Deadline Cloud 를 사용하기 위한 인터페이스 엔드포인트를 생성할 수 있습니다 AWS Command Line Interface .AWS CLI자세한 내용은 AWS PrivateLink 설명서의 인터페이스 엔드포인트 생성을 참조하십시오.

다음 서비스 이름을 Deadline Cloud 사용하기 위한 관리 및 일정 엔드포인트를 생성하십시오. `### ### #` AWS 리전 곳으로 바꾸십시오. Deadline Cloud

```
com.amazonaws.region.deadline.management
```

```
com.amazonaws.region.deadline.scheduling
```

인터페이스 엔드포인트에 프라이빗 DNS를 활성화하면 기본 지역 DNS 이름을 Deadline Cloud 사용하여 API 요청을 할 수 있습니다. 작업자 작업 또는 `management.deadline.us-east-1.amazonaws.com` 기타 모든 작업을 예로 들 수 있습니다. `worker.deadline.us-east-1.amazonaws.com`

또한 다음 서비스 이름을 AWS STS 사용하기 위한 엔드포인트를 생성해야 합니다.

```
com.amazonaws.region.sts
```

고객 관리형 플릿이 인터넷 연결이 없는 서브넷에 있는 경우 다음 서비스 이름을 사용하여 CloudWatch 로그 엔드포인트를 생성해야 합니다.

```
com.amazonaws.region.logs
```

작업 첨부를 사용하여 파일을 전송하는 경우 다음 서비스 이름을 사용하여 Amazon S3 엔드포인트를 생성해야 합니다.

```
com.amazonaws.region.s3
```

데드라인 클라우드의 보안 모범 사례

AWS 데드라인 클라우드 (Deadline Cloud) 는 자체 보안 정책을 개발하고 구현할 때 고려해야 할 여러 보안 기능을 제공합니다. 다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 환경에 적절하지 않거나 충분하지 않을 수 있으므로 참고용으로만 사용해 주십시오.

> **Note**
>
> 여러 보안 주제의 중요성에 대한 자세한 내용은 공동 책임 모델을 참조하십시오.

데이터 보호

데이터 보호를 위해 AWS Identity and Access Management (IAM) 을 AWS 계정 사용하여 자격 증명을 보호하고 개별 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신하세요. AWS TLS 1.2는 필수이며 TLS 1.3를 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon Simple Storage Service(S3)에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.

- 명령행 인터페이스 또는 API를 통해 AWS 에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2 섹션을 참조하세요.

명칭 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마세요. 여기에는 AWS Deadline Cloud를 사용하거나 콘솔 AWS CLI, API 또는 AWS 서비스 AWS SDK를 사용하여 다른 방법으로 작업하는 경우가 포함됩니다. Deadline Cloud 또는 기타 서비스에 입력하는 모든 데이터는 진단 로그에 포함되도록 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함하지 마십시오.

AWS Identity and Access Management 권한

사용자, AWS Identity and Access Management (IAM) 역할을 사용하고 사용자에게 최소 권한을 부여하여 AWS 리소스에 대한 액세스를 관리합니다. 액세스 자격 증명을 생성, 배포, 교체 및 취소하기 위한 자격 증명 관리 정책 및 절차를 수립하세요. AWS 자세한 설명은 IAM 사용자 가이드의 IAM 모범 사례 섹션을 참조하십시오.

사용자 및 그룹으로 작업 실행

Deadline Cloud에서 대기열 기능을 사용할 때는 OS (운영 체제) 사용자와 기본 그룹을 지정하여 OS 사용자가 대기열 작업에 대한 최소 권한을 갖도록 하는 것이 좋습니다.

'사용자 권한으로 실행' (및 그룹) 을 지정하면 대기열에 제출된 작업의 모든 프로세스는 해당 OS 사용자를 사용하여 실행되며 해당 사용자의 관련 OS 권한을 상속합니다.

플릿과 큐 구성이 결합되어 보안 태세를 확립합니다. 대기열 측에서는 대기열 작업에 OS 및 AWS 권한을 사용하도록 “Job run as user” 및 IAM 역할을 지정할 수 있습니다. 플릿은 특정 대기열에 연결된 경우 대기열 내에서 작업을 실행하는 인프라 (작업자 호스트, 네트워크, 마운트된 공유 스토리지) 를 정의합니다. 작업자 호스트에서 사용 가능한 데이터는 하나 이상의 관련 대기열에 있는 작업자가 액세스할 수 있어야 합니다. 사용자 또는 그룹을 지정하면 다른 대기열, 설치된 다른 소프트웨어 또는 작업자 호스트에 액세스할 수 있는 다른 사용자로부터 작업의 데이터를 보호하는 데 도움이 됩니다. 대기열에 사용자가 없는 경우 모든 대기열 사용자로 가장할 수 있는 에이전트 사용자로 실행됩니다 (sudo). 이렇게 하면 사용자가 없는 대기열도 권한을 다른 대기열로 승격할 수 있습니다.

네트워킹

트래픽이 가로채거나 리디렉션되는 것을 방지하려면 네트워크 트래픽이 라우팅되는 방법과 위치를 보호하는 것이 중요합니다.

다음과 같은 방법으로 네트워킹 환경을 보호하는 것이 좋습니다.

- Amazon VPC (가상 사설 클라우드) 서브넷 라우팅 테이블을 보호하여 IP 계층 트래픽이 라우팅되는 방식을 제어합니다.
- 팜 또는 워크스테이션 설정에서 Amazon Route 53 (Route 53) 을 DNS 공급자로 사용하는 경우 Route 53 API에 대한 보안 액세스를 확보하십시오.
- 온프레미스 워크스테이션이나 다른 데이터 센터를 사용하는 AWS 등 외부에서 Deadline Cloud에 연결하는 경우 모든 온프레미스 네트워킹 인프라를 보호하십시오. 여기에는 라우터, 스위치 및 기타 네트워킹 장치의 DNS 서버 및 라우팅 테이블이 포함됩니다.

작업 및 작업 데이터

데드라인 클라우드 작업은 작업자 호스트의 세션 내에서 실행됩니다. 각 세션은 작업자 호스트에서 하나 이상의 프로세스를 실행합니다. 일반적으로 결과를 생성하려면 데이터를 입력해야 합니다.

이 데이터를 보호하기 위해 운영 체제 사용자를 대기열로 구성할 수 있습니다. 작업자 에이전트는 queue OS 사용자를 사용하여 세션 하위 프로세스를 실행합니다. 이러한 하위 프로세스는 큐 OS 사용자의 권한을 상속합니다.

모범 사례에 따라 이러한 하위 프로세스가 액세스하는 데이터에 대한 액세스를 보호하는 것이 좋습니다. 자세한 내용은 공동 책임 모델을 참조하십시오.

팜 구조

데드라인 클라우드 플릿과 대기열을 다양한 방법으로 정렬할 수 있습니다. 하지만 특정 방식을 사용할 경우 보안에 영향을 미칠 수 있습니다.

팜은 Deadline Cloud 리소스를 플릿, 큐, 스토리지 프로필 등 다른 팜과 공유할 수 없기 때문에 가장 안전한 경계 중 하나입니다. 하지만 팜 내에서 외부 AWS 리소스를 공유할 수 있기 때문에 보안 경계가 손상될 수 있습니다.

적절한 구성을 사용하여 동일한 팜 내의 대기열 간에 보안 경계를 설정할 수도 있습니다.

다음 모범 사례에 따라 동일한 팜에 보안 대기열을 만드십시오.

- 플릿을 동일한 보안 경계 내의 대기열에만 연결하십시오. 유의할 사항:
  - 작업자 호스트에서 작업이 실행된 후에도 임시 디렉터리나 큐 사용자의 홈 디렉터리 등에 데이터가 남아 있을 수 있습니다.

  - 작업을 제출하는 대기열에 관계없이 동일한 OS 사용자가 서비스 소유의 플릿 작업자 호스트에서 모든 작업을 실행합니다.
  - 작업은 작업자 호스트에서 실행 중인 프로세스를 그대로 둘 수 있으므로 다른 대기열의 작업에서 실행 중인 다른 프로세스를 관찰할 수 있습니다.
- 동일한 보안 경계 내의 대기열만 작업 첨부용 Amazon S3 버킷을 공유하도록 하십시오.
- 동일한 보안 경계 내의 대기열만 OS 사용자를 공유하도록 하십시오.
- 팜에 통합된 다른 모든 AWS 리소스를 경계까지 보호하십시오.

Job 첨부 대기열

작업 첨부는 Amazon S3 버킷을 사용하는 대기열과 연결됩니다.

- 작업 첨부 파일은 Amazon S3 버킷의 루트 접두사에 쓰고 읽습니다. `CreateQueue`API 호출에서 이 루트 접두사를 지정합니다.
- 버킷에는 대기열 사용자에게 버킷에 대한 액세스 권한을 부여하는 역할과 루트 접두사를 지정하는 해당 `Queue Role` 버킷이 있습니다. 대기열을 생성할 때 작업 첨부 파일 버킷 및 루트 접두사와 함께 `Queue Role` Amazon 리소스 이름 (ARN) 을 지정합니다.
- `AssumeQueueRoleForRead`, `AssumeQueueRoleForUser`, 및 `AssumeQueueRoleForWorker` API 작업에 대한 승인된 호출은 에 대한 임시 보안 자격 증명 세트를 반환합니다. `Queue Role`

대기열을 생성하고 Amazon S3 버킷과 루트 접두사를 재사용하면 정보가 승인되지 않은 당사자에게 공개될 위험이 있습니다. 예를 들어 QueuEA와 QueueB는 동일한 버킷과 루트 접두사를 공유합니다. 보안 워크플로우에서 Artista는 QueueA에 액세스할 수 있지만 QueueB에는 액세스할 수 없습니다. 하지만 여러 대기열이 버킷을 공유하는 경우 QueueA와 동일한 버킷 및 루트 접두사를 사용하기 때문에 Artista는 QueuEB 데이터의 데이터에 액세스할 수 있습니다.

콘솔은 기본적으로 안전한 대기열을 설정합니다. 대기열에 Amazon S3 버킷과 루트 접두사가 서로 다르게 조합되어 있어야 합니다. 단, 공통 보안 경계에 속하지 않는 한 대기열이 있어야 합니다.

대기열을 격리하려면 버킷과 루트 접두사에 대한 대기열 액세스만 허용하도록 `Queue Role` 를 구성해야 합니다. 다음 예시에서는 각 `#######` 리소스별 정보로 바꿉니다.

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

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::JOB_ATTACHMENTS_BUCKET_NAME",
        "arn:aws:s3:::JOB_ATTACHMENTS_BUCKET_NAME/JOB_ATTACHMENTS_ROOT_PREFIX/*"
      ],
      "Condition": {
        "StringEquals": { "aws:ResourceAccount": "ACCOUNT_ID" }
      }
    },
    {
      "Action": ["logs:GetLogEvents"],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:REGION:ACCOUNT_ID:log-group:/aws/deadline/FARM_ID/*"
    }
  ]
}
```

또한 역할에 신뢰 정책을 설정해야 합니다. 다음 예시에서는 *## ###* 텍스트를 리소스별 정보로 바꾸십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Effect": "Allow",
      "Principal": { "Service": "deadline.amazonaws.com" },
      "Condition": {
        "StringEquals": { "aws:SourceAccount": "ACCOUNT_ID" },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:deadline:REGION:ACCOUNT_ID:farm/FARM_ID"
        }
      }
    },
    {
      "Action": ["sts:AssumeRole"],
      "Effect": "Allow",
```

```
      "Principal": { "Service": "credentials.deadline.amazonaws.com" },
      "Condition": {
        "StringEquals": { "aws:SourceAccount": "ACCOUNT_ID" },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:deadline:REGION:ACCOUNT_ID:farm/FARM_ID"
        }
      }
    }
  ]
}
```

사용자 지정 소프트웨어 Amazon S3 버킷

Amazon S3 버킷의 사용자 지정 소프트웨어에 `Queue Role` 액세스하기 위해 다음 명령문을 추가할 수 있습니다. 다음 예제에서는 *SOFTWARE_BUCKET_NAME# S3 ### ####* 대체합니다.

```
"Statement": [
    {
        "Action": [
            "s3:GetObject",
            "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::SOFTWARE_BUCKET_NAME",
            "arn:aws:s3:::SOFTWARE_BUCKET_NAME/*"
        ]
    }
]
```

Amazon S3 보안 모범 사례에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3의 보안 모범 사례를 참조하십시오.

워커 호스트

작업자 호스트를 보호하여 각 사용자가 지정된 역할에 대한 작업만 수행할 수 있도록 합니다.

작업자 호스트를 보호하려면 다음 모범 사례를 따르는 것이 좋습니다.

- 대기열에 제출된 작업이 동일한 보안 경계 내에 있지 않는 한 여러 대기열에 동일한 `jobRunAsUser` 값을 사용하지 마십시오.

- 작업자 에이전트가 실행되는 OS 사용자 이름으로 대기열을 `jobRunAsUser` 설정하지 마십시오.
- 대기열 사용자에게 의도한 대기열 워크로드에 필요한 최소 권한의 OS 권한을 부여하십시오. 에이전트 프로그램 파일 또는 기타 공유 소프트웨어를 작업할 수 있는 파일 시스템 쓰기 권한이 없는지 확인하십시오.
- 루트 사용자와 `Administrator` 소유한 계정만 Linux 작업자 에이전트 프로그램 파일을 Windows 소유하고 수정할 수 있도록 하십시오.
- Linux작업자 호스트에서는 작업자 에이전트 사용자가 대기열 사용자로 프로세스를 시작할 수 `/etc/sudoers` 있도록 umask 재정의를 구성하는 것을 고려해 보십시오. 이 구성은 다른 사용자가 대기열에 기록된 파일에 액세스할 수 없도록 하는 데 도움이 됩니다.
- 신뢰할 수 있는 개인에게 작업자 호스트에 대한 최소 권한 액세스 권한을 부여하세요.
- 로컬 DNS 재정의 구성 파일 (`/etc/hosts`온/온) 과 워크스테이션 Linux 및 `C:\Windows\system32\etc\hosts` 작업자 Windows 호스트 운영 체제의 라우팅 테이블에 대한 권한을 제한합니다.
- 워크스테이션과 작업자 호스트 운영 체제의 DNS 구성에 대한 권한을 제한합니다.
- 운영 체제와 설치된 모든 소프트웨어를 정기적으로 패치하십시오. 이 접근 방식에는 제출자, 어댑터, 작업자 에이전트, OpenJD 패키지 등과 같이 Deadline Cloud와 함께 특별히 사용되는 소프트웨어가 포함됩니다.
- 대기열에는 강력한 비밀번호를 사용하세요. Windows `jobRunAsUser`
- 대기열의 비밀번호를 정기적으로 교체하세요`jobRunAsUser`.
- 비밀번호에 대한 액세스 권한이 최소한으로 유지되도록 하고 사용하지 않는 Windows 비밀번호는 삭제하세요.
- 대기열 `jobRunAsUser` 권한에 미래에 실행할 스케줄 명령을 부여하지 마세요.
  - 아니요Linux, 해당 계정의 `cron` 및 `at` 에 대한 액세스를 거부하세요.
  - 켜기Windows, Windows 작업 스케줄러에 대한 이러한 계정 액세스를 거부하십시오.

> **Note**
>
> 운영 체제 및 설치된 소프트웨어를 정기적으로 패치하는 것의 중요성에 대한 자세한 내용은 공동 책임 모델을 참조하십시오.

워크스테이션

Deadline Cloud에 액세스할 수 있는 워크스테이션을 보호하는 것이 중요합니다. 이 접근 방식은 Deadline Cloud에 제출하는 모든 작업이 사용자에게 청구되는 임의의 워크로드를 실행할 수 없도록 하는 데 도움이 됩니다. AWS 계정

아티스트 워크스테이션을 보호하려면 다음 모범 사례를 따르는 것이 좋습니다. 자세한 내용은 공동 책임 모델을 참조하세요.

- Deadline Cloud를 포함하여 액세스를 AWS제공하는 모든 영구 자격 증명을 보호하세요. 자세한 내용은 IAM 사용 설명서의 IAM 사용자의 액세스 키 관리를 참조하세요.
- 신뢰할 수 있고 안전한 소프트웨어만 설치하세요.
- 사용자가 ID 공급자와 페더레이션하여 임시 자격 AWS 증명으로 액세스하도록 요구하십시오.
- Deadline Cloud 제출자 프로그램 파일에 대한 보안 권한을 사용하여 변조를 방지하세요.
- 신뢰할 수 있는 개인에게 아티스트 워크스테이션에 대한 최소 권한 권한을 부여하세요.
- Deadline Cloud Monitor를 통해 확보한 제출자와 어댑터만 사용하세요.
- 워크스테이션과 작업자 호스트 운영 체제의 권한 `/etc/hosts` 및 라우팅 테이블을 제한하세요.
- 워크스테이션 및 작업자 호스트 운영 체제에 `/etc/resolv.conf` 대한 권한을 제한하십시오.
- 운영 체제와 설치된 모든 소프트웨어를 정기적으로 패치하십시오. 이 접근 방식에는 제출자, 어댑터, 작업자 에이전트, OpenJD 패키지 등과 같이 Deadline Cloud와 함께 특별히 사용되는 소프트웨어가 포함됩니다.

AWS 데드라인 클라우드 모니터링

모니터링은 AWS 데드라인 클라우드 (Deadline Cloud) 와 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 있어 중요한 부분입니다. AWS 솔루션의 모든 부분에서 모니터링 데이터를 수집하여 다중 지점 장애가 발생할 경우 이를 보다 쉽게 디버깅할 수 있습니다. Deadline Cloud를 모니터링하기 전에 다음 질문에 대한 답변이 포함된 모니터링 계획을 세워야 합니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

AWS Deadline Cloud는 리소스를 모니터링하고 잠재적 사고에 대응하는 데 사용할 수 있는 도구를 제공합니다. 이러한 도구 중 일부는 모니터링을 대신 수행하지만 일부 도구는 수동 개입이 필요합니다. 모니터링 작업을 최대한 자동화해야 합니다.

- Amazon은 실행 중인 AWS 리소스와 애플리케이션을 AWS 실시간으로 CloudWatch 모니터링합니다. 지표를 수집 및 추적하고, 맞춤 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 설명서를 참조하십시오.

  데드라인 클라우드에는 세 가지 CloudWatch 지표가 있습니다.
- 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 설명서를 참조하십시오.
- Amazon을 사용하면 AWS 서비스를 자동화하고 애플리케이션 가용성 문제 또는 리소스 변경과 같은 시스템 이벤트에 자동으로 대응할 EventBridge 수 있습니다. AWS 서비스에서 발생하는 이벤트는 거의 EventBridge 실시간으로 전송됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하십시오.

- AWS CloudTrail계정에서 또는 AWS 계정을 대신하여 이루어진 API 호출 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 어떤 사용자와 계정이 전화를 걸었는지 AWS, 어떤 소스 IP 주소에서 호출이 이루어졌는지, 언제 호출이 발생했는지 식별할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

주제

- 를 사용하여 통화를 기록합니다. CloudTrail
- 를 통한 모니터링 CloudWatch
- 이벤트에 따른 조치 EventBridge

를 사용하여 통화를 기록합니다. CloudTrail

AWS Deadline Cloud는 Deadline Cloud와 AWS CloudTrail통합되어 사용자, 역할 또는 Deadline AWS 서비스 Cloud에서 수행한 작업의 기록을 제공하는 서비스입니다. CloudTrail 데드라인 클라우드에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 Deadline Cloud 콘솔에서의 호출 및 Deadline Cloud API 작업에 대한 코드 호출이 포함됩니다.

트레일을 생성하면 Deadline Cloud의 CloudTrail 이벤트를 포함하여 Amazon S3 버킷에 이벤트를 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 Deadline Cloud에 대한 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

자세한 CloudTrail 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

데드라인 클라우드 정보는 CloudTrail

CloudTrail 계정을 만들 AWS 계정 때 활성화됩니다. Deadline Cloud에서 활동이 발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. 내 사이트에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 이벤트 기록으로 CloudTrail 이벤트 보기를 참조하십시오.

CloudTrail 또한 사용자가 Deadline Cloud 모니터에 로그인하고 AWS 자격 증명을 받을 때 이벤트를 기록합니다. 사용자가 로그인하면 `signin.amazonaws.com` 소스와 이름이 포함된 CloudTrail 이벤트가 발생합니다`UserAuthentication`. 로그인한 사용자에게 `sts.amazonaws.com` 원본과 이름의 AWS 자격 증명이 제공되는 두 번째 이벤트가 있습니다. `AssumeRole` 사용자 ID는 역할 세션 이름 내의 두 번째 이벤트에 기록됩니다.

Deadline Cloud의 이벤트를 AWS 계정포함하여 내 이벤트의 진행 중인 기록을 보려면 트레일을 생성하세요. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 AWS 서비스 취하도록 기타를 구성할 수 있습니다.

자세한 내용은 다음을 참조하십시오.

추적 생성 개요

CloudTrail 지원되는 서비스 및 통합

에 대한 Amazon SNS 알림 구성 CloudTrail

여러 지역에서 CloudTrail 로그 파일 수신

여러 계정에서 CloudTrail 로그 파일 받기

Deadline Cloud는 다음과 같은 작업을 CloudTrail 로그 파일에 이벤트로 기록할 수 있도록 지원합니다.

- associate-member-to-farm
- associate-member-to-fleet
- associate-member-to-job
- associate-member-to-queue
- assume-fleet-role-for-read
- assume-fleet-role-for-작업자
- assume-queue-role-for-읽기
- assume-queue-role-for-사용자
- assume-queue-role-for-작업자
- 예산 만들기
- 농장 만들기
- create-fleet
- create-license-endpoint
- 모니터 생성
- 대기열 생성
- create-queue-environment

- create-queue-fleet-association
- create-storage-profile
- 워커 생성
- 예산 삭제
- 팜 삭제
- delete-fleet
- delete-license-endpoint
- delete-metered-product
- 삭제 모니터
- 삭제 대기열
- delete-queue-environment
- delete-queue-fleet-association
- delete-storage-profile
- 삭제 작업자
- disassociate-member-from-farm
- disassociate-member-from-fleet
- disassociate-member-from-job
- disassociate-member-from-queue
- get-application-version
- 예산 책정
- 농장을 짓다
- get-feature-map
- 겟 플릿
- get-license-endpoint
- 겟 모니터
- 겟 큐
- get-queue-environment
- get-queue-fleet-association
- get-sessions-statistics-aggregation
- get-storage-profile

- get-storage-profile-for-큐
- list-available-metered-products
- 리스트 예산
- list-farm-members
- 리스트 팜
- list-fleet-members
- 리스트 플릿
- list-job-members
- list-license-endpoints
- list-metered-products
- 리스트 모니터
- list-queue-environments
- list-queue-fleet-associations
- list-queue-members
- 목록 대기열
- list-storage-profiles
- list-storage-profiles-for-큐
- list-tags-for-resource
- put-metered-product
- start-sessions-statistics-aggregation
- tag-resource
- untag-resource
- 업데이트 예산
- 업데이트 팜
- 업데이트 플릿
- 업데이트 모니터
- 업데이트 대기열
- update-queue-environment
- update-queue-fleet-association
- update-storage-profile

- 업데이트 작업자

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 신원 정보를 이용하면 다음을 쉽게 알아볼 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부
- 역할 또는 연동 사용자를 위한 임시 보안 인증으로 요청을 생성하였는지.
- 다른 서비스에서 요청했는지.

자세한 내용은 CloudTrail사용자 ID 요소를 참조하십시오.

데드라인 클라우드 로그 파일 항목 이해

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

이 JSON 예제는 **`CreateFarm`** API 호출로 생성된 로그를 보여줍니다.

```
{
    "eventVersion": "0",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
        "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE-PrincipalID",
                "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
                "accountId": "111122223333",
                "userName": "EXAMPLE-UserName"
            },
            "webIdFederationData": {},
            "attributes": {
```

```
                "mfaAuthenticated": "false",
                "creationDate": "2021-03-08T23:25:49Z"
            }
        }
    },
    "eventTime": "2021-03-08T23:25:49Z",
    "eventSource": "deadline.amazonaws.com",
    "eventName": "CreateFarm",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "EXAMPLE-userAgent",
    "requestParameters": {
        "displayName": "example-farm",
        "kmsKeyArn": "arn:aws:kms:us-west-2:111122223333:key/111122223333",
        "X-Amz-Client-Token": "12abc12a-1234-1abc-123a-1a11bc1111a",
        "description": "example-description",
        "tags": {
            "purpose_1": "e2e"
            "purpose_2": "tag_test"
        }
    },
    "responseElements": {
        "farmId": "EXAMPLE-farmID"
    },
    "requestID": "EXAMPLE-requestID",
    "eventID": "EXAMPLE-eventID",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
}
```


이 예제에서는 AWS 지역, IP 주소 및 이벤트를 식별하는 데 도움이 되는 기타 `requestParameters` "“" 및 `displayName` `kmsKeyArn` "“와 같은 기타" “를 보여 줍니다.

를 통한 모니터링 CloudWatch

Amazon CloudWatch (CloudWatch) 은 원시 데이터를 수집하여 읽을 수 있는 거의 실시간 지표로 처리합니다. https://console.aws.amazon.com/cloudwatch/ CloudWatch 콘솔을 열어 데드라인 클라우드 지표를 보고 필터링할 수 있습니다.

- Deadline Cloud 고객 관리형 플릿에서는 다음과 같은 두 가지 `UnhealthyWorkerCount` 지표와 함께 다음을 CloudWatch 전송합니다. `RecommendedFleetSize`
- 이러한 지표의 네임스페이스는 다음과 같습니다. `AWS/DeadlineCloud.`
- 측정기준을 사용하여 `farmID` 측정항목을 `fleetID` 필터링할 수 있습니다.
- 두 지표 모두 단위를 사용합니다`count`.

이러한 통계는 15개월 동안 보관되므로 기록 정보에 액세스하여 웹 애플리케이션 또는 서비스 성능을 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.

Deadline Cloud에는 작업 로그와 작업자 로그라는 두 종류의 로그가 있습니다. 작업 로그는 스크립트로 또는 DCC가 실행될 때 실행 로그를 실행하는 경우입니다. 작업 로그에는 에셋 로드, 타일 렌더링 또는 텍스처를 찾을 수 없는 등의 이벤트가 표시될 수 있습니다.

작업자 로그에는 작업자 에이전트 프로세스가 표시됩니다. 여기에는 작업자 에이전트가 시작되거나, 직접 등록하거나, 진행 상황을 보고하거나, 구성을 로드하거나, 작업을 완료하는 시기가 포함될 수 있습니다.

Deadline Cloud의 경우 작업자는 이러한 로그를 로그에 CloudWatch 업로드합니다. 기본적으로 로그는 만료되지 않습니다. 작업에서 대량의 데이터가 출력되는 경우 추가 비용이 발생할 수 있습니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

각 로그 그룹의 보존 정책을 조정할 수 있습니다. 보존 기간을 줄이면 오래된 로그가 제거되므로 스토리지 비용을 줄이는 데 도움이 될 수 있습니다. 로그를 보관하려면 로그를 제거하기 전에 Amazon 심플 스토리지 서비스에 로그를 보관하면 됩니다. 자세한 내용은 Amazon 사용 CloudWatch 설명서의 콘솔을 사용하여 Amazon S3로 로그 데이터 내보내기를 참조하십시오.

> **Note**
>
> CloudWatch 로그 읽기는 로 제한됩니다 AWS. 많은 아티스트를 온보딩할 계획이라면 AWS 고객 지원팀에 문의하여 `GetLogEvents` 할당량 증가를 CloudWatch 요청하는 것이 좋습니다. 또한 디버깅하지 않을 때는 로그 테일링 포털을 닫는 것이 좋습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 로그 할당량을 참조하십시오.

이벤트에 따른 조치 EventBridge

Deadline Cloud는 EventBridge Amazon에 이벤트를 전송하여 서비스 상태 변경 사항을 알립니다. EventBridge 및 이러한 이벤트를 사용하여 플릿에 변경 사항이 있을 경우 이를 알리는 등의 조치를 취하는 규칙을 작성할 수 있습니다. 자세한 내용은 Amazon이란 무엇입니까? 를 참조하십시오. EventBridge

차량 크기 권장 변경

이벤트 기반 Auto Scaling을 사용하도록 플릿을 구성하면 Deadline Cloud는 플릿을 관리하는 데 사용할 수 있는 이벤트를 전송합니다. 각 이벤트에는 플릿의 현재 크기 및 요청된 크기에 대한 정보가 포함되어 있습니다. EventBridge 이벤트를 사용하는 예제와 Lambda 함수를 사용하여 이벤트를 처리하는 예는 을 참조하십시오. 데드라인 클라우드 스케일 권장 기능을 사용하여 Amazon EC2 플릿을 자동 확장하십시오.

플릿 크기 권장 사항 변경 이벤트는 다음과 같은 경우 전송됩니다.

- 권장 플릿 크기가 `oldFleetSize` 변경되고 이와 다른 `newFleetSize` 경우
- 서비스에서 실제 플릿 크기가 권장 플릿 크기와 일치하지 않는 것을 감지한 경우 GetFleet운영 응답에서 실제 플릿 크기를 확인할 수 있습니다. `workerCount` 이는 활성 Amazon EC2 인스턴스가 데드라인 클라우드 워커로 등록되지 않을 때 발생할 수 있습니다.

이벤트의 형식은 다음과 같습니다.

```
{
    "version": "0",
    "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
    "detail-type": "Fleet Size Recommendation Change",
    "source": "aws.deadline",
    "account": "111122223333",
    "time": "2017-12-22T18:43:48Z",
    "region": "us-west-1",
    "resources": [],
    "detail": {
        "farmId": "farm-12345678900000000000000000000000",
        "fleetId": "fleet-12345678900000000000000000000000",
        "oldFleetSize": 1,
        "newFleetSize": 5,
    }
}
```

다음 필드는 이벤트 패턴을 정의합니다.

"source": "aws.deadline"

이 이벤트의 소스가 데드라인 클라우드임을 식별합니다.

"detail-type": "Fleet Size Recommendation Change"

이벤트 타입을 식별합니다.

"detail": { }

플릿 크기의 권장 변경 사항에 대한 정보를 제공합니다.

"farmId": "farm-12345678900000000000000000000000"

플릿이 포함된 팜의 식별자입니다.

"fleetId": "fleet-12345678900000000000000000000000"

크기 변경이 필요한 플릿의 식별자.

"oldFleetSize": 1

플릿의 현재 크기.

"newFleetSize": 5

권장되는 새 플릿 크기.

에 대한 할당량 Deadline Cloud

AWS Deadline Cloud 작업을 처리하는 데 사용할 수 있는 리소스 (예: 팜, 플릿, 큐) 를 제공합니다. 를 생성할 때 각 AWS 계정리소스에 대해 기본 할당량을 설정합니다. AWS 리전

Service Quotas는 할당량을 보고 관리할 수 있는 중앙 위치입니다. AWS 서비스사용하는 많은 리소스에 대해 할당량 증가를 요청할 수도 있습니다.

에 대한 Deadline Cloud할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스을(를) 선택한 다음 Deadline Cloud을(를) 선택합니다.

할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요. Service Quotas에서 할당량을 아직 사용할 수 없는 경우 서비스 할당량 증가 양식을 사용하세요.

를 사용하여 AWS 데드라인 클라우드 리소스 생성 AWS CloudFormation

AWS Deadline Cloud는 리소스를 모델링하고 설정하는 데 도움이 되는 서비스인 과 통합되어 있으므로 AWS 리소스와 인프라를 만들고 관리하는 데 소요되는 시간을 줄일 수 있습니다. AWS CloudFormation원하는 모든 리소스 (예: 팜, 큐, 플릿) 를 설명하는 템플릿을 만들고 해당 AWS 리소스를 자동으로 AWS CloudFormation 프로비저닝 및 구성합니다.

를 사용하면 템플릿을 재사용하여 AWS CloudFormation Deadline Cloud 리소스를 일관되고 반복적으로 설정할 수 있습니다. 리소스를 한 번 설명한 다음 여러 AWS 계정 지역과 지역에서 동일한 리소스를 반복해서 프로비저닝하세요.

데드라인 클라우드 및 AWS CloudFormation 템플릿

Deadline Cloud 및 관련 서비스를 위한 리소스를 프로비저닝하고 구성하려면 AWS CloudFormation 템플릿을 이해해야 합니다. 템플릿은 JSON 또는 YAML로 서식 지정된 텍스트 파일입니다. 이 템플릿은 AWS CloudFormation 스택에 프로비저닝하려는 리소스를 설명합니다. JSON이나 YAML에 익숙하지 않은 경우 AWS CloudFormation Designer를 사용하여 템플릿을 시작하는 데 도움을 받을 수 있습니다. AWS CloudFormation 자세한 내용은 AWS CloudFormation 사용 설명서에서 AWS CloudFormation Designer이란 무엇입니까?를 참조하세요.

Deadline Cloud는 팜, 큐, 플릿 생성을 지원합니다. AWS CloudFormation팜, 큐, 플릿을 위한 JSON 및 YAML 템플릿의 예를 비롯한 자세한 내용은 사용 설명서의 Deadline Cloud를 참조하십시오.AWS AWS CloudFormation

에 대해 자세히 알아보십시오. AWS CloudFormation

자세히 AWS CloudFormation알아보려면 다음 리소스를 참조하십시오.

- AWS CloudFormation
- AWS CloudFormation 사용 설명서
- AWS CloudFormation API Reference
- AWS CloudFormation 명령줄 인터페이스 사용 설명서

데드라인 클라우드 사용 설명서의 문서 기록

다음 표에는 AWS Deadline Cloud 사용 설명서의 각 릴리스에서 변경된 주요 내용이 설명되어 있습니다.

변경 사항	설명	날짜
최초 릴리스	데드라인 클라우드 사용 설명서의 초기 릴리스입니다.	2024년 4월 2일

AWS 용어집

최신 AWS 용어는 참조의 AWS 용어집을 참조하십시오.AWS 용어집

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.