



구현 안내서

# AWS 인스턴스 스케줄러



# AWS 인스턴스 스케줄러: 구현 안내서

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

# Table of Contents

솔루션 개요 .....	1
기능 및 이점 .....	2
사용 사례 .....	3
개념 및 정의 .....	3
비용 .....	3
비용 조정 요인 .....	4
예약 대상 계산 .....	4
비용 최적화 전략 .....	5
참조 요금 예제(월별) .....	5
배포에 대한 비용 추정 .....	8
할당량 .....	8
조정 제한 사항 .....	8
추가 고려 사항 .....	9
AWS 서비스 할당량 .....	9
지원되는 AWS 리전 .....	10
계정 ID 또는 AWS Organizations ID를 사용한 교차 계정 인스턴스 예약 .....	10
계정 ID를 사용한 교차 계정 예약 활성화 .....	10
AWS Organizations ID를 사용하여 교차 계정 예약 활성화 .....	11
AWS Systems Manager Parameter Store를 사용한 계정 ID 관리 .....	11
예약에 지원되는 서비스 .....	11
인스턴스 종료 동작 .....	12
Amazon EC2 .....	12
Amazon RDS, Amazon Neptune 및 Amazon DocumentDB .....	12
Amazon RDS 유지 관리 기간 .....	13
Amazon EC2 Auto Scaling 그룹 .....	13
아키텍처 .....	14
아키텍처 다이어그램 .....	14
AWS Well-Architected 설계 고려 사항 .....	16
운영 우수성 .....	16
보안 .....	17
신뢰성 .....	17
성능 효율성 .....	17
비용 최적화 .....	17
지속 가능성 .....	18

스케줄러 구성 테이블 .....	18
스케줄러 CLI .....	18
이 솔루션에 사용되는 AWS 서비스 .....	18
보안 .....	20
KMS .....	20
Amazon IAM .....	21
암호화된 EC2 EBS 볼륨 .....	21
EC2 라이선스 관리자 .....	22
시작하기 .....	24
배포 프로세스 개요 .....	24
AWS CloudFormation 템플릿 .....	25
1단계: 인스턴스 스케줄러 허브 스택 시작 .....	25
2단계(선택 사항): 보조 계정에서 원격 스택 시작 .....	31
솔루션 구성 .....	34
운영자 안내서 .....	35
일정 구성 .....	35
코드형 인프라 사용(권장) .....	36
Amazon DynamoDB 콘솔 및 AWS 인스턴스 스케줄러 CLI 사용 .....	36
예약을 위한 인스턴스 태그 지정 .....	37
태그 값 설정 .....	37
암호화된 EBS 볼륨이 포함된 EC2 인스턴스 .....	37
License Manager에서 관리되는 EC2 인스턴스 .....	38
일정 추론 .....	38
기간 .....	38
시간대 .....	38
새 인스턴스 중지 필드 .....	38
최대 절전 모드 필드 .....	38
적용 필드 .....	39
실행 중인 필드 유지 .....	39
Systems Manager 유지 관리 기간 필드(EC2 인스턴스에만 적용됨) .....	39
인스턴스 유형 .....	40
일정 정의 .....	40
기간 참조 .....	42
시작 및 중지 시간 .....	42
요일 .....	44
해당 월의 일 .....	44

개월 .....	44
기간 정의 .....	44
샘플 일정 .....	47
오전 9시~오후 5시 표준 근무 시간 .....	47
오후 5시 이후 인스턴스 중지 .....	49
주말 동안 인스턴스 중지 .....	51
솔루션 리소스 .....	54
스케줄러 CLI .....	55
사전 조건 .....	55
자격 증명 .....	55
스케줄러 CLI 설치 .....	56
명령 구조 .....	57
공통 인수 .....	57
가용 명령 .....	58
create-period .....	58
create-schedule .....	60
delete-period .....	63
delete-schedule .....	63
describe-periods .....	64
describe-schedules .....	65
describe-schedule-usage .....	67
update-period .....	68
update-schedule .....	68
help .....	69
전역 구성 설정 업데이트 .....	70
코드형 인프라(IaC)를 사용하여 일정 관리 .....	70
EC2 용량 부족 오류 처리 .....	72
구성 .....	73
작동 방식 .....	73
요구 사항 및 제한 사항 .....	73
예제 .....	73
EC2 Auto Scaling 그룹 예약 .....	74
ASG 예약 개요 .....	74
ASG 실행/중지의 정의 .....	74
ASG 시작/중지 동작 .....	74
솔루션 모니터링 .....	75

로그 및 알림 .....	75
정보 태그 .....	76
CloudWatch Logs Insights 쿼리 .....	79
운영 인사이트 대시보드 .....	79
EventBridge 이벤트 모니터링 .....	81
문제 해결 .....	85
알려진 문제 해결 .....	85
문제: 원격 계정에서 예약되지 않은 인스턴스(v1.4-v3.0) .....	85
해결 방법 .....	85
문제: 예약되지 않은 인스턴스(v3.1 이상) .....	86
해결 방법 .....	86
문제: 암호화된 EC2 인스턴스가 시작되지 않음 .....	86
해결 방법 .....	86
문제: 정보 태그 지정으로 인한 예상치 못한 API 비용 .....	86
해결 방법 .....	86
문제: RDS 스냅샷 생성이 활성화되어 있으면 RDS 인스턴스가 중지되지 않음 .....	87
해결 방법 .....	87
AWS Support에 문의 .....	87
사례 생성 .....	87
지원 방법 .....	87
추가 정보 .....	88
사례를 더 빠르게 해결할 수 있도록 지원 .....	88
지금 해결 또는 문의 .....	88
솔루션 업데이트 .....	89
특정 버전의 주요 변경 사항 .....	90
v1.5.0 .....	90
v3.0.0 .....	90
v3.1.0 .....	91
솔루션 제거 .....	93
AWS 관리 콘솔 사용 .....	93
AWS Command Line Interface 사용 .....	93
개발자 안내서 .....	94
소스 코드 .....	94
레퍼런스 .....	95
데이터 수집 .....	95
관련 리소스 .....	95

---

기여자 .....	96
개정 .....	98
Notices .....	99
.....	c

# AWS 인스턴스 시작 및 중지 자동화

AWS 인스턴스 스케줄러 솔루션은 [Amazon Elastic Compute Cloud\(Amazon EC2\)](#) 및 [Amazon Relational Database Service\(Amazon RDS\)](#) 인스턴스를 비롯한 다양한 AWS 서비스의 시작 및 중지를 자동화합니다.

이 솔루션은 사용하지 않는 리소스는 중지하고 용량이 필요할 때 리소스를 시작하여 운영 비용을 줄여 줍니다. 예를 들어, 회사는 AWS 인스턴스 스케줄러를 사용하여 매일 업무 시간 외에도 인스턴스를 자동으로 중지할 수 있습니다. 모든 인스턴스를 최대 사용률로 유지하면 이 솔루션은 정규 업무 시간 동안에만 필요한 인스턴스에 대해 최대 70%의 비용을 절감할 수 있습니다(주간 사용률이 168시간에서 50시간으로 단축됨).

AWS 인스턴스 스케줄러는 Amazon Web Services(AWS) 리소스 태그와 [AWS Lambda](#)를 활용하여 고객이 정의한 일정에 따라 여러 AWS 리전 및 계정에서 인스턴스를 자동으로 중지하고 다시 시작합니다. 또한 이 솔루션을 사용하면 중지된 EC2 인스턴스에 최대 절전 모드를 사용할 수 있습니다.

이 구현 가이드에서는 AWS 인스턴스 스케줄러 솔루션의 개요, 참조 아키텍처 및 구성 요소, 배포 계획 시 고려 사항, AWS 클라우드에 솔루션을 배포하기 위한 구성 단계를 제공합니다.

이 가이드는 자신의 환경에서 AWS 인스턴스 스케줄러를 구현하려는 IT 인프라 아키텍트, 관리자 및 DevOps 전문가를 대상으로 합니다.

이 탐색 테이블을 사용하여 다음 질문에 대한 답을 빠르게 찾을 수 있습니다.

다음을 수행하려는 경우 ...	읽기 ...
이 솔루션을 실행하는 데 드는 비용을 파악합니다. 미국 동부(버지니아 북부) 리전에서 이 솔루션을 실행하는 데 드는 예상 비용은 월 13.15 USD입니다.	<a href="#">비용</a>
이 솔루션의 보안 고려 사항을 이해합니다.	<a href="#">AWS Well-Architected 보안, 보안</a>
일정을 구성합니다.	<a href="#">스케줄러 구성 테이블</a>
이 솔루션에서 지원되는 AWS 리전을 파악합니다.	<a href="#">지원되는 AWS 리전</a>

다음을 수행하려는 경우 ...	읽기 ...
이 솔루션에 포함된 AWS CloudFormation 템플릿을 보거나 다운로드하여 이 솔루션의 인프라 리소스("스택")를 자동으로 배포합니다.	<a href="#">AWS CloudFormation 템플릿</a>
소스 코드에 액세스하고 선택적으로 AWS Cloud Development Kit(AWS CDK)를 사용하여 솔루션을 배포합니다.	<a href="#">GitHub 리포지토리</a>

## 기능 및 이점

AWS의 인스턴스 일정 솔루션은 다음과 같은 기능을 제공합니다.

### 교차 계정 인스턴스 예약

이 솔루션에는 보조 계정의 인스턴스를 시작하고 중지하는 데 필요한 [AWS Identity and Access Management\(IAM\)](#) 역할을 생성하는 템플릿이 포함되어 있습니다. 자세한 내용은 [교차 계정 인스턴스 예약](#) 섹션을 참조하세요.

### 자동 태그 지정

AWS 인스턴스 스케줄러는 시작하거나 중지하는 모든 인스턴스에 태그를 자동으로 추가할 수 있습니다. 이 솔루션에는 태그에 변수 정보를 추가할 수 있는 매크로도 포함되어 있습니다.

### 스케줄러 CLI를 사용하여 일정 또는 기간 구성

이 솔루션에는 일정 및 기간을 구성하기 위한 명령을 제공하는 명령줄 인터페이스(CLI)가 포함되어 있습니다. CLI를 사용하면 고객은 지정된 일정에 대한 비용 절감을 추정할 수 있습니다. 자세한 내용은 [스케줄러 CLI](#)를 참조하세요.

### 코드형 인프라(IaC)를 사용하여 일정 관리

이 솔루션은 코드형 인프라(IaC)를 사용하여 일정을 관리하는 데 사용할 수 있는 AWS CloudFormation CustomResource를 제공합니다. 자세한 내용은 [코드형 인프라\(IaC\)를 사용하여 일정 관리](#)를 참조하세요.

### Systems Manager 유지 관리 기간과의 통합

Amazon EC2 인스턴스의 경우 AWS 인스턴스 스케줄러는 해당 인스턴스와 동일한 리전에 정의된 [AWS Systems Manager](#) 유지 관리 기간과 통합하여 유지 관리 기간에 따라 인스턴스를 시작하고 중지할 수 있습니다.

## 사용 사례

### 근무 시간 중에만 인스턴스 실행

모든 인스턴스를 최대 사용률로 유지하면 이 솔루션은 정규 업무 시간 동안에만 필요한 인스턴스에 대해 최대 76%의 비용을 절감할 수 있습니다(주간 사용률이 168시간에서 40시간으로 단축됨). 자세한 내용은 [샘플 일정](#)을 참조하세요.

### 근무 시간 후 인스턴스 중지

몇 시간 후 다시 필요할 때까지 개발 인스턴스가 꺼져 있는지 확인하려면 이 솔루션을 사용하여 시작 시간 없이 종료 시간을 설정할 수 있습니다. 자세한 내용은 [샘플 일정](#)을 참조하세요.

## 개념 및 정의

이 섹션에서는 이 솔루션과 관련된 핵심 개념 및 용어에 대해 설명합니다.

### 일정

인스턴스가 바인딩되는 하나 이상의 기간 그룹입니다.

### period

시작 및 중지 시간으로 정의된 실행 기간입니다.

### 인스턴스

예약할 수 있는 지원되는 리소스입니다. 예를 들어, Amazon EC2 인스턴스 또는 Amazon RDS 클러스터 Amazon EC2 및 Amazon RDS가 있습니다.

### 정규 업무 시간

평일 9:00~17:00(오전 9시~오후 5시) 동부 표준시입니다.

AWS 용어에 대한 일반 참조는 [AWS 용어집](#)을 참조하세요.

## 비용

Instance Scheduler를 실행하는 동안 사용되는 AWS 서비스의 비용은 사용자가 부담합니다. 배포 크기에 따라 비용이 어떻게 조정되는지 이해하면 구현을 계획하고 최적화하는 데 도움이 됩니다.

## 비용 조정 요인

인스턴스 스케줄러 비용은 여러 요인에 따라 조정됩니다.

**예약 대상 수:** 관리 중인 고유한 account-region-service 조합의 수입니다. 각 대상에는 예약 간격당 별도의 Lambda 호출이 필요합니다.

**대상당 리소스:** 각 대상 내의 리소스(EC2 인스턴스, RDS 데이터베이스 등) 수는 Lambda 실행 시간 및 기간 비용에 영향을 미칩니다.

**운영 지표 복잡성:** 선택적 CloudWatch 지표 비용은 배포 전반에서 추적되는 고유한 인스턴스 유형 및 활성 일정의 수에 따라 조정됩니다.

**일정 빈도:** 솔루션은 구성된 빈도(기본값: 5분)에 따라 실행됩니다. 검사 빈도가 높을수록 Lambda 호출이 매일 24회(시간당)에서 매일 288회(5분 간격)로 증가합니다.

## 예약 대상 계산

예약 대상은 활성 관리형 인스턴스를 하나 이상 포함하는 account-region-service의 고유한 조합입니다. 동일한 account-region-service 조합 내의 여러 인스턴스는 단일 예약 대상으로 계산됩니다.

계산 예:

- 계정 A, us-east-1, EC2 인스턴스 5개 = 예약 대상 1개
- 계정 A, us-east-1, RDS 데이터베이스 3개 = 예약 대상 1개
- 계정 A, us-east-1, Auto Scaling 그룹 2개 = 예약 대상 1개
- 계정 A, us-west-2, EC2 인스턴스 2개 = 예약 대상 1개
- 계정 B, us-east-1, EC2 인스턴스 10개 = 예약 대상 1개

합계: 5개의 예약 대상

즉, 솔루션은 예약 간격당 5개의 개별 Lambda 함수를 호출하여 이러한 account-region-service 조합의 모든 리소스를 관리합니다.

### Note

대상은 일정 범위 내에 있을 수 있지만 하나 이상의 리소스에 해당 대상에서 일정 지정 태그가 지정될 때까지 비용 계산을 위한 "활성"으로 간주되지 않습니다.

비용 최적화를 위해 Instance Scheduler는 모든 Amazon RDS 관련 서비스를 단일 호출로 그룹화합니다. 따라서 Amazon RDS, [Amazon Aurora](#), [Amazon Neptune](#) 및 [Amazon DocDB](#) 예약은 모두 비용 계산을 위한 하나의 "RDS" 서비스로 계산됩니다.

## 비용 최적화 전략

1. Lambda 요금이 더 낮은 리전에 배포
2. 단일 대상 규모 제한으로 늘릴 필요가 없는 한 기본 512MB Lambda 메모리 설정 사용
3. 활성 사용 중인 고유한 일정 및 인스턴스 유형 수 최소화
4. 요구 사항에 따라 예약 빈도 조정
5. 운영 지표 대시보드를 사용할 계획이 없는 경우 비활성화

[이 솔루션의 각 AWS 서비스](#)에 대한 요금 웹 페이지를 참조하세요.

비용 관리에 도움이 되도록 AWS Cost Explorer를 통해 [예산](#)을 생성하는 것이 좋습니다. 요금은 변경될 수 있습니다.

## 참조 요금 예제(월별)

다음 예제에서는 다양한 배포 크기에서 비용이 어떻게 조정되는지 보여줍니다. 이를 참조 요소로 사용하여 특정 배포에 대한 비용을 추정합니다.

### Note

모든 참조 요금은 솔루션에서 사용하는 기본 서비스의 대략적인 비용 추정치입니다.

## 소규모 배포(월~\$9)

이 예제는 일반적인 개발 또는 소규모 프로덕션 배포를 나타냅니다.

- 활성 대상 5개
- 관리형 리소스 20개
- 3개의 활성 일정
- 인스턴스 유형 2개
- 5분 예약 간격

- 512MB Lambda 함수, 5초 평균 런타임

AWS 서비스	월별 비용(USD)
AWS Lambda	~\$2.00
KMS	약 1.50 USD
CloudWatch Logs	~\$0.30
CloudWatch 지표	~\$5.30
Amazon DynamoDB	~\$0.05
합계:	~\$9.15

## 중간 배포(월~\$161)

이 예제는 중간 규모의 엔터프라이즈 배포를 나타냅니다.

- 활성 대상 250개
- 1,000개의 관리형 리소스
- 15개의 활성 일정
- 인스턴스 유형 15개
- 5분 예약 간격
- 512MB Lambda 함수, 5초 평균 런타임
- EC2 유지 관리 기간 5개

AWS 서비스	월별 비용(USD)
AWS Lambda	~\$95.00
Amazon DynamoDB	~\$1.00
CloudWatch Logs	약 10.00 USD

AWS 서비스	월별 비용(USD)
CloudWatch 지표	약 40.00 USD
KMS	~\$15.00
합계:	~\$161.00

## 대규모 배포(월 최대 630 USD)

이 예제는 대규모 엔터프라이즈 배포를 나타냅니다.

- 활성 대상 1,000개
- 5,000개의 관리형 리소스
- 500개의 활성 일정
- 인스턴스 유형 50개
- 5분 예약 간격
- 512MB Lambda 함수, 5초 평균 런타임
- EC2 유지 관리 기간 100개

AWS 서비스	월별 비용(USD)
AWS Lambda	~\$380.00
Amazon DynamoDB	~\$5.00
CloudWatch Logs	~\$50.00
CloudWatch 지표	~\$140.00
KMS	~\$55.00
합계:	~\$630.00

## 배포에 대한 비용 추정

특정 배포에 대한 비용을 추정하려면:

1. 총 관리형 리소스 수(EC2 인스턴스, RDS 데이터베이스 등)
2. 관리할 계정 및 리전 수 결정
3. 필요한 예약 빈도 고려
4. 운영 지표가 필요한지 여부 결정
5. 위의 참조 예제를 사용하여 예상 비용을 보간합니다.

## 할당량

### 조정 제한 사항

인스턴스 스케줄러는 두 개의 기본 축을 확장하여 대규모 엔터프라이즈 배포를 관리합니다.

#### 수직 조정(대상당 리소스 수)

수직 조정은 단일 예약 요청 Lambda 함수가 단일 예약 대상(account/region/service 조합) 내에서 효율적으로 처리할 수 있는 리소스 수에 따라 제한됩니다.

인스턴스 스케줄러는 단일 [예약 대상](#)에서 1,000EC2s, 100개의 ASGs 및 100개의 RDS dbs/클러스터를 처리할 수 있도록 설계되었지만 리전 간 지연 시간으로 인해 제한될 수 있습니다.

최적의 성능을 보장하려면 예약 요청 Lambda의 실행 시간을 모니터링하는 것이 좋습니다([운영 인사이트 대시보드](#) 참조). 평균 런타임을 90초 미만으로 유지하고 최대 피크 시간은 4분 이하로 유지하는 것이 좋습니다.

#### 수평 조정(대상 수)

수평 조정은 관리 중인 [활성 예약 대상](#) 수에 따라 제한됩니다. 활성 대상은 활성 태그가 지정된 리소스가 하나 이상 있는 account/region/service 조합입니다. 인스턴스 스케줄러는 더 많은 계정 및 리전에 배포할 수 있지만, 활성 태그가 지정된 리소스가 있는 대상만 성능에 영향을 미칩니다.

기본 Lambda 동시성 할당량 1000을 사용하면 활성 대상 1000개를 동시에 실행할 수 있습니다. Lambda는 추가 실행을 자동으로 대기열에 추가하므로 이 제한을 초과하여 확장할 수 있습니다. 최적의 성능을 위해 누적 지연 시간을 3분 미만으로 유지하는 것이 좋습니다.

예를 들어 15초 평균 런타임 및 lambda 동시성 제한이 1000인 경우 누적 지연 시간을 3분(3분 ÷ 15초 × 1000 = 12000개 대상) 미만으로 유지하면서 최대 12000개의 활성 대상을 관리할 수 있습니다.

대규모 배포의 경우 AWS Support에 Lambda 동시성 할당량 증가를 요청할 수 있습니다.

## 추가 고려 사항

**AWS 리소스 태그:** AWS 리소스에는 일반적으로 리소스당 50개의 태그 제한이 있습니다. 인스턴스 스케줄러는 솔루션을 운영하기 위해 6개의 정보 및 제어 태그가 필요합니다. 리소스에 인스턴스 스케줄러 태그와 기존 태그 지정 전략을 모두 수용할 수 있는 충분한 태그 용량이 있는지 확인합니다.

**Lambda 실행 제한:** 각 예약 요청 핸들러 Lambda 함수의 실행 제한 시간은 5분입니다.

**DynamoDB 규모 조정:** 이 솔루션은 [Amazon DynamoDB](#) 테이블에 온디맨드 규모 조정을 사용하여 워크로드에 따라 용량을 자동으로 조정합니다.

**API 속도 제한:** AWS 서비스 API 제한은 매우 큰 배포에서 발생할 수 있습니다. 솔루션에는 임시 제한을 처리하기 위한 재시도 로직이 포함되어 있지만 과도한 제한은 솔루션의 조정 상한을 줄일 수 있습니다.

## AWS 서비스 할당량

### AWS 서비스에 대한 서비스 할당량

서비스 할당량(제한이라고도 함)은 AWS 계정의 최대 서비스 리소스 또는 작업 수입입니다. 이 솔루션에 구현된 각 서비스의 할당량이 충분한지 확인하세요. 자세한 내용은 [AWS 서비스 할당량](#)을 참조하세요.

### AWS CloudFormation 할당량

AWS 계정에는 이 솔루션에서 스택을 시작할 때 알아두어야 하는 AWS CloudFormation 할당량이 있습니다. 이러한 할당량을 이해하면 이 솔루션을 성공적으로 배포하지 못하는 제한 오류를 방지할 수 있습니다. 자세한 내용은 [AWS CloudFormation 사용 설명서의 AWS CloudFormation 할당량](#)을 참조하세요 AWS CloudFormation.

### AWS Lambda 할당량

계정의 기본 AWS Lambda 동시 실행 할당량은 1000입니다. 대규모 배포의 경우 Lambda 동시성을 위해 다른 워크로드와 경쟁하지 않도록 전용 계정에 인스턴스 스케줄러를 배포하는 것이 좋습니다. 이 값은 조정할 수 있습니다. 자세한 내용은 [AWS Lambda 시작하기 가이드](#)를 참조하세요.

## 지원되는 AWS 리전

AWS GovCloud(미국) 리전 및 일부 [옵트인 리전](#)(기본적으로 비활성화된 리전)을 포함하여 모든 AWS 리전에 인스턴스 스케줄러를 배포할 수 있습니다. 솔루션을 배포한 후 계정의 모든 리전에서 태그가 지정된 EC2 및 RDS DB 인스턴스에 적절한 시작 또는 중지 작업을 적용하도록 솔루션을 구성할 수 있습니다. 교차 계정 인스턴스 예약을 사용하는 경우 솔루션은 모든 계정에서 구성된 모든 리전의 인스턴스에 작업을 적용합니다.

### Important

AWS 인스턴스 스케줄러 작업은 Lambda 함수가 단일 리전에서 실행되더라도 계정의 모든 AWS 리전에서 적절하게 태그가 지정된 인스턴스에 영향을 줍니다.

솔루션의 여러 배포를 사용하여 많은 수의 인스턴스 또는 많은 계정 및 리전의 인스턴스를 예약할 수 있습니다. 여러 스케줄러를 배포할 때 각 스택에 대해 다른 태그 이름을 사용하고 각 배포에 대해 중첩되지 않는 리전 세트를 구성합니다.

각 배포는 계정의 구성된 모든 리전에 있는 모든 인스턴스에서 예약해야 하는 리소스를 식별하는 태그 키를 확인합니다. 여러 배포의 리전이 겹치는 경우 여러 배포에서 각 인스턴스를 확인합니다.

### Note

AWS의 인스턴스 스케줄러는 솔루션 스택이 표준 AWS 리전에 배포된 경우에도 예약을 위해 모든 옵트인 리전 내의 인스턴스를 대상으로 지정할 수 있습니다.

## 계정 ID 또는 AWS Organizations ID를 사용한 교차 계정 인스턴스 예약

이 솔루션에는 [AWS Identity and Access Management\(IAM\)](#) 역할과 솔루션이 보조 계정에서 예약을 시작하는 데 필요한 기타 리소스를 생성하는 템플릿([instance-scheduler-on-aws-remote.template](#))이 포함되어 있습니다. 스택을 시작하기 전에 원격 템플릿에서 권한을 검토하고 수정할 수 있습니다.

### 계정 ID를 사용한 교차 계정 예약 활성화

보조 계정의 리소스에 자동 시작-중지 일정을 적용하려면 다음 단계를 따르세요.

1. [AWS Management Console](#)에 로그인하고 버튼을 선택하여 기본 계정에서 [instance-scheduler-on-aws](#) AWS CloudFormation 템플릿을 시작합니다.
2. 해당하는 각 보조 계정에서 원격 템플릿([instance-scheduler-on-aws-remote](#))을 시작합니다. 각 원격 스택이 시작되면 교차 계정 역할 Amazon 리소스 이름(ARN)이 생성됩니다.
3. 조직 ID 또는 원격 계정 ID 목록 제공 파라미터에 계정 ID가 포함된 기본 솔루션 스택을 업데이트합니다. 이를 통해 솔루션이 보조 계정의 인스턴스에 대해 시작 및 중지 작업을 수행할 수 있습니다.

## AWS Organizations ID를 사용하여 교차 계정 예약 활성화

보조 계정의 리소스에 자동 시작-중지 일정을 적용하려면 다음 단계를 따르세요.

1. [AWS Management Console](#)에 로그인하고 버튼을 선택하여 기본 계정에서 [instance-scheduler-on-aws](#) AWS CloudFormation 템플릿을 시작합니다.
2. CloudFormation 파라미터 AWS Organizations 사용 여부를 예로 설정하고, 조직 ID 또는 원격 계정 ID 목록 제공 CloudFormation 파라미터에서 조직 ID를 제공합니다.
3. 기본 계정에 스택을 배포한 후 기본 계정의 솔루션과 동일한 리전의 해당하는 각 보조 계정에서 원격 템플릿 ([instance-scheduler-on-aws-remote](#))를 시작합니다. 각 원격 스택이 시작되면 기본 솔루션 계정이 기본 계정의 추가 변경 없이 계정 ID로 업데이트됩니다.

## AWS Systems Manager Parameter Store를 사용한 계정 ID 관리

AWS Systems Manager Parameter Store를 사용하여 원격 계정 ID를 저장합니다. 원격 계정 ID를 각 항목이 계정 ID인 목록 파라미터로 저장하거나, 심표로 구분된 원격 계정 ID 목록을 포함하는 문자열 파라미터로 저장할 수 있습니다. 파라미터의 형식은 {param:\_name\_}입니다. 여기서 이름은 Parameter Store의 파라미터 이름입니다.

이 기능을 활용하려면 Parameter Store와 동일한 계정에서 AWS 인스턴스 스케줄러 허브 스택을 시작해야 합니다.

## 예약에 지원되는 서비스

AWS 인스턴스 스케줄러는 현재 다음과 같은 서비스의 예약을 지원합니다.

- Amazon EC2
- Amazon EC2 Auto Scaling 그룹

- Amazon RDS
- Amazon Aurora 클러스터
- Amazon DocumentDB
- Amazon Neptune

## 인스턴스 종료 동작

### Amazon EC2

이 솔루션은 EC2 인스턴스를 자동으로 종료하도록 설계되었으며 인스턴스 종료 동작이 종료가 아닌 중지로 설정되어 있다고 가정합니다. Amazon EC2 인스턴스가 종료된 후에는 다시 시작할 수 없습니다.

기본적으로 EC2 인스턴스는 운영 체제가 종료될 때 종료되지 않고 중지되도록 구성되어 있지만, [이동작은 수정](#)할 수 있습니다. 따라서 AWS 인스턴스 스케줄러를 사용하여 제어하는 인스턴스가 종료 중지 동작으로 구성되어 있는지 확인합니다. 그렇지 않으면 종료됩니다.

### Amazon RDS, Amazon Neptune 및 Amazon DocumentDB

이 솔루션은 RDS, Neptune 및 DocDB 인스턴스를 삭제하지 않고 자동으로 중지하도록 설계되었습니다. 솔루션이 인스턴스를 중지하기 전에 RDS 인스턴스 스냅샷 생성 AWS CloudFormation 템플릿 파라미터를 사용하여 RDS DB 인스턴스의 스냅샷을 생성할 수 있습니다. 스냅샷은 다음에 인스턴스가 중지되고 새 스냅샷이 생성될 때까지 유지됩니다.

#### Note

Amazon Aurora 클러스터에는 스냅샷을 사용할 수 없습니다. Aurora 클러스터 예약 템플릿 파라미터를 사용하여 Aurora 클러스터의 일부이거나 Aurora 데이터베이스를 관리하는 RDS DB 인스턴스를 시작하고 중지할 수 있습니다. 초기 구성 중에 정의한 태그 키와 일정 이름을 해당 클러스터를 예약할 태그 값으로 사용하여 클러스터(개별 인스턴스 아님)에 태그를 지정해야 합니다.

RDS DB 인스턴스 시작 및 중지 제한에 대한 자세한 내용은 Amazon RDS 사용 설명서의 [Amazon RDS DB 인스턴스의 일시적 중지](#)를 참조하세요.

RDS DB 인스턴스가 중지되면 캐시가 지워져 인스턴스가 다시 시작될 때 성능이 느려질 수 있습니다.

## Amazon RDS 유지 관리 기간

모든 RDS DB 인스턴스에는 시스템 변경 내용이 적용되는 주 단위 [유지 관리 기간](#)이 있습니다. 유지 관리 기간 동안 Amazon RDS는 유지 관리를 적용하기 위해 7일 이상 중지된 인스턴스를 자동으로 시작합니다. Amazon RDS는 유지 관리 이벤트가 완료되면 인스턴스를 중지하지 않습니다.

솔루션을 사용하면 RDS DB 인스턴스의 기본 유지 관리 기간을 일정에 실행 기간으로 추가할지 여부를 지정할 수 있습니다. 다른 실행 기간에서 인스턴스를 실행하도록 지정하지 않고 유지 관리 이벤트가 완료된 경우 솔루션은 유지 관리 기간 시작 시 인스턴스를 시작하고 유지 관리 기간 종료 시 인스턴스를 중지합니다.

유지 관리 기간이 끝날 때까지 유지 관리 이벤트가 완료되지 않으면 유지 관리 이벤트가 완료된 후 예약 간격까지 인스턴스가 실행됩니다. Amazon RDS 유지 관리 기간에 대한 자세한 내용은 Amazon RDS 사용 설명서의 [DB 인스턴스 유지 관리](#)를 참조하세요.

## Amazon EC2 Auto Scaling 그룹

이 솔루션은 예약된 스케일링 작업을 사용하여 Amazon EC2 Auto Scaling 그룹을 자동으로 중지하도록 설계되었습니다. 솔루션을 사용하여 Auto Scaling 그룹(ASG)에서 예약된 스케일링 작업을 구성할 수 있습니다. 예약된 스케일링 작업에 의해 ASG가 중지되면 ASG가 자동으로 다시 시작될 때까지 최소, 원하는 및 최대 용량이 0으로 설정됩니다. 그러면 최소, 원하는 및 최대 용량이 원래 값으로 되돌아갑니다.

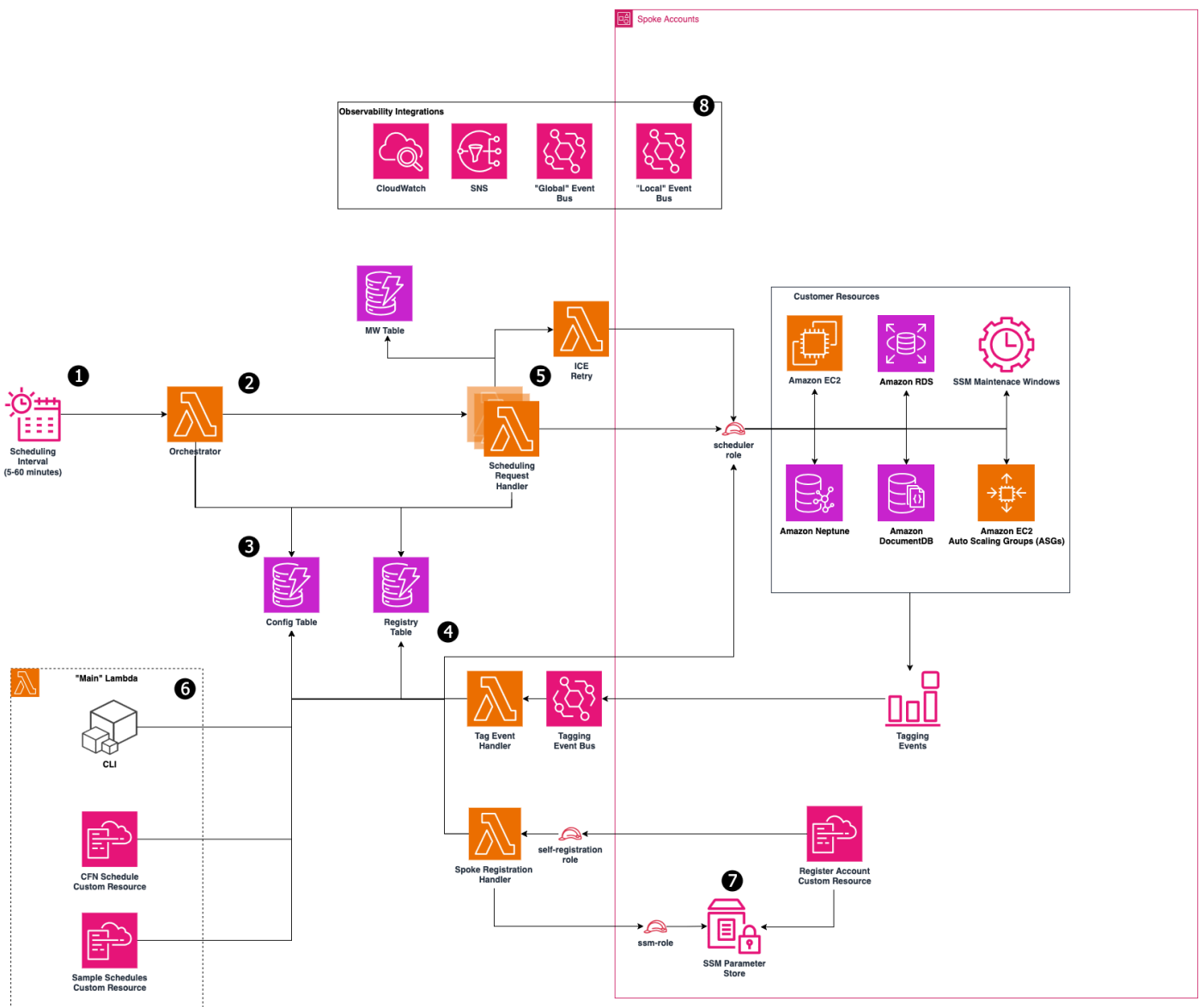
# 아키텍처

이 섹션에서는 참조 구현 아키텍처 다이어그램, [AWS Well-Architected 설계 고려 사항](#), [보안 구성 요소](#), [스케줄러 구성](#) 및 [이 솔루션에 사용되는 AWS 서비스](#)를 제공합니다.

## 아키텍처 다이어그램

이 솔루션을 배포하면 AWS 계정에 다음 구성 요소가 배포됩니다.

AWS 클라우드에서의 인스턴스 스케줄러



1. [Amazon EventBridge 규칙](#)은 구성 가능한 간격으로 오케스트레이션 Lambda 함수를 트리거합니다 (기본값: 5분마다).
2. EventBridge 규칙은 DynamoDB 구성 테이블을 쿼리하여 [활성 예약 대상](#)을 식별하는 [AWS Lambda](#) 오케스트레이션 함수를 호출합니다. 그런 다음 오케스트레이터는 각 활성 대상에 대해 병렬 예약 Lambda 함수를 호출합니다.
3. 일정 정의 및 기간은 [Amazon DynamoDB](#) 구성 테이블에 저장됩니다. 이 테이블에서 일정 및 기간을 원하는 수만큼 정의하여 인스턴스 시작 및 중지 시기를 제어할 수 있습니다.
4. DynamoDB 레지스트리 테이블은 모든 관리형 리소스를 자동으로 추적합니다. 리소스에 [예약 태그](#)가 지정되면 AWS 태그 지정 이벤트에 대한 응답으로 리소스가 테이블에 등록됩니다.
5. 각 예약 Lambda 함수는 태그가 지정된 리소스를 설명하고, 현재 시간을 기준으로 일정을 평가하고, 적절한 시작 또는 중지 작업을 실행합니다.
  - a. EC2 인스턴스의 경우 용량 부족으로 인해 시작 작업이 실패하면 시작 작업을 다시 시도하기 전에 인스턴스 크기를 [대체 인스턴스 유형](#)으로 조정하도록 솔루션을 구성할 수 있습니다.
6. 일정 관리는 [DynamoDB 콘솔](#), [스케줄러 CLI 도구](#) 또는 [AWS CloudFormation Custom 리소스를 통해](#) 사용할 수 있습니다. [??? AWS CloudFormation](#) 이 솔루션은 몇 가지 예제 일정이 미리 구성된 상태로 배포됩니다.
7. 교차 계정 배포는 스포크 계정이 허브 계정에 자동으로 자체 등록하는 허브 스포크 아키텍처를 사용합니다. 스포크 스택은 허브 스택과 동일한 리전에 배포해야 하며 허브 스택 또는 동일한 [AWS Organization](#)의 멤버의 사전 승인을 받아야 합니다.
8. 이 솔루션은 허브 계정(글로벌 [이벤트](#)) 및 스포크 계정(리전별 로컬 [이벤트](#))의 EventBridge 버스에 [예약 및 등록](#) 이벤트를 게시합니다. EventBridge

### Note

AWS CloudFormation 리소스는 [\(AWS CDK\)](#) 구문에서 생성됩니다.

이 솔루션에서 사용하는 모든 Lambda 함수는 리소스에 대한 권한 요구 사항에 AWS IAM을 활용하고 [Amazon Simple Notification Service](#)(Amazon SNS 주제) 및 DynamoDB 테이블의 암호화에 AWS KMS를 활용합니다.

솔루션이 간격 예약을 수행할 때마다 적절한 태그가 지정된 인스턴스의 현재 상태를 확인하여 관련된 일정에서 인스턴스 태그의 일정에 정의된 하나 이상의 [기간](#)으로 지정된 목표 상태와 비교합니다. 그 후 예약 간격에 따라 필요에 맞게 적절한 시작 또는 중지 작업이 적용됩니다.

예를 들어 Lambda 함수가 금요일 오전 9시(ET)에 간접 호출되고 Schedule=office-hours 태그로 지정된 중지 상태의 EC2 또는 RDS DB 인스턴스를 식별하면 Amazon DynamoDB에서 office-hours 일정 구성 세부 정보를 확인합니다. office-hours 일정에 인스턴스를 월요일부터 금요일까지 동부 표준시 기준 오전 9시부터 오후 5시까지 실행되어야 한다는 기간이 포함되어 있으면 Lambda 함수는 해당 인스턴스를 시작합니다.

또한 Lambda 함수는 리소스에 대한 정보를 기록하며, 이를 선택적으로 [Amazon CloudWatch Custom 대시보드](#)에 표시합니다. 기록된 정보에는 각 일정에 대해 태그가 지정된 인스턴스 수, 해당 인스턴스의 크기, 해당 인스턴스가 현재 실행 중인 상태인지 또는 중지된 상태인지 여부가 포함됩니다. 이 사용자 지정 대시보드에 대한 자세한 내용은 [운영 인사이트 대시보드](#)를 참조하세요.

### Note

Amazon EC2 인스턴스를 중지하는 것은 Amazon EC2 인스턴스를 종료하는 것과 다릅니다. 기본적으로 Amazon EC2 인스턴스는 운영 체제가 종료될 때 종료되지 않고 중지되도록 구성되어 있지만, 이 동작은 수정할 수 있습니다. 이 솔루션을 사용하기 전에 인스턴스가 적절하게 중지 또는 종료되도록 설정되어 있는지 확인합니다.

## AWS Well-Architected 설계 고려 사항

이 솔루션은 고객이 클라우드에서 안정적이고 안전하며 효율적이며 비용 효율적인 워크로드를 설계하고 운영할 수 있도록 지원하는 [AWS Well-Architected Framework](#)의 모범 사례를 바탕으로 설계되었습니다.

이 섹션에서는 이 솔루션을 구축할 때 Well-Architected Framework의 설계 원칙과 모범 사례가 어떻게 적용되었는지 설명합니다.

### 운영 우수성

이 섹션에서는 [운영 우수성 요소](#)의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 솔루션은 지표를 Amazon CloudWatch로 푸시하여 구성 요소(예: 인프라 및 Lambda 함수)에 대한 관찰성을 제공합니다.
- AWS X-Ray 추적 Lambda 함수.
- 오류 보고에 Amazon SNS를 사용합니다.

## 보안

이 섹션에서는 [보안 요소](#)의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 모든 서비스 간 통신은 IAM 역할을 사용합니다.
- 모든 다중 계정 통신은 IAM 역할을 사용합니다.
- 솔루션에서 사용하는 모든 역할은 최소 권한 액세스를 따릅니다. 즉, 여기에는 서비스가 제대로 작동하는 데 필요한 최소 권한만 포함됩니다.
- DynamoDB 테이블을 포함한 모든 데이터 스토리지에는 저장 시 데이터 암호화가 적용됩니다.

## 신뢰성

이 섹션에서는 [신뢰성 요소](#)의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 솔루션은 가능한 경우 서버리스 AWS 서비스(예: Lambda 및 DynamoDB)를 사용하여고가용성을 확보하고 서비스 장애로부터 복구합니다.
- 데이터 처리는 Lambda 함수를 사용합니다. 솔루션은 DynamoDB에 데이터를 저장하므로 기본적으로 여러 가용 영역에 유지됩니다.

## 성능 효율성

이 섹션에서는 [성능 효율성 요소](#)의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 솔루션은 서버리스 아키텍처를 사용합니다.
- 이 솔루션에 사용되는 AWS 서비스(예: Lambda 및 DynamoDB)를 지원하는 모든 AWS 리전에서 솔루션을 시작할 수 있습니다. 자세한 내용은 [지원되는 AWS 리전](#)을 참조하세요.
- 솔루션은 매일 자동으로 테스트 및 배포됩니다. 솔루션 아키텍트와 주제 전문가는 솔루션을 검토하여 실험하고 개선할 영역을 찾습니다.

## 비용 최적화

이 섹션에서는 [비용 최적화 요소](#)의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 솔루션은 서버리스 아키텍처를 사용하며, 고객은 사용한 만큼만 비용을 지불합니다.

- 컴퓨팅 계층은 기본적으로 종량제 모델을 사용하는 Lambda로 설정됩니다.

## 지속 가능성

이 섹션에서는 [지속 가능성 요소](#)의 원칙과 모범 사례를 사용하여 이 솔루션을 설계한 방법을 설명합니다.

- 솔루션은 관리형 및 서버리스 서비스를 사용하여 백엔드 서비스의 환경 영향을 최소화합니다.
- 솔루션의 서버리스 설계는 지속적으로 운영되는 온프레미스 서버의 탄소 발자국에 비해 탄소 배출을 줄이는 것을 목표로 합니다.

## 스케줄러 구성 테이블

배포되면, AWS 인스턴스 스케줄러는 전역 구성 설정을 포함하는 Amazon DynamoDB 테이블을 생성합니다.

전역 구성 항목에는 구성 테이블에서 값이 config인 유형의 속성이 포함됩니다. 일정 및 기간에는 각각 값이 schedule 및 period인 유형의 속성이 포함됩니다. DynamoDB 콘솔 또는 솔루션의 [명령줄 인터페이스](#)를 사용하여 구성 테이블에서 일정 및 기간을 추가, 업데이트 또는 제거할 수 있습니다. 그러나 유형이 config인 항목은 솔루션에서 관리하므로 편집할 수 없습니다.

## 스케줄러 CLI

솔루션에는 일정 및 기간을 구성하기 위한 명령을 제공하는 CLI가 포함되어 있습니다. CLI를 사용하면 특정 일정에 대한 비용 절감 예상치를 확인할 수 있습니다. 일정 CLI에서 제공하는 비용 추정치는 참고용으로만 제공됩니다. 스케줄러 CLI 구성 및 사용에 대한 자세한 내용은 [스케줄러 CLI](#)를 참조하세요.

## 이 솔루션에 사용되는 AWS 서비스

AWS 서비스	설명
<a href="#">Lambda</a>	Core. 솔루션은 인스턴스를 예약하고 사용자 지정 리소스 기능을 사용하여 CloudFormation 스택에 대한 업데이트를 관리하기 위한 모든 로직이 포함된 Lambda 함수를 배포합니다.

AWS 서비스	설명
<a href="#">Amazon DynamoDB</a>	Core. 솔루션은 일정 구성, 상태 정보, 인스턴스에 대해 수행된 마지막 작업을 저장하는 DynamoDB 테이블과 예약 목적으로 Systems Manager 유지 관리 기간을 저장하는 테이블을 생성합니다.
<a href="#">Amazon CloudWatch</a>	Core. 솔루션은 디버깅 및 정보 로그를 저장합니다.
<a href="#">AWS IAM</a>	Core. 솔루션은 IAM을 사용하여 인스턴스 예약 권한을 얻습니다.
<a href="#">Amazon SNS</a>	Core. 솔루션은 사용자가 구독하고 오류 발생 시 문제를 해결할 수 있도록 오류 메시지를 보내는 SNS 주제를 생성합니다.
<a href="#">KMS</a>	Core. 솔루션은 SNS 주제를 암호화하는 AWS KMS 키를 생성합니다.
<a href="#">Amazon EventBridge</a>	Core. 솔루션은 EventBridge를 생성하며, 일관된 간격으로 AWS lambda를 간접 호출하는 EventBridge 예약 규칙을 생성합니다.
<a href="#">AWS Systems Manager</a>	지원. 리소스 운영 및 비용 데이터에 대한 애플리케이션 수준의 리소스 모니터링 및 시각화를 제공합니다.
<a href="#">Amazon EC2</a>	예약. 솔루션은 EC2 인스턴스를 시작하고 중지하는 데 사용됩니다. 인스턴스는 솔루션에 구성된 특정 태그 키/값으로 식별됩니다.
<a href="#">Amazon RDS</a>	예약. 솔루션은 RDS DB 인스턴스 상태를 사용 가능 또는 중지됨으로 변경하는 데 사용됩니다. 인스턴스는 솔루션에 구성된 특정 태그 키/값으로 식별됩니다.

AWS 서비스	설명
<a href="#">Amazon Aurora</a>	<p>예약. 솔루션은 Aurora 클러스터 상태를 사용 가능 또는 중지됨으로 변경하는 데 사용됩니다. 클러스터는 솔루션에 구성된 특정 태그 키/값으로 식별됩니다.</p>
<a href="#">Amazon Neptune</a>	<p>예약. 솔루션은 Neptune 인스턴스 상태를 사용 가능 또는 중지됨으로 변경하는 데 사용됩니다. 인스턴스는 솔루션에 구성된 특정 태그 키/값으로 식별됩니다.</p>
<a href="#">Amazon DocumentDB</a>	<p>예약. 솔루션은 DocumentDB 인스턴스 상태를 사용 가능 또는 중지됨으로 변경하는 데 사용됩니다. 인스턴스는 솔루션에 구성된 특정 태그 키/값으로 식별됩니다.</p>
<a href="#">Amazon EC2 Auto Scaling 그룹</a>	<p>예약. 솔루션은 EC2 Auto Scaling 그룹에 대해 예약된 스케일링 규칙을 관리하는 데 사용됩니다. 이러한 규칙은 연결된 일정에 따라 Auto Scaling 그룹을 시작/중지합니다. 그룹은 솔루션에 구성된 특정 태그 키/값으로 식별됩니다.</p>

## 보안

AWS 인프라에 시스템을 빌드하면 보안 책임은 사용자와 AWS가 분담합니다. 이 [공동 책임 모델](#)은 운영 부담을 덜어줍니다. AWS가 호스트 운영 체제, 가상화 계층, 서비스 운영 시설의 물리적 보안을 포함한 구성 요소를 운영, 관리, 제어하기 때문입니다. AWS 보안에 대한 자세한 내용은 [AWS 클라우드 보안](#)을 참조하세요.

## KMS

솔루션은 SNS 주제 및 DynamoDB 테이블에 대한 서버 측 암호화를 구성하는 데 사용되는 AWS 관리형 고객 관리형 키를 생성합니다.

## Amazon IAM

솔루션의 Lambda 함수에는 허브 계정 리소스에 액세스하고 Systems Manager 파라미터를 가져오거나 입력하기 위한 액세스, CloudWatch 로그 그룹에 대한 액세스, AWS KMS 키 암호화/복호화 및 SNS 메시지 게시 권한이 필요합니다. 또한 인스턴스 스케줄러는 EC2, RDS, Autoscaling resurces, DB 인스턴스, 인스턴스 속성 수정 및 해당 리소스에 대한 태그 업데이트에 대한 액세스 권한을 제공하는 모든 관리형 계정에서 예약 역할을 생성합니다. 필요한 모든 권한은 솔루션 템플릿의 일부로 생성된 Lambda 서비스 역할에 대한 솔루션에서 제공됩니다.

배포 시 인스턴스 스케줄러는 배포된 허브 템플릿에서 Lambda를 특정 예약해야만 수입할 수 있는 스케줄러 역할과 함께 각 Lambda 함수에 대해 범위가 축소된 IAM 역할을 배포합니다. 이러한 일정 역할은 패턴 {namespace}-Scheduler-Role 및 {namespace}-ASG-Scheduling-Role을 따르는 이름을 갖습니다.

각 서비스 역할에 제공되는 권한에 대한 자세한 내용은 [CloudFormation 템플릿](#)을 참조하세요.

## 암호화된 EC2 EBS 볼륨

AWS KMS로 암호화된 EBS 볼륨에 연결된 EC2 인스턴스를 예약할 때는 Instance Scheduler에 연결된 AWS KMS 키(들)를 사용할 수 있는 권한을 부여해야 합니다. 이렇게 하면 Amazon EC2가 시작된 함수 중에 연결된 EBS 볼륨을 복호화할 수 있습니다. 키를 사용해 이 권한을 EC2 인스턴스와 동일한 계정의 예약 역할에 부여해야 합니다.

인스턴스 스케줄러에서 AWS KMS 키를 사용할 수 있는 권한을 부여하려면 키(들)를 사용하여 EC2 인스턴스(들)와 동일한 계정의 인스턴스 스케줄러 스택(허브 또는 스포크)에 AWS KMS 키의 ARN을 추가합니다.

### EC2용 KMS 키 ARN

#### Kms Key Arns for EC2

comma-separated list of kms arns to grant Instance Scheduler kms:CreateGrant permissions to provide the EC2 service with Decrypt permissions for encrypted EBS volumes. This allows the scheduler to start EC2 instances with attached encrypted EBS volumes. provide just (\*) to give limited access to all kms keys, leave blank to disable. For details on the exact policy created, refer to security section of the implementation guide (<https://aws.amazon.com/solutions/implementations/instance-scheduler-on-aws/>)

Enter CommaDelimitedList

이렇게 하면 다음과 같은 정책이 자동으로 생성되어 해당 계정의 예약 역할에 추가됩니다.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Condition": {
      "StringLike": {
        "kms:ViaService": "ec2.*.amazonaws.com"
      },
      "Null": {
        "kms:EncryptionContextKeys": "false",
        "kms:GrantOperations": "false"
      },
      "ForAllValues:StringEquals": {
        "kms:EncryptionContextKeys": [
          "aws:ebs:id"
        ],
        "kms:GrantOperations": [
          "Decrypt"
        ]
      },
      "Bool": {
        "kms:GrantIsForAWSResource": "true"
      }
    },
    "Action": "kms:CreateGrant",
    "Resource": [
      "Your-KMS-ARNs-Here"
    ],
    "Effect": "Allow"
  }
]
}

```

## EC2 라이선스 관리자

AWS License Manager에서 관리되는 EC2 인스턴스를 예약할 때 연결된 라이선스 구성을 사용할 수 있는 권한을 인스턴스 스케줄러에 부여해야 합니다. 이를 통해 솔루션은 라이선스 규정 준수를 유지하면서 인스턴스를 올바르게 시작하고 중지할 수 있습니다. License Manager를 사용하여 EC2 인스턴스(들)와 동일한 계정의 예약 역할에이 권한을 부여해야 합니다.

인스턴스 스케줄러와 함께 AWS License Manager를 사용할 수 있는 권한을 부여하려면 License Manager를 사용하여 EC2 인스턴스(들)와 동일한 계정의 인스턴스 스케줄러 스택(허브 또는 스포크)에 License Manager 구성 ARNs을 추가합니다.

## EC2용 License Manager 구성 ARNs

### License Manager Arns for EC2

comma-separated list of license manager arns to grant Instance Scheduler ec2:StartInstance permissions to provide the EC2 service with license manager permissions to start the instances. This allows the scheduler to start EC2 instances with license manager configuration enabled. Leave blank to disable. For details on the exact policy created, refer to security section of the implementation guide (<https://aws.amazon.com/solutions/implementations/instance-scheduler-on-aws/>)

Enter CommaDelimitedList

이렇게 하면 다음과 같은 정책이 자동으로 생성되어 해당 계정의 예약 역할에 추가됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ec2:StartInstances",
      "Resource": [
        "Your-License-Manager-ARNs-Here"
      ],
      "Effect": "Allow"
    }
  ]
}
```

License Manager 권한에 대한 자세한 내용은 [AWS License Manager 사용 설명서의 AWS License Manager에 대한 자격 증명 및 액세스 관리를](#) 참조하세요.

# 시작하기

이 가이드에는 솔루션을 빠르게 배포하기 위한 간략한 개요와 지침이 포함되어 있습니다. 이 솔루션은 [AWS CloudFormation 템플릿 및 스택](#)을 사용하여 솔루션의 배포를 자동화합니다. CloudFormation 템플릿은 이 솔루션에 포함된 AWS 리소스와 해당 속성을 지정합니다. CloudFormation 스택은 템플릿에 설명된 리소스를 프로비저닝합니다.

## 배포 프로세스 개요

### Important

이 솔루션에는 익명화된 운영 지표를 AWS로 전송하는 옵션이 포함되어 있습니다. 당사는 이 데이터를 사용하여 고객이 이 솔루션과 관련 서비스 및 제품을 어떻게 사용하는지 더 잘 이해합니다. AWS는 이 설문 조사를 통해 수집된 데이터를 소유합니다. 데이터 수집에는 [개인정보 취급방침](#)이 적용됩니다.

이 기능을 사용하지 않으려면 템플릿을 다운로드하고 AWS CloudFormation 매핑 섹션을 수정한 다음 AWS CloudFormation 콘솔을 사용하여 템플릿을 업로드하고 솔루션을 배포합니다.

이 섹션의 단계별 지침에 따라 솔루션을 구성하고 계정에 배포합니다.

배포 시간: 약 5~10분(구성 제외)

### 1단계: 인스턴스 스케줄러 스택 시작

1. AWS 계정에서 AWS CloudFormation 템플릿을 실행합니다.
2. 필수 파라미터에 값을 입력합니다.
3. 다른 템플릿 파라미터를 검토하고 필요한 경우 조정합니다.

### 2단계(선택 사항): 보조 계정에서 원격 스택 시작

1. AWS 계정에서 AWS CloudFormation 템플릿을 실행합니다.
2. 필수 파라미터에 값을 입력합니다.

## AWS CloudFormation 템플릿

이 솔루션은 [AWS CloudFormation 템플릿 및 스택](#)을 사용하여 솔루션의 배포를 자동화합니다. CloudFormation 템플릿은 이 솔루션에 포함된 AWS 리소스와 해당 속성을 지정합니다. CloudFormation 스택은 템플릿에 설명된 리소스를 프로비저닝합니다.

이 솔루션을 배포하기 전에 솔루션의 CloudFormation 템플릿을 다운로드할 수 있습니다.

### View template

instance-scheduler-on-aws.template - 이 템플릿을 사용하여 솔루션 및 모든 관련 구성 요소를 시작합니다. 기본 구성은 AWS Lambda 함수, Amazon DynamoDB 테이블, Amazon CloudWatch 이벤트 및 CloudWatch 사용자 지정 지표를 배포하지만, 특정 요구 사항에 따라 템플릿을 사용자 지정할 수도 있습니다.

### View template

instance-scheduler-on-aws-remote.template - 이 템플릿을 사용하여 솔루션이 스포크 계정의 인스턴스를 예약하는 데 사용되는 교차 계정 역할을 시작합니다. AWS Organizations를 사용하는 배포의 경우 템플릿을 배포하면 스포크 계정도 허브에 등록되므로 수동 구성이 필요하지 않습니다.

#### Note

이전에 이 솔루션을 배포한 경우 업데이트 지침은 [솔루션 업데이트](#)를 참조하세요.

## 1단계: 인스턴스 스케줄러 허브 스택 시작

이 섹션의 단계별 지침에 따라 솔루션을 계정에 배포합니다.

배포에 소요되는 시간: 약 5분

### Launch solution

1. [AWS Management Console](#)에 로그인하고 \*instance-scheduler-on-aws.template\* AWS CloudFormation 템플릿을 시작하는 버튼을 선택합니다.

- 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 실행하려면 콘솔 탐색 표시줄의 리전 선택기를 사용합니다.
- 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
- 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 이름 지정 시 적용되는 문자 제한에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 [IAM 및 AWS STS 할당량](#)을 참조하세요.
- 파라미터에서 이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	기본값	설명
인프라		
네임스페이스	default	여러 솔루션 배포를 구분하고 유 식별자를 공백 없이 입력합니다(예: Dev).
AWS Organizations 사용	No	AWS Organizations을 사용하여 스포크 계정 등록을 자동화합니다.
조직 ID/원격 계정 ID	<선택 사항 입력>	AWS Organizations을 사용하는 경우 이 필드는 필수입니다. 조직 ID를 입력합니다. 예: o-xxxxxyyy . 그렇지 않으면 다음과 같이 예약에 자신을 등록할 수 있는 신뢰할 수 있는 스포크 계정 IDs의 쉼표로 구분된 목록(최대 40개)을 제공합니다. 1111111111, 2222222222
일정 태그 키	Schedule	리소스 일정을 결정하기 위해 솔루션이 읽는 태그 키입니다. 리소스의 값은 일정의 이름을 지정합니다. 기본값을 수정하

파라미터	기본값	설명
		기로 선택한 경우 필요한 모든 인스턴스에 일관적으로 올바르게 적용할 수 있는 이름을 할당합니다. 참고: 태그 키는 대/소문자를 구분합니다.
데이터 및 로그 보존	Enabled	솔루션에서 사용하는 DynamoDB 테이블에 대해 삭제 방지를 활성화합니다. 이렇게 하면 이 스택을 삭제하더라도 테이블이 유지됩니다. 이 스택을 삭제하면서 해당 테이블도 삭제하려면 먼저 이 파라미터를 비활성화합니다.
<b>글로벌 설정</b>		
예약 활성화	Yes	모든 예약 작업을 일시 중지No하려면 로 설정합니다.
기본 시간대	UTC	시간대를 지정하지 않은 일정에 적용되는 기본 IANA(International assigned Numbers Authority) 시간대 식별자입니다. 유효한 시간대 식별자 목록은 <a href="#">List of tz database time zones</a> 의 TZ identifier 열을 참조하세요.

파라미터	기본값	설명
예약 간격(분)	5	스케줄러 실행 사이의 분 간격입니다. 간격이 짧을수록 정확도와 응답성이 향상되지만 비용도 증가합니다. 프로덕션 배포는 안정적인 작동을 위해 최소 5분이 필요하며, 더 짧은 값은 소규모 테스트에만 해당됩니다.
EC2 SSM 유지 관리 기간 활성화	No	일정에서 하나 이상의 Systems Manager 유지 관리 기간 이름을 지정할 수 있도록 허용합니다. 그 후 AWS 인스턴스 스케줄러는 해당 일정이 태그가 지정된 인스턴스가 연결된 유지 관리 기간 시작 최소 10분 전에 시작되도록 보장합니다.
중지 시 RDS 인스턴스 스냅샷 생성	No	RDS DB 인스턴스를 중지하기 전에 스냅샷을 생성할지 여부를 선택합니다. 참고: Amazon Aurora 클러스터에는 스냅샷을 사용할 수 없습니다.
ASG 작업 이름 접두사	IS-	Auto Scaling 그룹에 대해 예약된 스케일링 작업의 이름을 지정할 때 솔루션이 사용하는 접두사입니다. 이 접두사가 있는 작업은 필요에 따라 솔루션에 의해 추가 및 제거됩니다.
ASG 예약 태그 키	scheduled	사용되지 않음. 이 파라미터는 마이그레이션 목적으로만 존재하며 편집해서는 안 됩니다.

파라미터	기본값	설명
허브 계정 예약		
리전	<선택 사항 입력>	인스턴스가 예약될 리전 목록입니다. 예: us-east-1 ,us-west-1 . 참고: 이 파라미터를 비워 두면 솔루션이 현재 리전을 사용합니다.
EC2용 KMS 키 ARN	<선택 사항 입력>	암호화된 EBS 볼륨에 대해 EC2 서비스에 복호화 권한을 제공할 수 있도록, AWS 인스턴스 스케줄러에 kms:CreateGrant 권한을 부여할 KMS ARN을 쉼표로 구분한 목록입니다. 이렇게 하면 스케줄러가 암호화된 EBS 볼륨이 연결된 EC2 인스턴스를 시작할 수 있습니다. 모든 KMS 키에 제한된 액세스 권한을 부여하려면 (*)를 입력하고, 비활성화하려면 비워 둡니다. 생성된 정책에 대한 자세한 내용은 <a href="#">암호화된 EC2 EBS 볼륨</a> 을 참조하세요.
EC2용 License Manager ARNs	<선택 사항 입력>	License Manager에서 관리하는 EC2 인스턴스를 시작할 수 있는 권한을 인스턴스 스케줄러에 부여하는 License Manager 구성 ARNs의 쉼표로 구분된 목록입니다. 비활성화하려면 비워 둡니다. 자세한 내용은 <a href="#">EC2 License Manager</a> 를 참조하세요.

파라미터	기본값	설명
모니터링		
정보 태그 지정 활성화	Yes	활성화된 경우 Instance Scheduler는 마지막으로 수행된 예약 작업과 발생한 오류를 나타내는 정보 태그를 관리형 리소스에 기록합니다. 자세한 내용은 <a href="#">정보 태그</a> 를 참조하세요.
CloudWatch 디버그 로그 활성화	No	CloudWatch 로그에서 디버그 수준 로깅을 활성화합니다.
로그 보존 기간(일)	30	CloudWatch 로그에 대한 로그 보존 기간(일)입니다.
운영 모니터링	Enabled	CloudWatch에 운영 인사이트 대시보드를 배포하고 솔루션 운영에 대한 사용자 지정 지표 데이터를 수집합니다. 원하는 경우 <a href="#">관련 비용을</a> 줄이기 위해 대시보드를 비활성화할 수 있습니다.
기타		
SchedulingRequestHandler 메모리 크기(MB)	512	리소스를 예약하는 AWS Lambda 함수의 메모리 크기입니다. 메모리 사용량이 많거나 제한 시간이 초과되면 메모리 크기를 늘리세요.

파라미터	기본값	설명
오케스트레이터 메모리 크기 (MB)	512	오케스트레이터 Lambda 함수의 메모리 크기입니다. 메모리 사용량이 많거나 제한 시간이 초과되면 메모리 크기를 늘리세요.

- 다음을 선택합니다.
- 스택 옵션 구성 페이지에서 다음을 선택합니다.
- 검토 및 생성 페이지에서 설정을 검토하고 확인합니다. 템플릿이 IAM 리소스를 생성할 것임을 확인하는 확인란을 선택합니다.
- 제출을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 5분 후에 CREATE\_COMPLETE 상태를 받게 됩니다.

## 2단계(선택 사항): 보조 계정에서 원격 스택 시작

### Important

원격 스택은 허브 스택과 동일한 리전에 배포해야 합니다.

이 자동화된 AWS CloudFormation 템플릿은 허브 스택이 다른 계정의 인스턴스를 예약할 수 있도록 보조 계정 권한을 구성합니다. 기본/허브 스택이 허브 계정에 성공적으로 설치된 후에만 원격 템플릿을 설치합니다.

### Launch solution

- 해당 보조 계정의 AWS Management Console에 로그인하고 instance-scheduler-on-aws-remote AWS CloudFormation 템플릿을 시작하는 버튼을 선택합니다.
- 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 실행하려면 콘솔 탐색 표시줄의 리전 선택기를 사용합니다. 허브 스택이 AWS Organizations을 사용하도록 구성되어 있는 경우 허브 스택과 동일한 리전에 원격 템플릿을 배포합니다.

3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 세부 정보 지정 페이지에서 원격 스택 이름을 할당합니다.
5. 파라미터에서 템플릿의 파라미터를 검토하고 수정합니다.
6. AWS Organizations 옵션이 활성화되어 있고 허브 스택이 유사하게 구성되어 있는 경우 예약을 시작하기 위해 기본 스택에서 추가로 변경할 사항은 없습니다.
7. AWS Organizations 옵션이 아니므로 설정되어 있는 경우 허브 스택을 새 계정 ID로 업데이트해야 합니다.

파라미터	기본값	설명
인프라		
네임스페이스	default	여러 솔루션 배포를 구분하는데 사용되는 고유 식별자입니다. 허브 스택과 동일한 값으로 설정해야 합니다.
AWS Organizations 사용	No	AWS Organizations을 사용하여 스포크 계정 등록을 자동화합니다. 허브 스택과 동일한 값으로 설정해야 합니다.
허브 계정 ID	<입력 필수>	이 계정의 리소스를 예약할 AWS 인스턴스 스케줄러 허브 스택의 계정 ID입니다.
일정 태그 키	Schedule	리소스 일정을 결정하기 위해 솔루션이 읽는 태그 키입니다. 허브 스택과 동일한 값으로 설정해야 합니다.
멤버 계정 예약		
리전	<선택 사항 입력>	인스턴스가 예약될 리전 목록입니다. 예: us-east-1 ,us-west-1 .(허브와 동일한 리전

파라미터	기본값	설명
		<p>목록일 필요는 없음). 이 파라미터를 비워 두면 솔루션에서 현재 리전을 사용합니다.</p>
<p>EC2용 KMS 키 ARN</p>	<p>&lt;선택 사항 입력&gt;</p>	<p>쉼표로 구분된 KMS ARN 목록으로, 해당 솔루션에 kms:CreateGrant 권한을 부여하여 암호화된 EBS 볼륨에 대한 복호화 권한을 EC2 서비스에 제공합니다. 이렇게 하면 스케줄러가 암호화된 EBS 볼륨이 연결된 EC2 인스턴스를 시작할 수 있습니다. 모든 KMS 키에 제한된 액세스 권한을 부여하려면 (*)를 입력하고, 비활성화하려면 비워 둡니다. 자세한 내용은 <a href="#">암호화된 EC2 EBS 볼륨을 참조하세요</a>.</p>
<p>EC2용 License Manager ARNs</p>	<p>&lt;선택 사항 입력&gt;</p>	<p>License Manager에서 관리하는 EC2 인스턴스를 시작할 수 있는 권한을 인스턴스 스케줄러에 부여하는 License Manager 구성 ARNs의 쉼표로 구분된 목록입니다. 비활성화하려면 비워 둡니다. 자세한 내용은 <a href="#">EC2 License Manager</a>를 참조하세요.</p>

1. 다음을 선택합니다.
2. 옵션 페이지에서 다음을 선택합니다.
3. 검토 및 생성 페이지에서 설정을 검토하고 확인합니다. 템플릿이 IAM 리소스를 생성한다는 것을 확인하는 확인란을 선택해야 합니다.
4. 제출을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 5분 후에 CREATE\_COMPLETE 상태를 확인할 수 있습니다.

## 솔루션 구성

이제 솔루션이 배포되었으므로 스케줄러에 대한 일정 구성 및 인스턴스 태그 지정을 시작할 수 있습니다. 이러한 작업을 수행하는 방법에 대해 자세히 알아보려면 [일정 구성](#) 및 [예약을 위한 인스턴스 태그 지정](#)을 참조하세요.

# 운영자 안내서

이 가이드는 이 솔루션의 사용자와 운영자를 대상으로 하며 [일정을 구성](#)하고 [솔루션을 모니터링하는](#) 방법에 대한 세부 정보를 포함합니다.

## 일정 구성

### Important

일정을 잘못 구성하면 인스턴스가 지속적으로 실행되어 예기치 않은 비용이 발생할 수 있습니다. 리소스에 일정을 적용하기 전에 다음을 확인합니다.

- 리소스 태그의 일정 이름은 구성 테이블에 정의된 일정과 정확히 일치합니다. 일정 이름을 잘못 입력하거나 존재하지 않으면 UnknownSchedule 오류가 발생하고 스케줄러가 인스턴스를 중지하지 않습니다. 리소스의 IS-Error 태그를 확인하여 이 조건을 식별합니다.
- stop\_new\_instances가 로 설정된 경우 첫 번째 태그가 지정된 예약된 기간 외에 실행되는 false 인스턴스는 예약된 다음 중지 전환까지 중지되지 않습니다. 이로 인해 인스턴스가 예상보다 오래 실행될 수 있습니다.
- retain\_running이 로 설정된 경우 실행 기간이 시작되기 전에 수동으로 시작된 true 인스턴스는 해당 기간이 끝날 때 중지되지 않습니다. 이는 설계상 제공되지만 모니터링하지 않으면 인스턴스가 무기한 실행될 수 있습니다.
- enforced: false (기본값)를 사용하는 경우 스케줄러는 실행 기간 동안 수동으로 중지된 인스턴스를 다시 시작하지 않으며 초기 중지 전환 후 실행 기간 이후에 수동으로 시작된 인스턴스를 중지하지 않습니다.

[정보 태그 지정](#)(기본적으로 활성화됨)을 활성화하고 리소스의 IS-Error 및 IS-LastAction 태그를 주기적으로 검토하여 예약이 예상대로 작동하는지 확인하는 것이 좋습니다.

솔루션이 성공적으로 배포되면 일정 구성을 시작할 수 있습니다. AWS 인스턴스 스케줄러는 아래 설명과 같이 두 가지 일정 관리 방법을 지원합니다.

**Note**

솔루션은 일정 수를 제한 없이 지원할 수 있으며, 각 일정에는 해당 일정으로 제어되는 인스턴스가 실행되어야 하는 시기를 정의하는 기간이 하나 이상 포함될 수 있습니다. 자세한 내용은 [일정 및 기간](#)을 참조하세요.

## 코드형 인프라 사용(권장)

AWS 인스턴스 스케줄러는 코드형 인프라(IaC)를 사용하여 일정 및 기간을 관리하는 데 사용할 수 있는 AWS CloudFormation CustomResource를 제공합니다.

IaC를 사용하여 일정을 관리하는 방법에 대한 자세한 내용은 [코드형 인프라\(IaC\)를 사용하여 일정 관리](#)를 참조하세요.

## Amazon DynamoDB 콘솔 및 AWS 인스턴스 스케줄러 CLI 사용

**Important**

사용자 지정 리소스를 사용하여 IaC로 일정을 관리하는 경우 DynamoDB 콘솔 또는 스케줄러 CLI를 사용하여 해당 일정 또는 기간을 삭제하거나 수정해서는 안 됩니다. 이렇게 하면 CloudFormation에 저장된 파라미터와 테이블의 값 간에 충돌이 발생합니다. 또한 DynamoDB 콘솔 또는 스케줄러 CLI를 사용하여 생성된 일정에 CloudFormation에서 관리하는 기간을 사용하지 마십시오.

AWS 인스턴스 스케줄러 허브 스택을 배포할 때 솔루션은 사용자 지정 기간 및 일정을 생성하는 데 참조로 사용할 수 있는 여러 샘플 기간 및 일정이 포함된 Amazon DynamoDB 테이블을 생성했습니다. DynamoDB에서 일정을 생성하려면 구성 테이블(ConfigTable)에서 일정 중 하나를 수정하거나 새 일정을 생성합니다. CLI를 사용하여 일정을 생성하려면 먼저 [스케줄러 CLI를 설치](#)한 다음 사용 [가용 명령](#)을 사용합니다.

**Note**

IaC, DynamoDB 및 InstanceScheduler CLI를 사용하여 여러 샘플 일정을 생성하는 방법의 예는 [Sample schedules](#)를 참조하세요.

이 섹션에서는 솔루션 사용, 모니터링 및 업데이트 방법과 문제 해결 및 지원 정보에 대한 지침과 참조 자료를 제공합니다.

## 예약을 위한 인스턴스 태그 지정

AWS CloudFormation 템플릿을 배포할 때 솔루션의 사용자 지정 태그에 대한 이름(태그 키)을 정의했습니다. AWS 인스턴스 스케줄러가 Amazon EC2 또는 Amazon RDS 인스턴스를 인식하려면 해당 인스턴스의 태그 키가 이 사용자 지정 태그 키와 일치해야 합니다. 따라서 해당하는 모든 인스턴스에 태그를 일관적으로 올바르게 적용하는 것이 중요합니다. 이 솔루션을 사용하는 동안 인스턴스에 대한 기존 [태그 지정 모범 사례](#)를 계속 사용할 수 있습니다. 자세한 내용은 [Amazon EC2 리소스 태깅](#) 및 [Amazon RDS 리소스에 태그 지정](#)을 참조하세요.

AWS Management Console에서 [Tag Editor](#)를 사용하여 한 번에 여러 리소스에 대한 태그를 적용하거나 수정합니다. 콘솔에서 수동으로 태그를 적용하고 수정할 수도 있습니다.

리소스에 태그를 지정한 직후 인스턴스 스케줄러가 리소스에 IS-ManagedBy 태그를 적용하여 리소스가 이제 스케줄러에서 관리되고 있음을 나타냅니다. 이 태그를 찾아 리소스가 예약에 올바르게 등록되었는지 확인할 수 있습니다.

## 태그 값 설정

인스턴스에 태그를 적용할 때는 초기 구성 중에 정의한 태그 키(기본적으로 태그 키는 Schedule임)를 사용하고 태그 값을 인스턴스에 적용해야 하는 일정 이름으로 설정합니다. 태그 키를 변경하려면 [솔루션 파라미터를 업데이트](#)하면 됩니다.

### Note

Amazon RDS 인스턴스의 경우 태그 값은 길이가 1~256자(유니코드 문자)이며 aws:로 시작할 수 없습니다. 문자열에는 유니코드 문자 집합, 숫자, 공백, “\_”, “.”, “/”, “=”, “”, “-”(Java 정규식: `^[\\p{L}\\p{Z}\\p{N}_./=-]*$`) 기호만 포함될 수 있습니다. 자세한 내용은 [Amazon RDS 리소스에 태그 지정](#)을 참조하세요.

## 암호화된 EBS 볼륨이 포함된 EC2 인스턴스

EC2 DB 인스턴스에 고객 관리형 KMS 키로 암호화된 EBS 볼륨이 있는 경우 인스턴스 스케줄러 역할에 해당 인스턴스를 시작할 수 있는 KMS:CreateGrant 권한을 부여해야 합니다. 자세한 내용은 [암호화된 EC2 EBS 볼륨](#)을 참조하세요.

## License Manager에서 관리되는 EC2 인스턴스

EC2 인스턴스가 AWS License Manager에서 관리되는 경우 인스턴스 스케줄러 역할에 해당 인스턴스를 시작하고 중지할 수 있는 적절한 License Manager 권한을 부여해야 합니다. 자세한 내용은 [EC2 License Manager](#)를 참조하세요.

### 일정 추론

일정은 해당 일정으로 태그가 지정된 인스턴스를 실행해야 하는 시기를 지정합니다. 각 일정에는 태그가 지정된 리소스에 적용할 일정을 식별하는 태그 값으로 사용되는 고유한 이름이 있어야 합니다.

### 기간

각 일정에는 인스턴스가 실행되어야 하는 시기를 정의하는 기간이 하나 이상 포함되어야 합니다. 일정은 기간을 두 개 이상 포함할 수 있습니다. 일정에 두 개 이상의 기간이 사용될 때 AWS 인스턴스 스케줄러는 기간 중 하나 이상이 true일 때 적절한 시작 작업을 적용합니다. 자세한 내용은 [기간 참조](#)를 참조하세요.

### 시간대

일정의 시간대를 지정할 수도 있습니다. 시간대를 지정하지 않으면 솔루션을 시작할 때 지정한 기본 시간대가 일정에 사용됩니다. 허용 가능한 시간대 값 목록은 [List of tz database time zones](#)의 TZ 열을 참조하세요.

### 새 인스턴스 중지 필드

stop\_new\_instances 필드는 인스턴스 스케줄러가 현재 실행 기간 외에 실행 중인 경우 예약에 대한 태그가 처음 지정될 때 인스턴스를 중지해야 하는지 여부를 제어합니다. 기본적으로 이 필드는 true로 설정되어 있습니다.

true로 설정하면 예약된 실행 기간 이외의 실행 중인 인스턴스에 태그를 지정하면 인스턴스 스케줄러가 인스턴스를 즉시 중지합니다. false로 설정하면 인스턴스 스케줄러는 예약된 다음 중지 시간까지 인스턴스를 실행 상태로 유지합니다.

### 최대 절전 모드 필드

최대 절전 모드 필드를 사용하면 중지된 Amazon EC2 인스턴스에 최대 절전 모드를 사용할 수 있습니다. 이 필드가 true로 설정된 경우 EC2 인스턴스는 최대 절전 모드를 지원하는 Amazon Machine Image(AMI)를 사용해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서에서 [지원되는 Linux AMI](#)를

참조하세요. 최대 절전 모드는 인스턴스 메모리(RAM)의 내용을 Amazon Elastic Block Store(Amazon EBS) 루트 볼륨에 저장합니다. 이 필드를 true로 설정하는 경우 솔루션이 인스턴스를 중지할 때 중지되는 대신 인스턴스가 최대 절전 모드로 전환됩니다.

최대 절전 모드를 사용하도록 솔루션을 설정했지만 인스턴스가 [최대 절전 모드로 구성](#)되지 않았거나 [최대 절전 모드 사전 조건](#)을 충족하지 않는 경우 솔루션은 경고를 로깅하고 인스턴스는 최대 절전 모드 없이 중지됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 [온디맨드 인스턴스 또는 스팟 인스턴스 최대 절전 모드](#)를 참조하세요.

## 적용 필드

일정에는 인스턴스가 실행 기간 외에 수동으로 시작되거나 실행 기간 중에 수동으로 중지되는 것을 방지할 수 있는 적용 필드가 포함되어 있습니다. 이 필드가 true로 설정되어 있고 사용자가 실행 기간 외에 인스턴스를 수동으로 시작하는 경우 솔루션은 인스턴스를 중지합니다. 이 필드가 true로 설정된 경우 실행 기간 동안 수동으로 중지된 인스턴스도 다시 시작됩니다.

## 실행 중인 필드 유지

retain\_running 필드는 기간이 시작되기 전에 인스턴스가 수동으로 시작된 경우 실행 기간이 끝날 때 솔루션이 인스턴스를 중지하지 못하도록 합니다. 예를 들어, 오전 9시부터 오후 5시까지 실행되는 기간이 있는 인스턴스가 오전 9시 이전에 수동으로 시작된 경우 솔루션은 오후 5시에 인스턴스를 중지하지 않습니다.

## Systems Manager 유지 관리 기간 필드(EC2 인스턴스에만 적용됨)

ssm-maintenance-window 필드를 사용하면 AWS Systems Manager 유지 관리 기간을 일정에 실행 기간으로 자동으로 추가할 수 있습니다. Amazon EC2 인스턴스와 동일한 계정 및 AWS 리전에 있는 유지 관리 기간의 이름을 지정하면 솔루션은 유지 관리 기간이 시작되기 최소 10분 전에 인스턴스를 시작하고, 인스턴스를 실행하도록 지정하는 다른 실행 기간이 없는 경우 유지 관리 기간이 끝날 때 인스턴스를 중지합니다.

SSM 유지 관리 기간이 생성되고 일정이 SSM 유지 관리 기간의 이름으로 구성되면 변경 사항은 다음 순서로 예약된 Lambda가 실행될 때 반영됩니다. 예를 들어, 스케줄러 Lambda를 실행할 빈도를 5분으로 선택한 경우 Lambda가 다음 간격(5분 뒤)에 유지 관리 기간의 변경 사항을 적용합니다.

AWS 인스턴스 스케줄러는 유지 관리 기간이 시작되기 최소 10분 전에 인스턴스가 시작되도록 합니다. 예약 간격 AWS CloudFormation 파라미터에 대해 설정한 값에 따라 인스턴스가 최소 10분 전에 시작되도록 하기 위해 유지 관리 기간 시작보다 10분 일찍 시작될 수 있습니다. 예를 들어, 예약 간격을 30분으로 설정하면 스케줄러는 유지 관리 기간이 시작되기 10~40분 전에 인스턴스를 시작합니다.

**Note**

이 기능을 사용하려면 솔루션 허브 스택의 EC2 SSM 유지 관리 기간 CloudFormation 활성화 파라미터를 yes로 설정해야 합니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 [AWS Systems Manager Maintenance Windows](#)를 참조하세요.

## 인스턴스 유형

Amazon EC2 인스턴스에 한하여, 일정을 사용하면 일정의 각 기간에 대해 원하는 선택적 인스턴스 유형을 지정할 수 있습니다. 기간에 인스턴스 유형을 지정하면 솔루션은 요청된 인스턴스 유형과 일치하도록 EC2 인스턴스의 크기를 자동으로 조정합니다.

인스턴스 유형을 지정하려면 `<period-name>@<instance-type>` 명령문을 사용합니다. 예를 들어 `weekendset2.nano`입니다. Amazon EC2 인스턴스 및 Amazon RDS 인스턴스를 예약하는 기간 동안 인스턴스 유형을 지정하는 경우 Amazon RDS 인스턴스에 대한 인스턴스 유형이 무시됩니다.

실행 중인 인스턴스의 인스턴스 유형이 해당 기간에 지정된 인스턴스 유형과 다른 경우 솔루션은 실행 중인 인스턴스를 중지하고 지정된 인스턴스 유형으로 인스턴스를 다시 시작합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스 유형 변경](#)을 참조하세요.

## 일정 정의

Amazon DynamoDB의 AWS 인스턴스 스케줄러 구성 테이블에는 일정 정의가 포함되어 있습니다. 일정 정의에는 다음과 같은 필드가 포함될 수 있습니다.

필드	설명
description	일정에 대한 설명입니다(선택 사항).
hibernate	Amazon Linux를 실행하는 Amazon EC2 인스턴스를 최대 절전 모드로 전환할지 여부를 선택합니다. 이 필드가 true로 설정되면 스케줄러는 인스턴스를 중지할 때 인스턴스를 최대 절전 모드로 전환합니다. 인스턴스는 <a href="#">최대 절전 모드를 켜야 하며</a> , <a href="#">최대 절전 모드 사전 조건</a> 을 충족해야 합니다.

필드	설명
enforced	일정을 적용할지 여부를 선택합니다. 이 필드를 true로 설정하면 스케줄러가 실행 기간 이후에 수동으로 시작된 인스턴스를 중지하거나 실행 기간 중에 수동으로 중지된 인스턴스를 시작합니다.
name	일정을 식별하는 데 사용되는 이름입니다. 이 이름은 고유해야 하며 영숫자, 하이픈(-) 및 밑줄(_)만 포함해야 합니다.
periods	이 일정에 사용되는 기간의 이름입니다. 기간 이름 필드에 표시된 대로 이름을 정확히 입력합니다.  <period-name>@<instance-type> 명령문을 사용하여 해당 기간의 인스턴스 유형을 지정할 수도 있습니다. 예를 들어 weekdays@t2.large 입니다.
retain_running	기간이 시작되기 전에 인스턴스가 수동으로 시작된 경우 실행 기간이 끝날 때 솔루션이 인스턴스를 중지하지 않도록 할지 여부를 선택합니다.
use_maintenance_window	Amazon RDS 인스턴스 일정의 실행 기간으로 Amazon RDS 유지 관리 기간을 포함할지, 아니면 Amazon EC2 인스턴스 일정의 실행 기간으로 AWS Systems Manager 유지 관리 기간을 포함할지 여부를 선택합니다. 이 필드는 기본적으로 활성화되어 있으며 값을 "false"로 설정하여 비활성화할 수 있습니다.

필드	설명
ssm_maintenance_window	<p>AWS Systems Manager 유지 관리 기간을 이 일정에 대한 추가 실행 기간으로 추가할지 여부를 선택합니다. 예약된 EC2 인스턴스와 동일한 계정/리전의 기간 이름과 일치하는 StringSet의 유지 관리 기간 이름을 허용합니다.</p> <p>참고:이 기능은 EC2 인스턴스에만 적용됩니다.</p>
stop_new_instances	<p>실행 기간 외에 실행 중인 인스턴스에 처음으로 태그를 지정할 때 중지할지 여부를 선택합니다. 기본적으로 이 필드는 true로 설정되어 있습니다.</p>
timezone	<p>일정에서 사용할 시간대입니다. 시간대를 지정하지 않으면 기본 시간대(UTC)가 사용됩니다. 허용 가능한 시간대 값 목록은 <a href="#">List of tz database time zones</a>의 TZ 열을 참조하세요.</p>
use_metrics	<p>일정 수준에서 CloudWatch 지표를 활성화할지 여부를 선택합니다. 이 필드는 배포 시 지정한 CloudWatch 지표 설정을 덮어씁니다.</p> <p>참고:이 기능을 활성화하면 일정 또는 예약된 서비스당 월 0.90 USD의 요금이 발생합니다.</p>

## 기간 참조

기간에는 인스턴스가 실행될 특정 시간, 일 및 월을 설정할 수 있는 조건이 포함됩니다. 기간은 여러 조건을 포함할 수 있지만, AWS 인스턴스 스케줄러가 적절한 시작 또는 중지 작업을 적용하려면 모든 조건이 true여야 합니다.

## 시작 및 중지 시간

begintime 및 endtime 필드는 AWS 인스턴스 스케줄러가 인스턴스를 시작 및 중지하는 시기를 정의합니다. 시작 시간만 지정하는 경우 인스턴스를 수동으로 중지해야 합니다. [weekdays](#) 필드에 값을 지정하는 경우 솔루션은 해당 값을 사용하여 인스턴스를 중지할 시기를 결정합니다. 예를 들어,

endtime을 설정하지 않고 begintime을 오전 9시로 지정하고 weekdays 값을 월요일부터 금요일까지로 예약한 경우 금요일 오후 11시 59분에 인스턴스가 중지됩니다.

마찬가지로 중지 시간만 지정하는 경우 인스턴스를 수동으로 시작해야 합니다. 두 개의 시간 중 하나를 지정하지 않은 경우 이 솔루션은 해당 주의 요일, 해당 월의 일 또는 월 규칙을 사용하여 각 날짜의 시작/종료 시 인스턴스를 적절하게 시작하고 중지합니다.

기간의 begintime 및 endtime 값은 일정에 지정된 시간대에 포함되어야 합니다. 일정에 시간대를 지정하지 않으면 솔루션을 시작할 때 지정된 시간대가 솔루션에 사용됩니다.

일정에 여러 기간이 포함된 경우 항상 기간에 begintime과 endtime을 모두 지정하는 것이 좋습니다.

지정된 시작 시간 전에 인스턴스를 시작하는 경우 실행 기간이 끝날 때까지 인스턴스가 실행됩니다. 예를 들어, 사용자는 매일 오전 9시에 인스턴스를 시작하고 오후 5시에 해당 인스턴스를 중지하는 기간을 정의할 수 있습니다.

오전 9시~오후 5시에 예약된 시작 및 중지



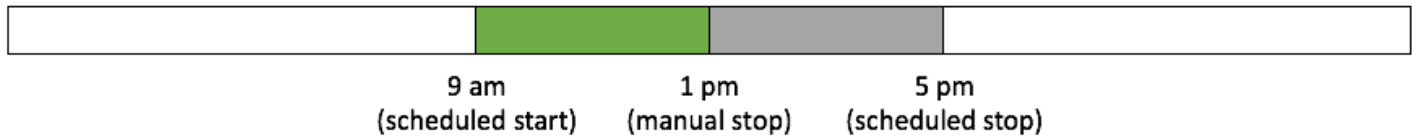
오전 5시에 해당 인스턴스를 수동으로 시작하는 경우 솔루션은 오후 5시에 인스턴스를 중지합니다. [실행 중인 필드 유지](#)를 사용하는 경우 솔루션은 오후 5시에 인스턴스를 중지하지 않습니다.

오전 5시에 예약된 중지



지정된 중지 시간 전에 인스턴스를 중지하면 다음 실행 기간이 시작될 때까지 인스턴스가 실행되지 않습니다. 위에 설명한 예시를 이어서 설명하자면, 사용자가 수요일 오후 1시에 인스턴스를 중지하면 솔루션은 목요일 오전 9시까지 인스턴스를 시작하지 않습니다.

예약된 시작 시간이 오전 9시, 수동 중지 시간이 오후 1시, 예약된 중지 시간이 오후 5시로 표시된 타임라인



## 인접 기간

일정에 인접한 두 개의 실행 기간이 포함된 경우 솔루션은 인스턴스 실행을 중지하지 않습니다. 예를 들어, 일정에서 한 기간의 `endtime`이 오후 11시 59분이고 다른 기간의 `begintime`이 다음 날 자정인 경우 인스턴스를 중지하는 `weekdays`, `monthdays` 또는 `months` 규칙이 없다면 솔루션은 인스턴스 실행을 중지하지 않습니다.

월요일 오전 9시부터 금요일 오후 5시까지 인스턴스를 실행하는 일정을 구현하기 위해서는 솔루션에 세 개의 기간이 필요합니다. 첫 번째 기간은 월요일 오전 9시부터 오후 11시 59분까지 해당 인스턴스를 실행합니다. 두 번째 기간은 화요일 자정부터 목요일 오후 11시 59분까지 인스턴스를 실행합니다. 세 번째 기간은 금요일 자정부터 금요일 오후 5시까지 인스턴스를 실행합니다. 자세한 내용은 [샘플 일정을 참조](#)하세요.

## 요일

`weekdays` 필드는 인스턴스가 주중 어느 요일에 실행될지를 정의합니다. 요일 목록, 요일 범위, 한 달 중 해당 요일이 n번째로 오는 시점 또는 한 달 중 해당 요일이 마지막으로 오는 시점으로 지정할 수 있습니다. 이 솔루션은 축약된 일 이름(Mon)과 숫자(0)를 지원합니다.

## 해당 월의 일

`monthdays` 필드는 인스턴스가 해당 월 중 어느 요일에 실행될지를 정의합니다. 일 목록, 일 범위, 해당 월 중 n일마다, 해당 월 중 마지막 일 또는 특정 날짜에 가장 가까운 평일을 지정할 수 있습니다.

## 개월

`months` 필드는 인스턴스가 실행될 월을 정의합니다. 월 목록, 월 범위 또는 n개월마다로 지정할 수 있습니다. 이 솔루션은 축약된 월 이름(Jan) 및 숫자(1)를 지원합니다.

## 기간 정의

Amazon DynamoDB의 AWS 인스턴스 스케줄러 구성 테이블에는 기간 정의가 포함되어 있습니다. 기간 정의에는 다음과 같은 필드가 포함될 수 있습니다. 일부 필드는 [Cron 비표준 문자](#)를 지원합니다.

**⚠ Important**

begintime, endtime, weekdays, months 또는 monthdays 중 하나 이상을 지정해야 합니다.

필드	설명
begintime	인스턴스가 시작할 HH:MM 형식의 시간입니다.
description	기간에 대한 설명입니다(선택 사항).
endtime	인스턴스가 중지될 HH:MM 형식의 시간입니다.
months	<p>인스턴스가 실행되는 동안 쉼표로 구분된 월 목록 또는 하이픈으로 표시한 월 범위를 입력합니다. 예를 들어, jan, feb, mar 또는 1, 2, 3을 입력하여 해당 월에 인스턴스를 실행할 수 있습니다. 아니면 jan-mar 또는 1-3을 입력할 수 있습니다.</p> <p>n개월마다 또는 범위 내에서 n개월마다 실행하도록 인스턴스를 예약할 수도 있습니다. 예를 들어, 1월부터 3개월마다 인스턴스를 실행하려면 Jan/3 또는 1/3을 입력하면 됩니다. Jan-Jul/2 를 입력하면 1월부터 7월까지 격월로 실행합니다.</p>
monthdays	<p>인스턴스가 실행되는 동안 쉼표로 구분된 해당 월의 일 목록 또는 하이픈으로 표시한 일 범위를 입력합니다. 예를 들어, 1, 2, 3 또는 1-3을 입력하여 해당 월의 첫 3일 동안 인스턴스를 실행할 수 있습니다. 범위를 여러 개 입력할 수도 있습니다. 예를 들어, 1-3, 7-9를 입력하여 1일부터 3일까지 그리고 7일부터 9일까지 인스턴스를 실행할 수 있습니다.</p> <p>해당 월 중 n개월마다 또는 범위 내에서 해당 월 중 n개월마다 실행하도록 인스턴스를 예약할 수</p>

필드	설명
	<p>도 있습니다. 예를 들어, 1/7을 입력하여 1일부터 시작해 7일마다 인스턴스를 실행할 수 있습니다. 1-15/2를 입력하여 1일부터 15일까지 격일로 인스턴스를 실행할 수 있습니다.</p> <p>L을 입력하여 해당 월의 마지막 날에 인스턴스를 실행할 수 있습니다. 지정된 날짜까지 가장 가까운 평일에 인스턴스를 실행하려면 날짜와 W를 입력합니다. 예를 들어, 15W를 입력하여 15일에 가장 가까운 평일에 인스턴스를 실행할 수 있습니다.</p>
name	<p>기간을 식별하는 데 사용되는 이름입니다. 이 이름은 고유해야 하며 영숫자, 하이픈(-) 및 밑줄(_)만 포함해야 합니다.</p>
weekdays	<p>인스턴스가 실행되는 동안 심표로 구분된 해당 주의 일 목록 또는 주의 일 범위를 입력합니다. 예를 들어, 0, 1, 2 또는 0-2를 입력하여 월요일부터 수요일까지 인스턴스를 실행할 수 있습니다. 범위를 여러 개 입력할 수도 있습니다. 예를 들어, 0-2, 4-6을 입력하여 목요일을 제외하고 매일 인스턴스를 실행할 수 있습니다.</p> <p>또한 해당 월의 평일 중 n번째로 오는 날마다 인스턴스가 실행되도록 예약할 수 있습니다. 예를 들어, Mon#1 또는 0#1을 입력하여 해당 월의 첫 번째 월요일에 인스턴스를 실행할 수 있습니다.</p> <p>일과 L을 입력하여 해당 월의 마지막 평일이 오는 시점에 인스턴스를 실행합니다. 예를 들어, friL 또는 4L을 입력하여 매월 마지막 금요일에 인스턴스를 실행할 수 있습니다.</p>

기간에 여러 조건이 포함된 경우 AWS 인스턴스 스케줄러가 적절한 작업을 적용하기 위해서는 모든 조건이 true여야 합니다. 예를 들어, 값이 Mon#1인 weekdays 필드와 값이 Jan/3인 months 필드가 포함된 기간은 분기의 첫 번째 월요일에 작업을 적용합니다.

## 샘플 일정

AWS 인스턴스 스케줄러를 사용하면 Amazon Elastic Compute Cloud(Amazon EC2) 및 Amazon Relational Database Service(Amazon RDS) 인스턴스의 자동으로 시작 및 중지할 수 있습니다. 다음 섹션에서는 여러 일반적인 사용 사례에 맞게 조정할 수 있는 몇 가지 예시 일정을 제공합니다.

### 오전 9시~오후 5시 표준 근무 시간

이 일정은 런던에서 평일 오전 9시~오후 5시에 인스턴스를 실행하는 방법을 보여줍니다.

#### 기간

이 기간은 오전 9시에 인스턴스를 시작하고 평일(월~금) 오후 5시에 인스턴스를 중지합니다.

Field	유형	값
begintime	String	09:00
endtime	String	16:59
name	String	weekdays-9-5
weekdays	StringSet	mon-fri

#### 일정

해당 일정 이름은 인스턴스에 적용해야 하는 태그 값과 사용할 시간대를 제공합니다.

Field	유형	값
name	String	london-working-hours
periods	StringSet	weekdays-9-5

Field	유형	값
timezone	String	Europe/London

## 인스턴스 태그

인스턴스에 이 일정을 적용하려면 인스턴스에 `Schedule=london-working-hours` 태그를 추가해야 합니다. AWS CloudFormation 인스턴스 스케줄러 태그 이름 파라미터에서 기본 태그 이름을 변경하면 태그가 바뀝니다. 예를 들어, 태그 이름으로 `Sked`를 입력한 경우 태그는 `Sked=london-working-hours`가 됩니다. 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 [리소스 태그 지정](#)을 참조하세요.

## 스케줄러 CLI

[인스턴스 스케줄러 CLI](#)를 사용하여 위 일정을 구성하려면 다음과 같은 명령을 사용합니다.

```
scheduler-cli create-period --stack <stackname> --name weekdays-9-5 --weekdays mon-fri
--begintime 9:00 --endtime 16:59

scheduler-cli create-schedule --stack <stackname> --name london-working-hours --periods
weekdays-9-5 --timezone Europe/London

Europe/London
```

## 사용자 지정 리소스

다음 CloudFormation 템플릿은 [일정 사용자 지정 리소스](#)를 사용하여 위의 일정을 생성합니다.

이 템플릿을 배포하려면 [이전에 배포한 인스턴스 스케줄러 허브 스택](#)을 선택한 다음 출력을 선택하여 AWS CloudFormation 콘솔에서 찾을 수 있는 `ServiceInstanceScheduleServiceToken` ARN을 제공해야 합니다.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
```

```

Resources:
  LondonWorkingWeek:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: london-working-hours
      Description: run instances from 9am to 5pm in London on weekdays
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: Europe/London
      Periods:
        - Description: 9am to 5pm on weekdays
          BeginTime: '09:00'
          EndTime: '16:59'
          WeekDays: mon-fri

```

## 오후 5시 이후 인스턴스 중지

인스턴스는 하루 중 언제든지 자유롭게 시작할 수 있으며, 이 일정은 중지 명령이 매일 오후 5시에 자동으로 전송되도록 합니다.

### 기간

이 기간은 매일 오후 5시에 인스턴스를 중지합니다.

Field	유형	값
endtime	String	16:59
name	String	stop-at-5

### 일정

해당 일정 이름은 인스턴스에 적용해야 하는 태그 값과 사용할 시간대를 제공합니다.

Field		값
name	String	stop-at-5-new-york
periods	StringSet	stop-at-5

Field		값
timezone	String	America/New York

## 인스턴스 태그

인스턴스에 이 일정을 적용하려면 인스턴스에 `Schedule=stop-at-5-new-york` 태그를 추가해야 합니다. AWS CloudFormation 인스턴스 스케줄러 태그 이름 파라미터에서 기본 태그 이름을 변경했다면 태그가 바뀝니다. 예를 들어, 태그 이름으로 `Sked`를 입력한 경우 태그는 `Sked=stop-at-5-new-york`가 됩니다. 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 [리소스 태그 지정](#)을 참조하세요.

## 스케줄러 CLI

[인스턴스 스케줄러 CLI](#)를 사용하여 위 일정을 구성하려면 다음과 같은 명령을 사용합니다.

```
scheduler-cli create-period --stack <stackname> --name stop-at-5 --endtime 16:59

scheduler-cli create-schedule --stack <stackname> --name stop-at-5-new-york --periods
stop-at-5 --timezone America/New_York
```

## 사용자 지정 리소스

다음 CloudFormation 템플릿은 [일정 사용자 지정 리소스](#)를 사용하여 위의 일정을 생성합니다.

이 템플릿을 배포하려면 [이전에 배포한 인스턴스 스케줄러 허브 스택](#)을 클릭하고 출력을 선택하여 AWS CloudFormation 콘솔에서 찾을 수 있는 `ServiceInstanceScheduleServiceToken` ARN을 제공해야 합니다.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  StopAfter5:
    Type: 'Custom::ServiceInstanceSchedule'
```

```

Properties:
  NoStackPrefix: 'True'
  Name: stop-at-5-new-york
  Description: stop instances at 5pm ET every day
  ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
  Timezone: America/New_York
  Periods:
  - Description: stop at 5pm
    EndTime: '16:59'

```

## 주말 동안 인스턴스 중지

이 일정은 월요일 동부 표준시 기준 오전 9시부터 금요일 오후 5시까지 인스턴스를 실행하는 방법을 보여줍니다. 월요일과 금요일은 하루 전체에 대하여 실행하는 것이 아니기 때문에 이 일정에는 월요일, 화요일~목요일, 금요일의 세 기간이 포함됩니다.

### 기간

첫 번째 기간은 월요일 오전 9시에 태그가 지정된 인스턴스를 시작하고 자정에 중지됩니다. 이 기간에는 다음 필드와 값이 포함됩니다.

Field	유형	값
begintime	String	09:00
endtime	String	23:59
name	String	mon-start-9am
weekdays	StringSet	mon

두 번째 기간은 화요일부터 목요일까지 하루 전체에 대해 태그가 지정된 인스턴스를 실행합니다. 이 기간에는 다음 필드와 값이 포함됩니다.

Field		값
name	String	tue-thu-full-day
weekdays	StringSet	tue-thu

세 번째 기간은 금요일 오후 5시에 태그가 지정된 인스턴스를 중지합니다. 이 기간에는 다음 필드와 값이 포함됩니다.

Field		값
begintime	String	00:00
endtime	String	16:59
name	String	fri-stop-5pm
weekdays	StringSet	fri

## 일정

해당 일정은 세 기간을 태그가 지정된 인스턴스의 일정에 결합합니다. 일정에는 다음과 같은 필드와 값이 포함됩니다.

Field		값
name	String	mon-9am-fri-5pm
periods	StringSet	mon-start-9am,tue-thu-full-day,fri-stop-5pm
timezone	String	America/New_York

## 인스턴스 태그

인스턴스에 이 일정을 적용하려면 인스턴스에 `Schedule=mon-9am-fri-5pm` 태그를 추가해야 합니다. AWS CloudFormation 인스턴스 스케줄러 태그 이름 파라미터에서 기본 태그 이름을 변경했다면 태그가 바뀝니다. 예를 들어, 태그 이름으로 Sked를 입력한 경우 태그는 `Sked=mon-9am-fri-5pm`이 됩니다. 자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 [리소스 태그 지정](#)을 참조하세요.

## 스케줄러 CLI

[인스턴스 스케줄러 CLI](#)를 사용하여 위 일정을 구성하려면 다음과 같은 명령을 사용합니다.

```

scheduler-cli create-period --stack <stackname> --name
mon-start-9am --weekdays mon --begintime 9:00 --endtime 23:59
scheduler-cli create-period --stack <stackname> --name
tue-thu-full-day --weekdays tue-thu
scheduler-cli create-period --stack <stackname> --namefri-stop-5pm --weekdays fri --
begintime 0:00 --endtime 17:00

scheduler-cli create-schedule --stack <stackname> --name
mon-9am-fri-5pm --periods
mon-start-9am,tue-thu-full-day,fri-stop-5pm -timezone
America/New_York

```

## 사용자 지정 리소스

다음 CloudFormation 템플릿은 [일정 사용자 지정 리소스](#)를 사용하여 위의 일정을 생성합니다.

이 템플릿을 배포하려면 [이전에 배포한 인스턴스 스케줄러 허브 스택](#)을 선택한 다음 출력을 선택하여 AWS CloudFormation 콘솔에서 찾을 수 있는 ServiceInstanceScheduleServiceToken ARN을 제공해야 합니다.

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  StopOnWeekends:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      NoStackPrefix: 'True'
      Name: mon-9am-fri-5pm
      Description: start instances at 9am on monday and stop them at 5pm on friday
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN
      Timezone: America/New_York
      Periods:
        - Description: 9am monday start
          BeginTime: '09:00'
          EndTime: '23:59'
          WeekDays: mon
        - Description: all day tuesday-thursday
          WeekDays: tue-thu

```

```
- Description: 5pm friday stop
  BeginTime: '00:00'
  EndTime: '16:59'
  WeekDays: fri
```

## 솔루션 리소스

다음 리소스는 AWS 인스턴스 스케줄러 스택의 일부로 생성됩니다.

리소스 이름	유형	설명
기본	AWS::Lambda::Function	AWS 인스턴스 스케줄러의 Lambda 함수입니다.
스케줄러 구성 헬퍼	Custom::ServiceSetup	전역 구성 설정을 Amazon DynamoDB에 저장합니다.
스케줄러 간접 호출 권한	AWS::Lambda::Permission	Amazon CloudWatch 이벤트가 인스턴스 스케줄러의 AWS Lambda 함수를 간접 호출하도록 허용합니다.
스케줄러 로그	AWS::Logs::LogGroup	인스턴스 스케줄러에 대한 CloudWatch 로그 그룹입니다.
스케줄러 정책	AWS::IAM::Policy	스케줄러가 시작 및 중지 작업을 수행하고, Amazon EC2 인스턴스 속성을 변경하며, 태그를 설정하고, 스케줄러 리소스에 액세스할 수 있도록 허용하는 정책입니다.
스케줄러 규칙	AWS::Events::Rule	스케줄러의 Lambda 함수를 간접 호출하는 Amazon EventBridge 이벤트 규칙입니다.
구성 지표 이벤트 규칙	AWS::Events::Rule	구성 설명 익명화된 지표 함수를 주기적으로 간접 호출하는

리소스 이름	유형	설명
		Amazon EventBridge 이벤트 규칙입니다. 익명화된 지표가 비활성화되면 비활성화됩니다.
상태 테이블	AWS::DynamamoDB::Table	원하는 인스턴스의 마지막 상태를 저장하는 DynamoDB 테이블입니다.
구성 테이블	AWS::DynamamoDB::Table	전역 구성, 일정 및 기간 데이터를 저장하는 DynamoDB 테이블입니다.
인스턴스 스케줄러 SNS 주제	AWS::SNS::Topic	구독한 이메일 주소로 경고 및 오류 메시지를 보냅니다.

## 스케줄러 CLI

AWS 인스턴스 스케줄러 명령줄 인터페이스(CLI)를 사용하면 일정 및 기간을 구성하고 지정된 일정에 대한 비용 절감을 추정할 수 있습니다.

### 사전 조건

이 솔루션의 CLI에는 Python 3.8 이상과 최신 버전의 boto3가 필요합니다.

### 자격 증명

스케줄러 CLI를 사용하기 위해서는 AWS CLI에 대한 자격 증명이 있어야 합니다. 자세한 내용은 AWS CLI 사용 설명서의 [구성 및 자격 증명 파일 설정](#)을 참조하세요.

다음과 같은 자격 증명이 있어야 합니다.

- `lambda:InvokeFunction` - 스케줄러 스택에서 InstanceSchedulerMain 함수를 간접 호출하고 명령줄에서 스케줄러 구성 데이터베이스의 일정 및 기간 정보를 업데이트
- `cloudformation:DescribeStackResource` - 스택에서 AWS Lambda 함수의 물리적 리소스 ID를 검색하여 CLI 요청을 처리

스케줄러 CLI 및 응답의 요청은 AdminCliRequestHandler-yyyyymmdd 로그 스트림에 로깅됩니다.

### Note

프로필 이름 인수를 사용하여 프로필을 지정하는 경우 지정하는 프로필에 이러한 권한이 있어야 합니다. 프로필 이름 인수에 대한 자세한 내용은 [공통 인수](#)를 참조하세요.

## 스케줄러 CLI 설치

1. 스케줄러 CLI 패키지(instance\_scheduler\_cli.zip)를 [다운로드](#)하여 컴퓨터의 디렉터리에 배치합니다.

### Important

파일을 자체 디렉터리에 배치하지 않고 해당 디렉터리에서 설치하면 설치에 실패합니다.

2. zip 아카이브의 압축을 자체 디렉터리(instance\_scheduler\_cli)로 풉니다.
3. 압축을 푼 CLI 패키지를 배치한 디렉터리와 동일한 디렉터리에서 Scheduler-cli를 환경에 다음 코드를 사용하여 설치합니다.

### Note

Scheduler-CLI에는 Python 3.8 이상과 최신 버전의 pip 및 boto3가 필요합니다. 로컬 시스템에 이러한 모든 항목이 설치되어 있지 않은 경우 Scheduler-CLI를 설치하기 전에 [pip 공식 설명서](#)에서 설치 지침을 참조하세요.

```
pip install --no-index --find-links=instance_scheduler_cli instance_scheduler_cli
```

4. 다음 코드를 사용하여 설치가 성공했는지 확인합니다.

```
scheduler-cli --help
```

**Note**

원하는 경우 위와 동일한 프로세스를 사용하여 [CLI의 sdist](#)를 다운로드하고 설치할 수 있습니다.

## 명령 구조

스케줄러 CLI는 명령줄에서 멀티파트 구조를 사용합니다. 다음 부분에서는 스케줄러 CLI python 스크립트를 지정합니다. 스케줄러 CLI에는 기간 및 일정에 대해 수행할 작업을 지정하는 명령이 있습니다. 작업의 특정 인수는 명령줄에 원하는 순서로 지정할 수 있습니다.

```
scheduler-cli <command> <arguments>
```

## 공통 인수

스케줄러 CLI는 모든 명령이 사용할 수 있는 다음과 같은 인수를 지원합니다.

인수	설명
<code>--stack [replaceable]&lt;stackname&gt;</code>	스케줄러의 이름입니다.  중요: 이 인수는 모든 명령에 필요합니다.
<code>--region [replaceable]&lt;regionname&gt;</code>	스케줄러 스택이 배포되는 리전의 이름입니다.  참고: 솔루션 스택과 동일한 리전에 기본 구성 및 자격 증명 파일이 설치되지 않은 경우 이 인수를 사용해야 합니다.
<code>--profile-name [replaceable]&lt;profilename&gt;</code>	명령을 실행하는 데 사용할 프로필의 이름입니다. 프로필 이름이 지정되어 있지 않으면 기본 프로필이 사용됩니다.
<code>--query</code>	명령 출력을 제어하는 JMESPath 표현식입니다. 출력 제어에 대한 자세한 내용은 AWS CLI 사용 설명서의 <a href="#">AWS CLI에서 명령 출력 제어</a> 를 참조하세요.

인수	설명
--help	스케줄러 CLI에 유효한 명령과 인수를 표시합니다. 특정 명령과 함께 사용하면 해당 명령에 유효한 하위 명령과 인수가 표시됩니다.
--version	스케줄러 CLI의 버전 번호를 표시합니다.

## 가용 명령

- [create-period](#)
- [create-schedule](#)
- [delete-period](#)
- [delete-schedule](#)
- [describe-periods](#)
- [describe-schedules](#)
- [describe-schedule-usage](#)
- [update-period](#)
- [update-schedule](#)
- [help](#)

## create-period

### 설명

기간을 생성합니다. 기간에는 begintime, endtime, weekdays, months 또는 monthdays 항목 중 하나 이상이 포함되어야 합니다.

### 인수

--name

- 해당 기간의 이름

유형: 문자열

필수 항목 여부: 예

#### --description

- 규칙에 대한 설명입니다.

유형: 문자열

필수 항목 여부: 아니요

#### --begintime

- 실행 기간이 시작되는 시간입니다. begintime 및 endtime을 지정하지 않으면 실행 기간은 00:00~23:59으로 설정됩니다.

유형: 문자열

제약 조건: H:MM 또는 HH:MM 형식

필수 여부: 아니요

#### --endtime

- 실행 기간이 중지되는 시간입니다. begintime 및 endtime을 지정하지 않으면 실행 기간은 00:00~23:59으로 설정됩니다.

유형: 문자열

제약 조건: H:MM 또는 HH:MM 형식

필수 여부: 아니요

#### --weekdays

- 해당 기간의 요일

유형: 문자열

제약 조건: 축약된 일 이름(mon) 또는 숫자(0)의 심표로 구분된 목록 형식 -을 사용하여 범위를 지정합니다. /를 사용하여 해당 주 중 n번째 요일마다로 지정합니다.

필수 여부: 아니요

#### --months

- 해당 기간의 월

유형: 문자열

제약 조건: 축약된 월 이름(jan) 또는 숫자(1)의 쉼표로 구분된 목록 형식 -을 사용하여 범위를 지정합니다. /를 사용하여 n번째 월마다로 지정합니다.

필수 여부: 아니요

## --monthdays

- 기간 중 해당 월의 일

유형: 문자열

제약 조건: 축약된 월 이름(jan) 또는 숫자(1)의 쉼표로 구분된 목록 형식 -을 사용하여 범위를 지정합니다. /를 사용하여 해당 월의 n번째 일마다로 지정합니다.

필수 여부: 아니요

## 예제

```
$ scheduler-cli create-period --name "weekdays" --begintime 09:00 --endtime 18:00 --
weekdays mon-fri --stack Scheduler
{
  "Period": {
    "Name": "weekdays",
    "Endtime": "18:00",
    "Type": "period",
    "Begintime": "09:00",
    "Weekdays": [
      "mon-fri"
    ]
  }
}
```

## create-schedule

### 설명

일정을 생성합니다.

## 인수

### --name

- 일정의 이름입니다.

유형: 문자열

필수 항목 여부: 예

### --description

- 일정에 대한 설명입니다.

유형: 문자열

필수 항목 여부: 아니요

### --enforced

- 인스턴스에 대해 예약된 상태를 적용합니다.

필수 여부: 아니요

### --use-metrics

- Amazon CloudWatch의 지표를 수집합니다.

필수 여부: 아니요

### --periods

- 일정의 실행 기간 목록입니다. 여러 기간을 지정하는 경우 기간 중 하나가 true인 것으로 평가되면 솔루션이 인스턴스를 시작합니다.

유형: 문자열

제약 조건: 쉼표로 구분된 기간 목록 형식 <period-name>@[replaceable]<instance type>을 사용하여 기간의 인스턴스 유형을 지정합니다. 예를 들어 weekdayset2.large입니다.

필수 항목 여부: 예

### --retain-running

- 기간이 시작되기 전에 인스턴스가 수동으로 시작된 경우 실행 기간이 끝날 때 솔루션에 의해 인스턴스가 중지되는 것을 방지합니다.

필수 여부: 아니요

**--ssm-maintenance-window**

- AWS Systems Manager 유지 관리 기간을 Amazon EC2 인스턴스 일정에 실행 기간으로 추가합니다.

유형: 문자열

필수 항목 여부: 아니요

**--do-not-stop-new-instances**

- 실행 기간 외에 실행 중인 경우 인스턴스에 처음으로 태그를 지정할 때는 인스턴스를 중지하지 마십시오.

필수 여부: 아니요

**--timezone**

- 일정에서 사용할 시간대입니다.

유형: 문자열 배열

필수 여부: 아니요(이 인수를 사용하지 않으면 기본 솔루션 스택의 기본 시간대가 사용됨)

**--use-maintenance-window**

- Amazon RDS 인스턴스 일정의 실행 기간으로 Amazon RDS 유지 관리 기간을 추가하거나 Amazon EC2 인스턴스 일정의 실행 기간으로 AWS Systems Manager 유지 관리 기간을 추가합니다.

유형: true/false

필수 여부: 아니요(기본값은 true임)

**예제**

```
$ scheduler-cli create-schedule --name LondonOfficeHours --periods weekdays,weekends --
timezone Europe/London --stack Scheduler
{
  "Schedule": {
    "Enforced": false,
    "Name": "LondonOfficeHours",
    "StopNewInstances": true,
    "Periods": [
      "weekends",
      "weekdays"
    ]
  }
}
```

```

    ],
    "Timezone": "Europe/London",
    "Type": "schedule"
  }
}

```

## delete-period

--name

- 해당 기간의 이름입니다.

유형: 문자열

필수 항목 여부: 예

### Important

이 기간이 기존 일정에 사용되는 경우 삭제하기 전에 해당 일정에서 해당 기간을 제거해야 합니다.

## 예제

```

$ scheduler-cli delete-period --name weekdays --stack Scheduler
{
  "Period": "weekdays"
}

```

## delete-schedule

### 설명

기존 일정을 삭제합니다.

### 인수

--name

- 해당 일정의 이름입니다.

유형: 문자열

필수 항목 여부: 예

## 예제

```
$ scheduler-cli delete-schedule --name LondonOfficeHours --stack Scheduler
{
  "Schedule": "LondonOfficeHours"
}
```

## describe-periods

### 설명

인스턴스 스케줄러 스택에 대해 구성된 기간을 나열합니다.

### 인수

--name

- 설명할 특정 기간의 이름입니다.

유형: 문자열

필수 항목 여부: 아니요

## 예제

```
$ scheduler-cli describe-periods --stack Scheduler
{
  "Periods": [
    {
      "Name": "first-monday-in-quarter",
      "Months": [
        "jan/3"
      ],
      "Type": "period",
      "Weekdays": [
        "mon#1"
      ],
      "Description": "Every first Monday of each quarter"
    }
  ]
}
```

```
    },
    {
      "Description": "Office hours",
      "Weekdays": [
        "mon-fri"
      ],
      "Begintime": "09:00",
      "Endtime": "17:00",
      "Type": "period",
      "Name": "office-hours"
    },
    {
      "Name": "weekdays",
      "Endtime": "18:00",
      "Type": "period",
      "Weekdays": [
        "mon-fri"
      ],
      "Begintime": "09:00"
    },
    {
      "Name": "weekends",
      "Type": "period",
      "Weekdays": [
        "sat-sun"
      ],
      "Description": "Days in weekend"
    }
  ]
}
```

## describe-schedules

### 설명

인스턴스 스케줄러 스택에 대해 구성된 일정을 나열합니다.

### 인수

**--name**

- 설명할 특정 일정의 이름입니다.

유형: 문자열

필수 항목 여부: 아니요

## 예제

```
$ scheduler-cli describe-schedules --stack Scheduler

{
  "Schedules": [
    {
      "OverrideStatus": "running",
      "Type": "schedule",
      "Name": "Running",
      "UseMetrics": false
    },
    {
      "Timezone": "UTC",
      "Type": "schedule",
      "Periods": [
        "working-days@t2.micro",
        "weekends@t2.nano"
      ],
      "Name": "scale-up-down"
    },
    {
      "Timezone": "US/Pacific",
      "Type": "schedule",
      "Periods": [
        "office-hours"
      ],
      "Name": "seattle-office-hours"
    },
    {
      "OverrideStatus": "stopped",
      "Type": "schedule",
      "Name": "stopped",
      "UseMetrics": true
    }
  ]
}
```

## describe-schedule-usage

### 설명

일정 내에서 실행되는 모든 기간을 나열하고 인스턴스의 청구 시간을 계산합니다. 이 명령을 사용하여 일정을 시뮬레이션해 잠재적 절감액과 일정을 생성하거나 업데이트한 후의 실행 기간을 계산합니다.

### 인수

#### --name

- 해당 일정의 이름입니다.

유형: 문자열

필수 항목 여부: 예

#### --startdate

- 계산에 사용되는 기간의 시작 날짜입니다. 기본 날짜는 현재 날짜입니다.

유형: 문자열

필수 항목 여부: 아니요

#### --enddate

- 계산에 사용되는 기간의 종료 날짜입니다. 기본 날짜는 현재 날짜입니다.

유형: 문자열

필수 항목 여부: 아니요

### 예제

```
$ scheduler-cli describe-schedule-usage --stack InstanceScheduler --name seattle-office-hours
{
  "Usage": {
    "2017-12-04": {
      "BillingHours": 8,
      "RunningPeriods": {
        "Office-hours": {
          "Begin": "12/04/17 09:00:00",
          "End": "12/04/17 17:00:00",
```

```
        "BillingHours": 8,  
        "BillingSeconds": 28800  
    },  
    },  
    "BillingSeconds": 28800  
},  
"Schedule": "seattle-office-hours"
```

## update-period

### 설명

기존 프로필을 업데이트합니다.

### 인수

update-period 명령은 create-period 명령과 동일한 인수를 지원합니다. 인수에 대한 자세한 내용은 [기간 생성 명령](#)을 참조하세요.

#### Important

인수를 지정하지 않으면 해당 인수가 기간에서 제거됩니다.

## update-schedule

### 설명

기존 일정을 업데이트합니다.

### 인수

update-schedule 명령은 create-schedule 명령과 동일한 인수를 지원합니다. 인수에 대한 자세한 내용은 [create-schedule 명령](#)을 참조하세요.

#### Important

인수를 지정하지 않으면 해당 인수가 일정에서 제거됩니다.

# help

## 설명

스케줄러 CLI에 대한 유효한 명령과 인수 목록을 표시합니다.

## 예제

```
$ scheduler-cli --help
usage: scheduler-cli [-h] [--version]
                    {create-period,create-schedule,delete-period,delete-
schedule,describe-periods,describe-schedule-usage,describe-schedules,update-
period,update-schedule}
                    ...

optional arguments:
  -h, --help            show this help message and exit
  --version              show program's version number and exit

subcommands:
  Valid subcommands

  {create-period,create-schedule,delete-period,delete-schedule,describe-
periods,describe-schedule-usage,describe-schedules,update-period,update-schedule}

  create-period         Creates a period
  create-schedule       Creates a schedule
  delete-period         Deletes a period
  delete-schedule       Deletes a schedule
  describe-periods     Describes configured periods
  describe-schedule-usage
                        Calculates periods and billing hours in which
                        instances are running
  describe-schedules   Described configured schedules
  update-period         Updates a period
  update-schedule       Updates a schedule
```

특정 명령과 함께 사용하면 --help 인수가 해당 명령에 유효한 하위 명령과 인수를 표시합니다.

## 특정 명령 예시

```
$ scheduler-cli describe-schedules --help
```

```
usage: scheduler-cli describe-schedules [-h] [--name NAME] [--query QUERY]
                                         [--region REGION] --stack STACK

optional arguments:
  -h, --help                show this help message and exit
  --name NAME                Name of the schedule
  --query QUERY              JMESPath query to transform or filter the result
  --region REGION            Region in which the Instance Scheduler stack is
                             deployed
  --stack STACK, -s STACK   Name of the Instance Scheduler stack
```

## 전역 구성 설정 업데이트

AWS CloudFormation에 인스턴스 스케줄러의 Hub 템플릿을 처음 배포했을 때 여러 전역 구성 설정이 파라미터 입력으로 선택됩니다. 이러한 전역 구성 파라미터는 CloudFormation 콘솔 내에서 언제든지 업데이트할 수 있습니다.

인스턴스 스케줄러의 전역 구성을 업데이트하려면 허브 배포가 포함된 계정/리전에 로그인하고 AWS CloudFormation 콘솔로 이동합니다. 인스턴스 스케줄러 허브 스택을 찾고 업데이트 → 기존 템플릿 사용을 선택합니다. 변경하려는 전역 구성 파라미터를 업데이트한 후 다음 → 다음 → 제출을 선택하여 관련 솔루션 리소스의 CloudFormation 업데이트를 수행합니다.

## 코드형 인프라(IaC)를 사용하여 일정 관리

### Important

허브 스택 배포가 완료된 후 별도의 템플릿을 사용하여 일정을 배포합니다.

AWS 인스턴스 스케줄러는 AWS CloudFormation을 통해 일정을 구성하고 관리하는 데 사용할 수 있는 사용자 지정 리소스(ServiceInstanceSchedule)를 제공합니다. 사용자 지정 리소스는 Amazon DynamoDB의 인스턴스 스케줄러 구성 테이블과 동일한 데이터에 대해 PascalCase 키를 사용합니다 (예시는 아래 템플릿 참조). 일정 필드에 대한 자세한 내용은 [일정 정의](#)를 참조하세요. 기간 필드에 대한 자세한 내용은 [기간 정의](#)를 참조하세요.

사용자 지정 리소스를 사용하여 일정을 생성할 때 해당 일정의 이름은 기본적으로 사용자 지정 리소스의 논리적 리소스 이름입니다. 다른 이름을 지정하려면 사용자 지정 리소스의 이름 속성을 사용합니다.

또한 솔루션은 기본적으로 스택 이름을 일정 이름에 접두사로 추가합니다. 스택 이름을 접두사로 추가하지 않으려면 NoStackPrefix 속성을 사용합니다.

이름 및 NoStackPrefix 속성을 사용할 때는 고유한 일정 이름을 선택해야 합니다. 동일한 이름의 일정이 이미 있는 경우 리소스가 생성되거나 업데이트되지 않습니다.

laC를 사용하여 일정 관리를 시작하려면 다음 샘플 템플릿을 복사하여 붙여 넣고 일정을 원하는 대로 사용자 지정합니다. 파일을 .template 파일(예: my-schedules.template)로 저장한 다음 AWS CloudFormation을 사용하여 새 템플릿을 배포합니다. 완료된 일정 템플릿의 예는 [샘플 일정](#)을 참조하세요.

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  ServiceInstanceScheduleServiceTokenARN:
    Type: String
    Description: (Required) service token arn taken from InstanceScheduler outputs
Metadata:
  'AWS::CloudFormation::Designer': {}
Resources:
  SampleSchedule1:
    Type: 'Custom::ServiceInstanceSchedule'
    Properties:
      ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN #do not edit this line
      NoStackPrefix: 'False'
      Name: my-renamed-sample-schedule
      Description: a full sample template for creating cfn schedules showing all
possible values
      Timezone: America/New_York
      Enforced: 'True'
      Hibernate: 'True'
      RetainRunning: 'True'
      StopNewInstances: 'True'
      UseMaintenanceWindow: 'True'
      SsmMaintenanceWindow: 'my_window_name'
      Periods:
        - Description: run from 9-5 on the first 3 days of March
          BeginTime: '9:00'
          EndTime: '17:00'
          InstanceType: 't2.micro'
          MonthDays: '1-3'
          Months: '3'
        - Description: run from 2pm-5pm on the weekends
          BeginTime: '14:00'
```

```

EndTime: '17:00'
InstanceType: 't2.micro'
WeekDays: 'Sat-Sun'

```

#### SampleSchedule2:

```
Type: 'Custom::ServiceInstanceSchedule'
```

#### Properties:

```

ServiceToken: !Ref ServiceInstanceScheduleServiceTokenARN #do not edit this line
NoStackPrefix: 'True'
Description: a sample template for creating simple cfn schedules
Timezone: Europe/Amsterdam
Periods:
- Description: stop at 5pm every day
  EndTime: '17:00'

```

템플릿을 배포할 때 AWS 인스턴스 스케줄러를 배포할 수 있도록 ServiceTokenARN을 제공해야 합니다. 이 ARN은 배포된 인스턴스 스케줄러 스택으로 이동하여 출력을 선택하고 ServiceInstanceScheduleServiceToken.을 찾아 CloudFormation에서 찾을 수 있습니다.

#### Important

DynamoDB 콘솔 또는 스케줄러 CLI로 사용자 지정 리소스를 사용해 구성된 일정 및 기간을 삭제하거나 수정하지 마십시오. 이렇게 하면 스택에 저장된 파라미터와 테이블의 값 간에 충돌이 발생합니다. 또한 DynamoDB 콘솔 또는 스케줄러 CLI로 생성된 일정에 사용자 지정 리소스를 사용하여 구성된 기간을 사용하지 마십시오.

기본 인스턴스 스케줄러 스택을 삭제하기 전에 사용자 지정 리소스 스택에 기본 스택의 DynamoDB 테이블에 대한 종속성이 포함되므로 사용자 지정 리소스를 사용하여 생성된 일정 및 기간이 포함된 추가 스택을 모두 삭제해야 합니다.

구성 DynamoDB 테이블에서 사용자 지정 리소스로 구성된 일정 및 기간은 `configured_in_stack` 속성으로 식별할 수 있습니다. 속성에는 항목을 생성하는 데 사용된 스택의 Amazon 리소스 이름이 포함됩니다.

## EC2 용량 부족 오류 처리

용량 부족으로 인해 인스턴스 스케줄러가 인스턴스를 시작하지 못하는 경우 기본 동작은 시작 실패 이벤트를 실행하고( [EventBridge 이벤트](#) 참조) 다음 예약 간격에 다시 시도하는 것입니다. 또는 인스턴스

스케줄러를 구성하여 시작 작업을 다시 시도하기 전에 인스턴스 크기를 대체 인스턴스 유형으로 조정할 수 있습니다. 이 기능은 용량 제약 환경에서 인스턴스 가용성을 개선하는 데 도움이 됩니다.

## 구성

EC2 인스턴스에 대해 대체 인스턴스 유형을 활성화하려면 선호도(가장 선호되는 우선 순위)에 따라 심표로 구분된 인스턴스 유형 목록을 사용하여 인스턴스에 IS-PreferredInstanceTypes 태그를 추가합니다.

```
IS-PreferredInstanceTypes: t3.medium,t3.large,m5.large
```

## 작동 방식

대체 인스턴스 유형 목록은 기본 설정 순서대로 제공되며 첫 번째 유형이 가장 선호됩니다. 인스턴스 스케줄러가 EC2 인스턴스 시작을 시도하는 경우:

1. 인스턴스가 현재 가장 선호하는 크기가 아닌 경우는 시작하기 전에 가장 선호하는 크기로 크기를 조정하려고 시도합니다.
2. 시작 작업이 성공하면 더 이상 대체 작업을 시도하지 않습니다.
3. 용량 부족으로 인해 시작 작업이 실패하는 경우:
  - a. 목록의 다음 대체 인스턴스 유형으로 크기 조정 시도
  - b. 시작 작업 재시도
  - c. 여전히 실패할 경우는 다음 대체 유형을 시도합니다.
  - d. 성공하거나 모든 대체 항목이 소진될 때까지 계속됩니다.

## 요구 사항 및 제한 사항

인스턴스 호환성: 대체 인스턴스 유형은 인스턴스의 현재 구성(AMI, 서브넷, 보안 그룹 등)과 호환되어야 합니다. 자세한 내용은 Amazon EC2 사용 설명서 [의 인스턴스 유형 변경을](#) 참조하세요.

태그 형식: IS-PreferredInstanceTypes 태그 값은 유효한 EC2 인스턴스 유형의 심표로 구분된 목록이어야 합니다.

## 예제

원래 로 구성된 인스턴스의 경우 다음을 구성할 t3.small 수 있습니다.

```
Schedule: office-hours
```

```
IS-PreferredInstanceTypes: t3.small,t3.medium,t3.large,m5.large
```

용량 문제로 인해 t3.small 인스턴스가 시작되지 않는 경우 인스턴스 스케줄러는 인스턴스의 크기를 조정하여 로 시작한 t3.medium 다음 성공하거나 모든 옵션이 소진 t3.large m5.large 될 때까지 로 시작하려고 시도합니다.

## EC2 Auto Scaling 그룹 예약

AWS 인스턴스 스케줄러는 예약된 스케일링 작업을 사용하여 EC2 Auto Scaling 그룹(ASG)의 예약을 지원합니다. 이는 EC2/RDS 예약의 구현과 다르며, 이 섹션에서 자세히 설명합니다.

예약된 스케일링 작업에 대한 자세한 내용은 [Amazon EC2 Auto Scaling에 예약된 조정](#)을 참조하세요.

### ASG 예약 개요

ASG 예약은 [예약을 위한 인스턴스 태그 지정](#)에 설명된 대로 일정 태그를 적용하여 예약할 수 있습니다.

### ASG 실행/중지의 정의

Auto Scaling 그룹을 구성하면 사용자는 해당 ASG에 대한 최소 용량, 원하는 용량 및 최대 용량을 지정합니다. 인스턴스 스케줄러는 이러한 값을 ASG의 min-desired-max 또는 MDM이라고 합니다.

ASG의 실행 상태는 IS-MinDesiredMax 제어 태그를 사용하여 정의됩니다. 이 태그에는 원하는 MDM 값이 형식 min,desired,max(예: 1,3,5)으로 포함되어야 합니다.

ASG에 예약 태그가 처음 지정될 때 IS-MinDesiredMax 태그가 제공되지 않으면 태그 지정 시 ASG의 현재 크기에서 자동으로 생성됩니다.

모든 ASGs 대해 중지된 상태는 0-0-0의 MDM으로 정의됩니다.

### ASG 시작/중지 동작

인스턴스 스케줄러는 ASG를 시작하거나 중지할 때 ASG의 용량 설정을 수정합니다.

ASG 시작: 최소, 원하는 및 최대 용량을 IS-MinDesiredMax 태그에 정의된 값(또는 ASG에 처음 태그를 지정할 때에서 자동으로 생성된 값)으로 설정합니다.

ASG 중지: 최소, 원하는 및 최대 용량을 0-0-0으로 설정하여 ASG의 모든 인스턴스가 종료됩니다.

## 제한 사항

ASG 예약은 AWS 인스턴스 스케줄러 일정을 ASG 서비스와 호환되는 예약된 스케일링 규칙으로 변환하여 수행됩니다. 이 변환은 복잡한 cron 표현식을 사용하지 않는 간단한 단일 기간 일정에 가장 적합합니다.

ASG 예약에는 다음과 같은 일정 기능이 지원되지 않습니다.

- 적용 및 실행 유지와 같은 고급 일정 플래그
- N-번째 평일, 가장 가까운 평일 및 기간의 마지막 평일 표현식입니다.
- 바로 인접하거나 중첩되는 기간이 있는 다중 기간 일정
  - 다중 기간 일정에 대해 예약된 스케일링 작업을 구성할 때 AWS 인스턴스 스케줄러는 겹치거나 인접한 다른 기간으로 인해 일반적으로 해당 작업을 건너뛰더라도 ASG에 대한 시작/종지 작업을 위해 기간의 시작/종료를 직접 변환합니다.

## 솔루션 모니터링

### 로깅 및 알림

인스턴스 스케줄러는 CloudWatch Logs Insights 쿼리에 최적화된 구조화된 로깅을 사용합니다. 이 솔루션은 태그가 지정된 각 인스턴스에 대한 처리 정보, 인스턴스에 대한 기간 평가 결과, 해당 기간 동안 인스턴스의 원하는 상태, 적용된 작업 및 디버깅 메시지를 로깅합니다.

로그는 다음 두 로그 그룹으로 Amazon CloudWatch Logs에 기록됩니다.

`{stackName}-{namespace}-administrative-logs`

리소스 등록 및 등록 취소, 사용자 지정 리소스 작업, CLI 요청 및 기타 관리 활동에 대한 로그입니다.

`{stackName}-{namespace}-scheduling-logs`

오케스트레이션 및 요청 핸들러 실행을 포함한 예약 작업에 대한 로그입니다.

경고 및 오류 로그는 구독한 이메일 주소로 메시지를 보내도록 구성할 수 있는 솔루션 생성 Amazon SNS 주제로도 전달됩니다. 솔루션 스택의 출력 탭에서 Amazon SNS 주제의 이름을 찾을 수 있습니다.

## 정보 태그

정보 태그 지정이 활성화된 경우(기본값) Instance Scheduler는 관리형 리소스에 직접 태그를 작성하여 솔루션의 예약 활동에 at-a-glance 제공합니다. 이러한 태그는 AWS Resource Groups 태그 지정 API를 사용하여 적용되며 스케줄러가 리소스를 처리할 때마다 업데이트됩니다.

허브 스택에서 정보 태그 지정 활성화 파라미터를 사용하여이 기능을 활성화하거나 비활성화할 수 있습니다. 자세한 내용은 [글로벌 구성 설정 업데이트](#)를 참조하세요.

## 정보 태그 키

다음 태그는 관리형 리소스에 기록됩니다.

태그 키	설명
IS-ManagedBy	이 리소스를 관리하는 인스턴스 스케줄러 허브 스택의 ARN입니다. 리소스가 예약 및 각 후속 예약 작업에 처음 등록될 때 적용됩니다.
IS-LastAction	UTC 타임스탬프와 함께 리소스에 대해 수행된 마지막 예약 작업입니다. 예: Started 2025-06-15 09:00:00 UTC 또는 Stopped 2025-06-15 17:00:00 UTC . 이 태그는 스케줄러가 리소스를 적극적으로 시작하거나 중지할 때만 업데이트됩니다 (리소스를 평가하고 조치가 필요하지 않다고 판단할 때가 아님).
IS-Error	리소스를 처리하는 동안 스케줄러에 오류가 발생하면이 태그에는 오류 코드와 UTC 타임스탬프가 포함됩니다. 예를 들어 StartFailed 2025-06-15 09:00:05 UTC 입니다. 이 태그는 다음에 성공한 예약 작업에서 자동으로 지워집니다.
IS-ErrorMessage	사람이 읽을 수 있는 오류 설명입니다. 이 태그는 IS-Error도 있을 때만 존재하며 태그와 함께 지워집니다.

## 오류 코드

IS-Error 태그에 다음과 같은 오류 코드가 나타날 수 있습니다.

오류 코드	설명
UnknownSchedule	리소스의 일정 태그에 지정된 일정 이름이 구성 테이블에 정의된 일정과 일치하지 않습니다.
UnsupportedResource	리소스 유형은 예약에 지원되지 않습니다(예: 다른 RDS 인스턴스의 읽기 전용 복제본).
IncompatibleSchedule	리소스에 할당된 일정이 리소스 유형과 호환되지 않습니다(예: 지원되지 않는 cron 표현식을 사용하는 ASG 일정).
StartFailed	스케줄러가 리소스를 시작하려고 했지만 작업이 실패했습니다.
StopFailed	스케줄러가 리소스를 중지하려고 했지만 작업이 실패했습니다.
ConfigurationFailed	스케줄러가 Auto Scaling 그룹에서 예약된 조정 규칙을 구성하려고 시도했지만 작업이 실패했습니다.
UnknownError	리소스를 처리하는 동안 예기치 않은 오류가 발생했습니다.

## 태그 동작

- 리소스를 예약에 처음 등록하면 IS-ManagedBy 태그가 즉시 적용됩니다.
- 리소스가 등록 취소되면(일정 태그가 제거됨) 모든 정보 태그(IS-ManagedBy, IS-LastAction, IS-Error, IS-ErrorMessage)가 리소스에서 제거됩니다.
- 동일한 오류가 지속되고 기존 태그가 리소스에 여전히 있는 경우 각 예약 간격에서 오류 태그가 다시 작성되지 않습니다. 오류 코드가 변경될 때만 업데이트됩니다.
- 모든 태그 값은 AWS 태그 지정 제한을 준수하기 위해 256자로 잘립니다.

## 태그 거버넌스 고려 사항

### Important

인스턴스 스케줄러는 정상 작업의 일부로 관리형 리소스에 위에 나열된 태그를 생성하고 업데이트합니다. 조직에서 AWS Config 규칙, 태그 정책, 서비스 제어 정책 또는 자동 문제 해결을 통해 태그 거버넌스를 적용하는 경우 변경 관리 제어가 다음 태그 키를 허용하도록 구성되어 있는지 확인합니다.

- IS-ManagedBy
- IS-LastAction
- IS-Error
- IS-ErrorMessage
- IS-PreferredInstanceTypes (대체 인스턴스 유형을 사용하는 경우)
- IS-MinDesiredMax (Auto Scaling 그룹을 예약하는 경우)

거버넌스 정책에서 이러한 태그를 수용할 수 없는 경우 허브 스택No에서 정보 태그 지정 활성화 파라미터를 로 설정하여 정보 태그 지정을 비활성화합니다. 이렇게 하면 리소스 등록을 확인하는 데 사용되는 IS-ManagedBy 태그도 비활성화됩니다.

## 태그 제어

Instance Scheduler는 정보 태그 외에도 특정 기능에 대해 다음과 같은 제어 태그를 사용합니다.

태그 키	설명
IS-PreferredInstanceTypes	용량 부족으로 인해 인스턴스 시작에 실패할 때 시도할 대체 EC2 인스턴스 유형의 심표로 구분된 목록입니다. 자세한 내용은 <a href="#">EC2 용량 부족 오류 처리를 참조하세요</a> .
IS-MinDesiredMax	형식의 Auto Scaling 그룹에 대한 최소, 원하는 및 최대 용량 값입니다. min, desired, max . 자세한 내용은 <a href="#">EC2 Auto Scaling 그룹 예약을 참조하세요</a> .

## 태그 용량

### Important

AWS 리소스는 일반적으로 리소스당 50개의 태그로 제한됩니다. 인스턴스 스케줄러는 리소스에 최대 6개의 태그를 사용할 수 있습니다(정보 태그 4개와 제어 태그 최대 2개). 리소스에 기존 태그 지정 전략과 함께 인스턴스 스케줄러 태그를 수용할 수 있는 충분한 태그 용량이 있는지 확인합니다.

리소스가 50개 태그 제한에 가까워지면 정보 태그 쓰기가 실패할 수 있습니다. 스케줄러는 이러한 실패를 로깅하지만 예약 작업은 계속합니다. 태그 지정 문제가 의심되는 경우 CloudWatch Logs를 확인합니다.

## CloudWatch Logs Insights 쿼리

인스턴스 스케줄러의 구조화된 로깅 형식을 사용하면 CloudWatch Logs Insights를 사용하여 효율적으로 쿼리할 수 있습니다. Logs Insights를 사용하여 로그 데이터를 검색, 분석 및 시각화하여 운영 문제를 해결하고 예약 활동을 모니터링할 수 있습니다.

인스턴스 스케줄러는 CloudWatch Logs 콘솔의 저장된 쿼리 섹션에서 액세스할 수 있는 사전 형식의 로그 쿼리를 제공합니다.

### SchedulingHistory

시작 및 중지 작업을 포함하여 리소스에 대해 수행된 쿼리 예약 작업입니다.

### RegistrationEvents

리소스 등록 및 등록 취소 이벤트를 쿼리합니다.

### Errors

오류 로그를 쿼리하여 솔루션 관련 문제를 해결합니다.

CloudWatch Logs Insights에 대한 자세한 내용은 Amazon [CloudWatch Logs 사용 설명서의 CloudWatch Logs Insights를 사용하여 로그 데이터 분석](#)을 참조하세요. Amazon CloudWatch

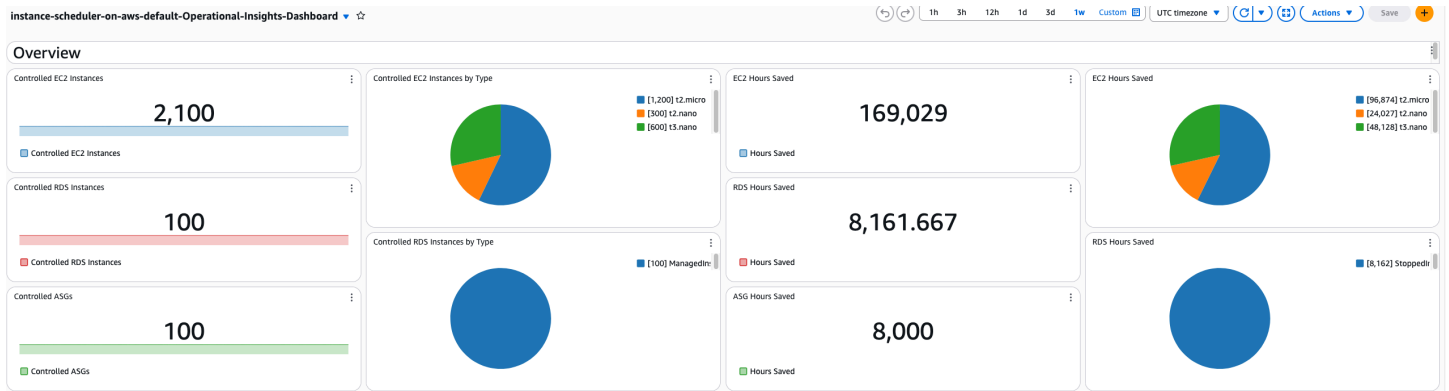
## 운영 인사이트 대시보드

Operational Insights 대시보드는 예약된 인스턴스 관리의 솔루션 성능과 비용 절감에 대한 가시성을 제공합니다.

대시보드에 액세스하려면 허브 스택 파라미터에서 운영 모니터링이 "활성화"로 설정되어 있는지 확인합니다. CloudWatch로 이동하여 탐색 메뉴에서 "대시보드"를 선택합니다. 대시보드 이름은 `*{stack-name}-Operational-Insights-Dashboard*`입니다.

대시보드에는 관리형 인스턴스 수, 저장된 실행 시간 및 Lambda 함수 성능 지표가 표시됩니다.

### 운영 인사이트 대시보드 개요

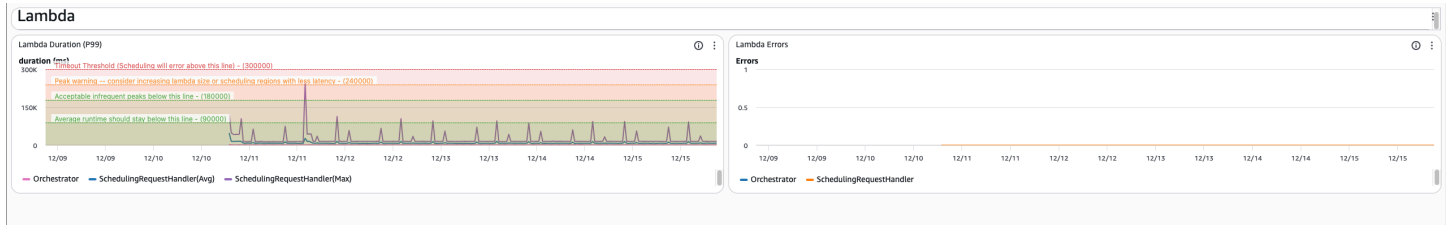


**Note**

이러한 그래프의 정보는 솔루션 허브 스택에 구성된 예약 간격에 따라 달라집니다. 솔루션의 예약 간격을 업데이트할 때 대시보드에는 가장 최근 일정 간격 업데이트 후의 일정 지표만 표시됩니다.

Lambda 실행 시간을 모니터링하여 최적의 성능을 보장합니다([할당량 참조](#)). 실행 시간이 지속적으로 제한 시간 임계값에 도달하는 경우 Lambda 크기 속성을 늘리거나 관리형 리전에 지연 시간이 더 짧은 리전에 인스턴스 스케줄러를 배포하는 것이 좋습니다.

기간 및 오류 수를 보여주는 Lambda 지표



이 기능과 관련된 추가 비용

이 운영 대시보드는 추가 비용이 발생하는 솔루션에서 수집한 사용자 지정 CloudWatch 지표를 기반으로 합니다. 솔루션 허브 스택에서 “운영 모니터링”을 비활성화하여 이 기능을 끌 수 있습니다. 이 기능은 월 3.00 USD의 추가 비용과 배포 규모에 따른 추가 스케일링 비용이 부과됩니다. 비용은 다음과 같습니다.

사용자 지정 CloudWatch 대시보드	3 USD
per-instance-type 지표	인스턴스 유형당 0.90 USD*

사용자 지정 CloudWatch 대시보드	3 USD
API 사용	<a href="#">활성 대상당</a> ~\$0.10**

\*이러한 비용은 서비스 범주(EC2/RDS)별로 추적되며 실제로 예약에 사용되는 인스턴스 유형에 대해서만 추적됩니다.

\*

## EventBridge 이벤트 모니터링

인스턴스 스케줄러는 스케줄링 및 등록 이벤트를 EventBridge 이벤트 버스에 게시하여 솔루션 작업에 대한 가시성을 제공하고 다른 AWS 서비스와 통합할 수 있도록 합니다.

### 이벤트 유형

솔루션은 두 가지 주요 이벤트 범주를 게시합니다.

**이벤트 예약:** 인스턴스 스케줄러가 관리형 리소스를 시작, 중지 또는 구성하는 작업을 수행할 때 게시됩니다. 이러한 이벤트에는 인스턴스, 일정 및 수행한 작업에 대한 세부 정보가 포함됩니다. 관리형 리소스를 시작, 중지 또는 구성합니다. 이러한 이벤트에는 인스턴스, 일정 및 수행한 작업에 대한 세부 정보가 포함됩니다.

**등록 이벤트:** 리소스가 태그 지정 작업에 따라 예약에 등록되거나 등록 취소될 때 게시됩니다.

### 이벤트 대상

**IS-LocalEvents 이벤트 버스:** IS-LocalEvents 이벤트 버스는 각 멤버 계정(허브 계정 포함)의 각 관리형 리전에 배포됩니다. 각 버스는 해당 리전 내에서 작업 및 리소스 등록을 예약하기 위한 이벤트를 수신합니다.

**IS-GlobalEvents 이벤트 버스:** 허브 계정의 IS-GlobalEvents 이벤트 버스는 모든 이벤트 버스로 전송된 각 IS-LocalEvents 이벤트의 사본을 수신하여 모든 계정 및 리전에서 중앙 집중식 모니터링을 제공합니다.

## EventBridge 이벤트 사용

EventBridge 규칙을 생성하여 다음을 수행할 수 있습니다.

- 인프라 전반의 예약 작업 모니터링
- 인스턴스가 시작되거나 중지될 때 알림 트리거
- 자동화된 워크플로를 위해 다른 AWS 서비스와 통합
- 규정 준수 모니터링 및 알림 구현

## 이벤트 구조

모든 이벤트는 표준 EventBridge 형식을 사용합니다. 다음 예제에서는 각 이벤트 유형의 구조를 보여줍니다.

이벤트 예약:

```
{
  "Source": "instance-scheduler",
  "DetailType": "Scheduling Action",
  "Resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"],
  "Detail": {
    "account": "123456789012",
    "region": "us-east-1",
    "service": "ec2",
    "resource_id": "i-1234567890abcdef0",
    "requested_action": "Start",
    "action_taken": "Started",
    "schedule": "office-hours"
  }
}
```

등록 이벤트:

```
{
  "Source": "instance-scheduler",
  "DetailType": "Resource Registered",
  "Resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"],
  "Detail": {
    "account": "123456789012",
    "region": "us-east-1",
    "service": "ec2",
    "resource_id": "i-1234567890abcdef0",
    "schedule": "office-hours"
  }
}
```

```
}

```

각 이벤트에는 다음과 같은 키 필드가 포함됩니다.

- Source - 이벤트를 소스를 "instance-scheduler"로 식별합니다.
- DetailType - 이벤트 범주를 지정합니다. 인스턴스 작업의 경우 "예약 작업", 이벤트 태그 지정의 경우 "리소스 등록"
- Resources - 영향을 받는 AWS 리소스의 ARNs 포함된 배열
- Detail - 계정 ID, 리전, 서비스 유형(ec2/rds), 리소스 ID, 일정 이름, 요청된 작업 및 실제 결과 모두의 이벤트 예약에 대한 이벤트 페이로드를 포함합니다.

이벤트 예약에 사용할 수 있는 requested\_action 값:

- Start: 인스턴스를 시작하기 위한 스케줄러
- Stop: 인스턴스를 중지하기 위한 스케줄러
- Configure: 인스턴스를 구성하기 위한 스케줄러

이벤트 예약에 사용할 수 있는 action\_taken 값:

- Started: 인스턴스가 시작됨
- Stopped: 인스턴스가 중지되었습니다.
- Hibernated: 인스턴스가 최대 절전 모드로 전환됨
- Configured: 인스턴스 구성이 수정되었습니다.
- Error: 예약 작업 중에 오류가 발생했습니다.

## EventBridge 규칙 생성

인스턴스 스케줄러 이벤트를 모니터링하려면:

1. AWS 계정의 EventBridge 콘솔로 이동합니다.
2. IS-GlobalEvents 이벤트 버스(중앙 집중식 모니터링용) 또는 IS-LocalEvents 이벤트 버스(로컬 모니터링용)를 대상으로 하는 새 규칙 생성
3. 인스턴스 스케줄러 이벤트와 일치하는 이벤트 패턴 정의
4. SNS 주제, Lambda 함수 또는 CloudWatch Logs와 같은 대상 구성

EventBridge에 대한 자세한 내용은 [Amazon EventBridge란 무엇입니까?](#)를 참조하세요. Amazon EventBridge 사용 설명서의 .

## 문제 해결

이 섹션에서는 솔루션 배포 및 사용에 대한 문제 해결 지침을 제공합니다.

알려진 문제 해결은 알려진 오류를 완화하기 위한 지침을 제공합니다. 이 지침으로 문제가 해결되지 않을 경우 [Support에 문의하세요](#)에서 이 솔루션에 대한 지원 사례를 여는 방법을 안내해 드립니다.

### 알려진 문제 해결

#### 문제: 원격 계정에서 예약되지 않은 인스턴스(v1.4-v3.0)

원격 계정에서 인스턴스가 예약되지 않는 경우입니다.

#### 해결 방법

보조 계정 ID로 허브 스택을 업데이트하거나 다음 작업을 완료합니다.

1. 기본 계정에서 [CloudWatch 콘솔](#)로 이동합니다.
2. 탐색 창에서 로그 > 로그 그룹을 선택합니다.
3. <STACK\_NAME>-logs라는 로그 그룹을 선택합니다.
4. 계정 ID(원격 계정)에 대한 로그 스트림을 검색합니다.
5. 예를 들어, 계정 ID로 명명된 로그 스트림이 없는 경우 DynamoDB 콘솔로 이동하여 <STACK\_NAME>-<ConfigTable>-<RANDOM>이라는 테이블을 선택합니다.
6. 항목 탐색을 선택하고 실행을 선택합니다.
7. 유형이 Config인 항목을 선택합니다.
8. 속성 remote\_account\_ids에 계정 ID가 있는지 확인합니다.
9. 이 속성에 계정 ID가 표시되지 않는지 확인합니다.
10. 솔루션이 AWS Organizations에 구성되어 있는 경우 원격 계정에서 원격 템플릿을 제거한 후 다시 설치합니다.
11. 솔루션이 원격 계정 ID를 사용하도록 구성된 경우 인스턴스가 예약될 계정 ID 목록과 원격 템플릿이 배포될 계정 ID 목록으로 CloudFormation 파라미터 조직 ID 또는 원격 계정 ID 목록 제공을 업데이트합니다.

## 문제: 예약되지 않은 인스턴스(v3.1 이상)

인스턴스가 예약되지 않은 것을 발견한 경우.

### 해결 방법

1. 리소스에 IS-ManagedBy 태그가 적용되었는지 확인합니다.
2. 태그가 없는 경우 일정 태그를 삭제하고 다시 생성하여 등록을 다시 트리거합니다.
3. 태그가 여전히 적용되지 않는 경우 리전에 예약이 활성화되어 있는지 확인합니다.
  - a. 리전의 허브/스포크 스택 구성을 확인하거나
  - b. 리소스와 동일한 리전의 [EventBridge 콘솔](#)로 이동하여 기본 이벤트 버스에 접두사 IS-Tagging이 있는 이벤트 규칙이 있는지 확인합니다.
4. 리전이 활성화되지 않은 경우 리전 CloudFormation 파라미터에 리전을 포함하도록 인스턴스 스케줄러 스택을 업데이트합니다.
5. 문제가 지속되면 [솔루션 관리 로그](#)에서 허브 등록 오류를 검토합니다.
6. 조직에서 이벤트가 계정에서 솔루션 허브 계정으로 전달되지 않도록 하는 정책이 없는지 확인합니다.

## 문제: 암호화된 EC2 인스턴스가 시작되지 않음

인스턴스 스케줄러는 암호화된 EBS 볼륨을 가진 EC2 인스턴스가 시작되고 있다고 보고하지만, 실제로는 시작되지 않습니다.

### 해결 방법

암호화된 EBS 볼륨으로 EC2 인스턴스를 예약할 수 있도록 인스턴스 스케줄러 액세스 권한을 부여하는 방법은 [암호화된 EC2 EBS 볼륨](#)을 참조합니다.

## 문제: 정보 태그 지정으로 인한 예상치 못한 API 비용

AWS Resource Groups 태깅 API 호출, AWS Config 평가 또는 관련 문제 해결 작업으로 인한 비용이 예기치 않게 높습니다.

### 해결 방법

인스턴스 스케줄러는 각 예약 간격에서 관리형 리소스에 [정보 태그](#)를 씁니다. 환경에서 AWS Config 규칙, 태그 정책 또는 자동 문제 해결을 통해 태그 거버넌스를 적용하는 경우 인스턴스 스케줄러의 태

그 키가 허용되는지 확인합니다. 태그 키 및 구성 지침의 전체 목록은 [태그 거버넌스 고려 사항](#)을 참조하세요.

태그 거버넌스 정책을 업데이트할 수 없는 경우 허브 스택에서 정보 태그 지정 활성화 파라미터를 로 설정하여 정보 태그 지정No를 비활성화합니다.

## 문제: RDS 스냅샷 생성이 활성화되어 있으면 RDS 인스턴스가 중지되지 않음

rds:CreateDBSnapshot 권한이 없어 StopDBInstance 작업을 직접 호출할 때 RDS 인스턴스가 중지되지 않고 솔루션의 스케줄러 로그가 (AccessDenied) 오류를 보고합니다.

### 해결 방법

솔루션을 v3.0.5 이상으로 업데이트하거나 예약된 각 계정에서 솔루션의 스케줄러 역할에 rds:CreateDBSnapshot 권한을 추가합니다.

## AWS Support에 문의

[AWS Business Support+](#), [AWS Enterprise Support](#) 또는 [통합 운영](#)이 있는 경우 AWS 지원 센터를 사용하여 솔루션에 대한 전문가 지원을 받을 수 있습니다. 이후 단원에서는 그 방법에 대해서 설명합니다.

### 사례 생성

1. [지원 센터](#)에 로그인합니다.
2. 사례 생성을 선택합니다.

### 지원 방법

1. 기술을 선택합니다.
2. 서비스에서 솔루션을 선택합니다.
3. 카테고리에서 AWS 인스턴스 스케줄러(Linux 또는 Windows)를 선택합니다.
4. 심각도에서 사용 사례에 가장 적합한 옵션을 선택합니다.
5. 서비스, 카테고리 및 심각도를 입력하면 인터페이스가 일반적인 문제 해결 질문에 대한 링크를 제공합니다. 이러한 링크로 질문을 해결할 수 없는 경우 다음 단계: 추가 정보를 선택합니다.

## 추가 정보

1. 제목에 질문 또는 문제를 요약하는 텍스트를 입력합니다.
2. 설명에서이 제품의 이름과 사용 중인 버전을 포함하여 문제를 자세히 설명합니다. 예: Instance Scheduler on AWS vX.Y.Z.
3. 파일 연결을 선택합니다.
4. Support에서 요청을 처리하는 데 필요한 정보를 첨부합니다.

## 사례를 더 빠르게 해결할 수 있도록 지원

1. 필요한 정보를 입력합니다.
2. 다음 단계: 지금 해결하거나 AWS에 문의하기를 선택합니다.

## 지금 해결 또는 문의

1. 지금 해결 솔루션을 검토합니다.
2. 이러한 솔루션의 문제를 해결할 수 없는 경우 문의를 선택하고 요청된 정보를 입력한 다음 제출을 선택합니다.

## 솔루션 업데이트

인스턴스 스케줄러는 AWS CloudFormation을 사용하여 현재 위치에서 안전하게 업데이트할 수 있도록 설계되었습니다. 이 작업을 수행하는 일반적인 절차는 다음과 같습니다.

1. [AWS CloudFormation 콘솔](#)에 로그인하여 허브 스택이 설치된 계정/리전에서 instance-scheduler-on-aws를 선택하고 스택 업데이트를 선택합니다.
2. 직접 업데이트를 선택합니다.
3. 기존 템플릿 교체를 선택합니다.
4. 템플릿 지정에서 다음을 수행합니다.
  - Amazon S3 URL을 선택합니다.
  - [최신 템플릿](#)의 링크를 복사합니다.
  - Amazon S3 URL 상자에 링크를 붙여넣습니다.
  - Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 표시되는지 확인하고 다음을 선택합니다. 다음을 다시 선택합니다.
5. 파라미터에서 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다(필수 파라미터 업데이트는 아래 주요 변경 사항 목록 참조). 각 파라미터에 대한 자세한 내용은 [1단계. 인스턴스 스케줄러 스택 시작](#)을 참조하세요.
6. 다음을 선택합니다.
7. 스택 옵션 구성 페이지에서 다음을 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management(IAM) 리소스를 생성할 것임을 승인하는 확인란을 선택합니다.
9. 변경 세트 보기를 선택하고 변경 사항을 확인합니다.
10. 스택 생성을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 몇 분 후에 UPDATE\_COMPLETE 상태를 받게 됩니다.

각 스포크 계정의 aws-instance-scheduler-remote 스택에 대해 위 단계를 반복합니다.

## 특정 버전의 주요 변경 사항

솔루션을 업데이트할 때 중요한 데이터가 손실되거나 일정이 중단되지 않고 이전 버전에서 최신 버전으로 직접 업그레이드할 수 있습니다. 각 메이저 버전의 동작 및 주요 변경 사항 목록은 아래를 참조하세요.

[솔루션의 GitHub 페이지에서 전체 변경 로그를 볼 수 있습니다.](#)

### v1.5.0

버전 1.5.0은 교차 계정 예약 역할 ARN 목록을 제공할 필요성을 AWS Organizations을 통해 자동으로 관리할 수 있는 기능으로 대체합니다. AWS Organizations을 사용하지 않으려면 스포크 계정 ID 목록을 대신 제공할 수 있습니다. 그러면 인스턴스 스케줄러가 예약 역할을 관리합니다.

v1.5.0 이상으로 업데이트할 때는 다음을 수행해야 합니다.

1. 다음과 같은 파라미터를 업데이트하는 동안 일반 업데이트 지침을 사용하여 허브 템플릿을 업데이트합니다.
  - a. 솔루션의 고유한 네임스페이스를 선택합니다.
  - b. AWS Organizations을 사용하여 향후 스포크 등록을 관리할지 여부를 선택합니다.
    - i. 예를 선택한 경우 조직 ID/원격 계정 ID를 AWS Organizations의 ID로 대체합니다.
    - ii. 아니요를 선택한 경우 조직 ID/원격 계정 ID를 스포크 계정의 계정 ID를 심포로 구분된 목록으로 대체합니다.
2. 다음과 같은 파라미터를 업데이트하는 동안 일반 업데이트 지침을 사용하여 모든 원격 스택을 업데이트합니다.
  - a. 네임스페이스 - 허브 계정에 대해 선택한 것과 동일합니다.
  - b. AWS Organizations 사용 - 허브 계정과 동일합니다.
  - c. 허브 계정 ID - 허브 계정의 계정 ID입니다(이전과 동일해야 함).

### v3.0.0

v3.0.0 EC2 Autoscaling 그룹에 대한 지원을 추가하고 솔루션의 코어 Lambda 함수를 개별 함수에 대해 더 나은 보안 격리를 제공하는 전용 책임이 있는 별도의 함수로 분리합니다. 또한 릴리스에서는 예약 작업에 대한 인사이트를 높이기 위해 "SchedulingDecision" 로그를 포함하도록 예약 로그 동작을 업데이트합니다.

V3.0.0에는 이전 버전과 비교하여 다음과 같은 주요 변경 사항이 포함되어 있습니다.

- 1.5.x의 “CloudWatch 지표” 기능이 [운영 인사이트 대시보드](#)로 교체되었습니다.
- CloudWatch의 일정별 지표가 Schedule/Service/MetricName에서 Schedule/Service/SchedulingInterval/MetricName으로 이동되었습니다.
- 기존 지표는 모두 유지되지만, 이제 새 네임스페이스에서 새 지표가 수집되어 솔루션 대시보드에서 사용할 수 있습니다.
- 이제 EC2 DB 인스턴스에서 암호화된 EBS 볼륨에 사용할 KMS 키 ARN을 해당 계정의 허브/스포크 CloudFormation 스택에 제공해야 합니다. (자세한 내용은 [암호화된 EC2 EBS 볼륨](#) 참조)
  - 암호화된 EBS 볼륨으로 EC2를 예약하는 경우 사용 중인 KMS 키 ARN을 허브/스포크 스택 파라미터에 복사해야 합니다.
- 예약된 서비스에 대한 CloudFormation 파라미터는 지원되는 각 서비스에 대한 개별 파라미터로 구분되었습니다.
  - 모든 서비스는 기본적으로 활성화되며 개별적으로 비활성화할 수 있습니다.
- 인스턴스 스케줄러 3.0은 이전 버전의 인스턴스 스케줄러 CLI와 호환되지 않습니다.
  - CLI 명령을 계속 사용하려면 인스턴스 스케줄러 CLI의 최신 버전으로 업데이트해야 합니다.

위의 내용 외에도 유지 관리 기간 테이블의 스키마가 업데이트되었으며, 이는 업데이트의 일환으로 교체될 예정입니다. 이렇게 하면 v3.x로 업데이트한 후 처음 몇 분 동안 EC2 유지 관리 기간에 대한 추적이 재설정되며, 드문 경우이긴 하지만 현재 유지 관리 기간 내에 있는 인스턴스가 업데이트 직후에 조기 중지될 수 있습니다. 이 데이터가 재생성되면 예약 작업이 정상적으로 계속됩니다.

## v3.1.0

v3.1.0은 AWS 태그 지정 이벤트를 사용하여 리소스에 예약 태그가 지정된 시기를 추적하도록 솔루션의 핵심 인프라를 리팩터링합니다. 조직의 권한이 이러한 태그 지정 이벤트를 멤버 계정에서 중앙 허브 계정으로 전송할 수 있도록 허용하는지 확인하세요.

v3.1.0 이상으로 업데이트하는 경우:

- 이제 스포크 계정은 허브 계정과 독립적으로 예약된 리전을 선언합니다. 각 스포크 스택은 리전(들) 파라미터를 사용하여 해당 계정에서 예약할 리전을 지정해야 합니다.
- 이제 총 계정이 40개 이상인 배포에는 AWS Organizations 모드가 필요합니다. 계정이 40개 이상이고 Organizations 모드를 사용하지 않는 경우 업데이트 중에 활성화해야 합니다.

- 예약하려는 AWS License Manager에서 EC2 인스턴스를 관리하는 경우 허브/스포크 CloudFormation 스택의 License Manager 구성 ARNs 파라미터에 License Manager 구성 ARNs를 추가합니다. 자세한 내용은 [EC2 License Manager](#)를 참조하세요.
- 솔루션은 스케줄러에 의해 관리되고 있음을 나타내기 위해 일정에 대한 태그가 지정된 후 리소스에 IS-ManagedBy 태그를 자동으로 적용합니다.
- (v3.2.0에서 복원됨) 예약된 인스턴스 크기 조정(일정period-name@size에서 정의)이 v3.1.0에서 일시적으로 제거되었지만 v3.2.0 이상에서 다시 구현되었습니다. [인스턴스 유형을 참조하세요](#).
- SSM 파라미터(허브 스택의 계정 파라미터{param: ssm-param-name}로 전달)를 통한 멤버 계정 나열은 더 이상 지원되지 않습니다. 신뢰할 수 있는 모든 계정은 배포 시 허브 스택으로 전달되어야 합니다.
- 인스턴스 스케줄러는 예약 중에 리소스에 최대 6개의 고유 태그를 요구합니다. 조직의 나머지 태그 지정 전략과 결합할 때 리소스에 대한 태그 지정 용량이 충분한지 확인하세요.
- CloudWatch에서 일정별 지표가 제거되었습니다.
- 솔루션 로그는 별도의 관리 및 예약 로그 그룹으로 다시 패키징되었으며 CloudWatch Log Insights를 사용한 쿼리에 최적화되었습니다. 자세한 내용은 [솔루션 모니터링](#)을 참조하세요.
- CloudFormation 파라미터를 통해 시작 및 중지 태그를 더 이상 구성할 수 없습니다. 이제 솔루션은 예약 작업을 추적하기 위해 더 풍부한 정보가 포함된 고정 태그 이름을 사용합니다.

#### Important

인스턴스 스케줄러는 정상 작동 중에 관리형 리소스에 최대 6개의 고유 태그를 씁니다. 태그 거버넌스 정책(예: AWS Config 규칙, 태그 정책 또는 자동 문제 해결)이 이러한 태그를 허용하도록 구성되어 있는지 확인합니다. 태그의 전체 목록과 중요한 거버넌스 고려 사항은 [정보 태그](#)를 참조하세요.

## 솔루션 제거

### Important

솔루션을 제거할 때는 솔루션 자체를 제거하기 전에 모든 사용자 지정 일정 스택을 제거해야 합니다.

AWS 인스턴스 스케줄러 솔루션은 AWS Management Console에서 제거하거나 AWS Command Line Interface를 사용하여 제거할 수 있습니다. 솔루션을 제거하려면 설치된 모든 원격 스택과 함께 AWS CloudFormation에서 허브 스택을 삭제합니다. 그런 다음 예약 목적으로 인스턴스에 적용된 모든 예약 태그를 제거할 수 있습니다.

### Note

솔루션의 허브 스택에서 DynamoDB 테이블 보호가 활성화되어 있는 경우 CloudFormation은 솔루션의 DynamoDB 테이블과 KMS 키를 삭제하지 않고 유지합니다. 이러한 리소스를 삭제하려면 허브 스택을 삭제하기 전에 이 속성이 비활성화됨으로 설정되어 있는지 확인합니다. 또는 허브 스택이 이미 삭제된 후에 수동으로 삭제할 수 있습니다.

## AWS 관리 콘솔 사용

1. [AWS CloudFormation 콘솔](#)에 로그인합니다.
2. 스택 페이지에서 이 솔루션의 설치 스택을 선택합니다.
3. 삭제를 선택합니다.

## AWS Command Line Interface 사용

사용자 환경에서 AWS Command Line Interface(AWS CLI)를 사용할 수 있는지 확인합니다. 설치 지침은 AWS CLI 사용 설명서의 [AWS Command Line Interface란 무엇입니까?](#)를 참조하세요. AWS CLI를 사용할 수 있는지 확인한 후 다음 명령을 실행합니다.

```
$ aws cloudformation delete-stack --stack-name
    <installation-stack-name>
```

## 개발자 안내서

이 섹션에서는 솔루션의 소스 코드를 제공하고, 여기에 추가된 섹션을 나열하며, 각 하위 주제에 대한 링크를 제공합니다.

### 소스 코드

이 솔루션의 소스 파일을 다운로드하고 사용자 지정 사항을 다른 사람과 공유하려면 [GitHub 리포지토리](#)를 방문하세요.

AWS 인스턴스 스케줄러 템플릿은 [AWS CDK](#)를 사용하여 생성됩니다. 자세한 내용은 [README.md](#) 파일을 참조하세요.

## 레퍼런스

이 섹션에는 데이터 수집, [관련 리소스](#)에 대한 포인터,이 솔루션에 기여한 [빌더 목록](#)에 대한 정보가 포함되어 있습니다.

## 데이터 수집

이 솔루션은이 솔루션 사용에 대한 운영 지표("데이터")를 AWS로 전송합니다. 당사는이 데이터를 사용하여 고객이이 솔루션과 관련 서비스 및 제품을 사용하는 방법을 더 잘 이해합니다. AWS의이 데이터 수집에는 [AWS 개인 정보 보호 고지](#)가 적용됩니다.

## 관련 리소스

[Resource Scheduler](#)는 AWS 인스턴스 스케줄러와 유사하지만, 구현 방식은 다음과 같은 점에서 다릅니다.

AWS 인스턴스 스케줄러는 Lambda 함수를 사용하여 구성에 저장된 일정 평가를 빈도 높게 수행하고 인스턴스가 원하는 상태인지 확인합니다. Resource Scheduler 빠른 설정에서는 시작 및 중지 시간을 사용하여 SSM 런북을 사용하여 시작 및 중지 작업을 수행합니다. 이는 현재 시간이 시작 시간과 같거나 현재 시간이 시작 시간 이후일 때 한 번 수행됩니다.

AWS 인스턴스 스케줄러는 현재 EC2, RDS 및 Aurora 클러스터에 대한 예약을 활성화합니다. Resource Scheduler는 EC2 인스턴스만 예약하거나 시작 및 중지합니다.

Resource Scheduler를 사용하여 EC2 인스턴스를 식별하고 특정 시간에 인스턴스를 시작/중지합니다.

계정이 인스턴스를 시작/중지하기 위해 정기적으로 스캔해야 하는 경우 AWS 인스턴스 스케줄러를 사용합니다.

이 표는 시나리오를 기반으로 어떤 솔루션이 더 적합한지 식별합니다.

시나리오	리소스 스케줄러	AWS 인스턴스 스케줄러
Amazon Neptune 인스턴스 예약	아니요	예
Amazon DocumentDB 인스턴스 예약	아니요	예

시나리오	리소스 스케줄러	AWS 인스턴스 스케줄러
Auto Scaling 그룹 인스턴스 예약	아니요	예
EC2 인스턴스 예약	예	예
RDS 인스턴스 예약	아니요	예
Aurora 클러스터 예약	아니요	예
단일 계정(허브 계정)에서 일정 관리	아니요	예
개별 계정에서 일정 관리	예	아니요
일정 통합 변경	예	아니요
시작 및 중지 작업만	예	아니요
인스턴스의 주기적 모니터링 및 인스턴스 현재 상태에 따른 시작 및 중지	아니요	예

## 기여자

- Arie Leeuwesteijn
- Mahmoud ElZayet
- Ruald Andreae
- Nikhil Reddy
- Caleb Pearson
- Jason DiDomenico
- Max Granat
- Pratyush Das
- Amanda Jones
- Kevin Hargita

- Beomseok Lee
- Abe Wubshet

# 개정

게시 날짜: 2020년 10월

GitHub 리포지토리의 [CHANGELOG.md](#) 파일을 보고 소프트웨어에 대한 모든 주요 변경 사항 및 업데이트를 확인하세요. 변경 로그는 각 버전의 개선 및 수정 사항에 대한 명확한 기록을 제공합니다.

## Notices

고객은 본 문서의 정보를 독립적으로 평가할 책임이 있습니다. 이 문서는 (a) 정보 제공만을 목적으로 하고, (b) AWS의 현행 제품 제공 및 관행을 나타내며, (c) AWS와 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 약정이나 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 “있는 그대로” 제공됩니다. 고객에 대한 AWS의 책임 및 채무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경 하지도 않습니다.

AWS 인스턴스 스케줄러는 [Apache License Version 2.0](#)의 약관에 따라 라이선스가 부여됩니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.