



AWS CloudFormation

학습 내용

강의의 핵심

배울 내용은 다음과 같습니다.

- AWS CloudFormation의 목적 설명하기
- AWS CloudFormation 템플릿 생성하기
- AWS CloudFormation 모범 실무 사용하기

주요 용어:

- AWS CloudFormation 템플릿
- AWS CloudFormation 스택



클라우드 배포의 당면 과제

클라우드 배포의 당면 과제

클라우드 배포 시 당면 과제는 다음과 같습니다.

여러 지리 위치
전반에 롤아웃 배포

배포 디버깅

모든 변경 사항을
문서화

라이브 서버
업데이트

롤백을 관리하기
위한 기능 보유

시스템 및 하위
시스템에 대한
종속성 관리

반복 가능하며
동일한 환경 배포



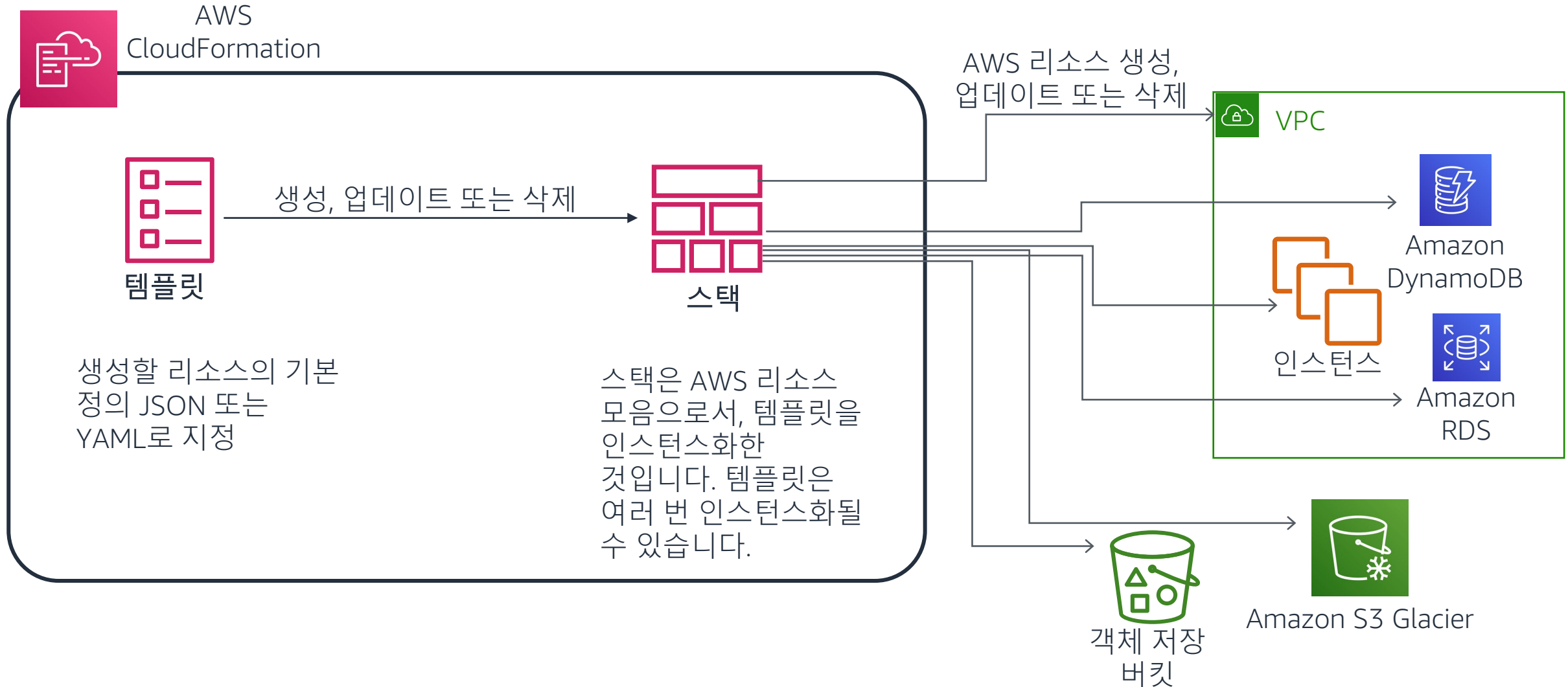
AWS CloudFormation

AWS CloudFormation

- 클라우드 인프라 리소스 모델링 및 프로비저닝
- 대다수 AWS 서비스 지원
- **스택**이라는 단일 단위로 리소스 세트 생성, 업데이트 및 삭제
- 스택 및 개별 리소스에서 '드리프트'라고 하는 변경 사항 감지



AWS CloudFormation 용어



템플릿 구조



템플릿

■ Parameters

■ Mappings

■ Resources

◆ AWS::EC2::Instance

- + Properties/UserData
- + Metadata/AWS::CloudFormation::Init
 - Packages
 - Groups
 - Users
 - Sources
 - Files
 - Commands
 - Services

◆ AWS::CloudFormation::WaitCondition

- DependsOn: EC2 Instance

◆ AWS::CloudFormation::WaitConditionHandle

■ Outputs

Parameters: 템플릿에 입력

Mappings: 정적 변수 - 일반적으로 최신 Amazon Machine Image(AMI)

Resources: 생성할 AWS 자산

Init: 시작 시 실행할 사용자 지정 애플리케이션

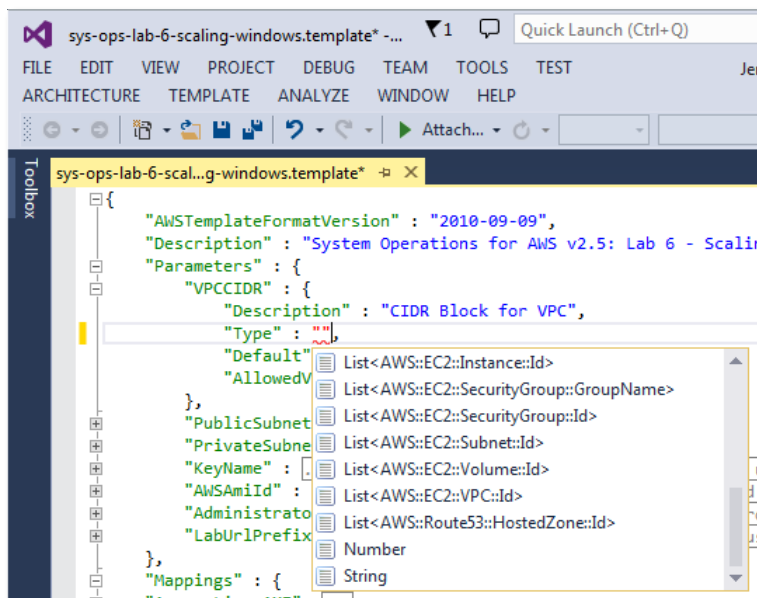
WaitCondition: 사용자 데이터가 실행을 완료한 인스턴스의 신호

Outputs: 템플릿이 생성하는 사용자 지정 리소스 값(URL, 사용자 이름 등)

AWS CloudFormation 템플릿 편집

AWS CloudFormation과 호환되는 편집기(또는 JSON 또는 YAML 편집기)를 사용하여 **구문 분석 기능**, **자동 완성** 및 **구문 검사**와 같은 기능을 사용합니다.

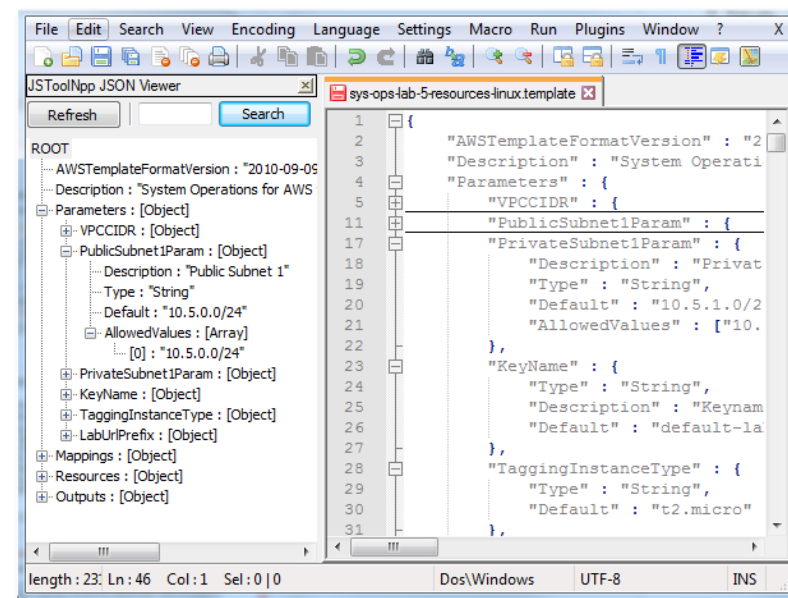
Visual Studio 또는 Eclipse용 AWS CloudFormation 템플릿 편집기



제3자 도구



텍스트 편집기 JavaScript 또는 JSON 플러그인

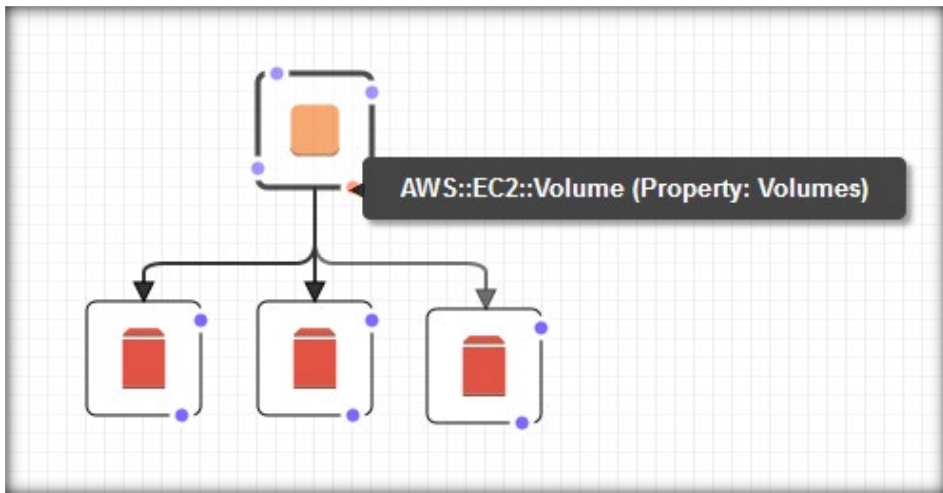


템플릿 디자인

AWS CloudFormation Designer는 끌어서 놓기 인터페이스를 사용해 AWS CloudFormation 템플릿을 생성 및 수정할 수 있게 하는 비주얼 도구입니다.

AWS CloudFormation Designer

Design Surface



JSON 또는 YAML 편집기



The screenshot shows the JSON or YAML editor in AWS CloudFormation Designer. The interface includes tabs for Properties, Metadata, CreationPolicy, DeletionPolicy, and DependsOn. The main area displays a JSON snippet for an EC2 instance resource named "I4NO6Y". The snippet includes properties for InstanceType, DisableApiTermination, Monitoring, and Volumes. The Volumes property is expanded, showing a list of volume configurations with VolumeId and Ref attributes.

```
1 {  
2   "Resources": {  
3     "I4NO6Y": {  
4       "Type": "AWS::EC2::Instance",  
5       "Properties": {  
6         "InstanceType": "t2.medium",  
7         "DisableApiTermination": "true",  
8         "Monitoring": "true",  
9       },  
10      "Volumes": [  
11        {  
12          "VolumeId": {  
13            "Ref": "VOL10ZQL"  
14          },  
15        },  
16      ]  
17    }  
18  }  
19 }
```

제3자 도구



스택 시작 및 삭제

- AWS CloudFormation 템플릿을 스택으로 시작 가능:
 - AWS Management Console
 - AWS CLI
 - AWS API
- 템플릿을 시작할 때 오류가 발생하면 모든 리소스가 기본적으로 롤백됩니다.
- 스택이 삭제되면 리소스가 롤백됩니다.
 - 선택적으로 스택에서 종료 방지 기능을 활성화할 수 있습니다.

템플릿에서 파라미터 정의

템플릿에서 파라미터를 정의하는 방법:

- 선택 사항인 **Parameters** 섹션을 사용하여 템플릿을 사용자 지정합니다.
- 파라미터를 사용하면 스택을 생성하거나 업데이트할 때마다 템플릿에 사용자 지정 값을 입력할 수 있습니다.

```
"Parameters": {  
  "VPCCIDR": {  
    "Description": "CIDR Block for VPC",  
    "Type": "String",  
    "Default": "10.200.0.0/20",  
    "AllowedValues": [ "10.200.0.0/20" ]  
  }  
}
```

예제 파라미터

논리명(템플릿 내에서 고유함)

지원되는 데이터
유형(String, Number, List,
CommaDelimitedList 중 하나,
AWS에 특정한 파라미터 또는
AWS Systems Manager
파라미터)을 지정해야 함

파라미터 참조

파라미터를 참조하는 방법:

- 파라미터를 참조하려면 **Ref** 내장 함수를 사용합니다.
- 파라미터 선언의 형식과 관계없이 문자열로 변환됩니다.

파라미터

```
"KeyName": {"Type": "AWS::EC2::KeyPair::KeyName"}
```

파라미터 참조

```
"KeyName": {"Ref": "KeyName"},
```

- **Fn::Select** 함수를 사용하여 심표로 구분된 목록에서 값을 선택합니다.

```
"AvailabilityZone" : { "Fn::Select" : [ "0", { "Ref" : "AvailableAZs" } ] }
```

Ref 및 기타 내장 함수

Ref 함수:

- AWS CloudFormation 템플릿에 정의된 참조 구성 요소를 활성화합니다.
- 필요한 경우:
 - 파라미터 참조
 - 맵 사용
 - 문자열 조인
 - 다른 함수 사용

```
"MyEIP" : {  
  "Type" : "AWS::EC2::EIP",  
  "Properties" : {  
    "InstanceId" : { "Ref" : "MyEC2Instance" }  
  }  
}
```

의사 파라미터

추가 사항:

- **Ref** 내장 함수를 사용하여 런타임 환경에 대한 정보에 액세스할 수 있습니다.
 - 예: 리전, 스택 이름, AWS 계정 ID
- 의사 파라미터는 미리 정의되어 있으므로 템플릿의 **Parameters** 섹션에서 지정할 필요가 없습니다.

```
"Outputs" {  
  "MyStacksRegion" : { "Value" : { "Ref" : "AWS::Region" } }  
}
```

템플릿에서 매핑 정의

템플릿에서 매핑을 정의하는 방법:

- 2단계 맵에서 이름-값 페어의 조회 테이블을 정의합니다.
- 여러 매핑 테이블을 결합하여 중첩되는 조회 결과를 생성합니다.
- 현재 Amazon Machine Image(AMI) 값을 조회하는 데 가장 일반적으로 사용됩니다.

매핑 예제

```
"Mappings" : {  
  "AWSRegionToAMI" : {  
    "us-east-1" : { "AMI" : "ami-76817c1e" },  
    "us-west-2" : { "AMI" : "ami-d13845e1" },  
    "us-west-1" : { "AMI" : "ami-f0d3d4b5" },  
    "eu-west-1" : { "AMI" : "ami-892fe1fe" }, ...  
  }  
}
```

```
"ImageId" : {  
  "Fn::FindInMap" ← [  
    "AWSRegionToAMI",  
    { "Ref" : "AWS::Region" },  
    "AMI"  
  ]  
}
```

조회에 Fn::FindInMap 내장
함수 사용

템플릿에서 리소스 정의

Resources 섹션은 생성할 AWS 리소스를 선언합니다.

선언하는 리소스 유형

특정 리소스의 생성이 또 다른 리소스 생성의 뒤를 이어
진행되게 함

리소스 속성(선택 사항)

```
"Resources" : {  
  "MyRDSInstance" : {  
    "Type" : "AWS::RDS::DBInstance",  
    "Properties" : {  
      "AllocatedStorage" : "5",  
      ...  
    }  
  },  
  "WebServer" : {  
    "Type" : "AWS::EC2::Instance",  
    "DependsOn" : [ "MyRDSInstance" ],  
    "Properties" : {  
      ...  
    }  
  }  
}
```

CloudFormation::Init

- EC2 인스턴스의 메타데이터에 대한 액세스를 제공하는 리소스 유형
- `cfn-init` 헬퍼 스크립트와 함께 사용할 수 있습니다.
- `cfn-init` 헬퍼 스크립트:
 - `AWS::CloudFormation::Init` 키에서 템플릿 메타데이터를 읽습니다.
 - 이 정보를 사용하여 `cfn-init`는 EC2 인스턴스에서 다음 작업을 수행할 수 있습니다.
 - » 패키지 설치
 - » 그룹 및 사용자 계정 관리
 - » 디스크에 파일 기록
 - » 명령 실행
 - » 서비스 활성화 또는 비활성화
 - » 서비스 시작 또는 중지

사용자 데이터 VS. CloudFormation::Init

AWS CloudFormation의 사용자 데이터:

- 파일 스크립팅(개별 인스턴스와 동일)
- 향상된 제어 제공
- 더 높은 오류 가능성 제공 보다 신중해야 함

CloudFormation::Init:

- 실패 시 자동으로 롤백할 수 있음
- CloudFormation 템플릿의 사용자 데이터 필드보다 관리가 쉬움
- 템플릿 내에서 메타데이터를 연결할 수 있음
- **Configset**는 단일 CloudFormation::Init에서 여러 구성을 허용

WaitCondition 및 WaitConditionHandle

`waitCondition` 및 `waitConditionHandle`은 인스턴스를 부트스트랩할 때 사용됩니다.
추가 사항:

- `waitCondition`은 `waitConditionHandle`이 호출되거나 시간 제한에 도달할 때까지 스택의 완료 상태를 차단합니다.
 - `waitCondition`이 충족되기 전에 수신되어야 하는 성공적인 신호의 수를 지정할 수 있습니다.
 - 기본값: 1
- `cfn-signal` 명령을 사용하여 성공 또는 실패 신호를 보낼 수 있습니다.

WaitCondition 및 WaitConditionHandle

WaitConditionHandle 지정

```
"webServerWaitHandle": {
  "Type":
  "AWS::CloudFormation::WaitConditionHandle"
},
"webServerWaitCondition": {
  "Type":
  "AWS::CloudFormation::WaitCondition",
  "DependsOn": "webServerInstance",
  "Properties": {
    "Handle": {
      "Ref": "webServerWaitHandle"
    },
    "Timeout": "3600"
  }
}
```

WaitCondition 신호 전송

```
"UserData": {
  "Fn::Base64": {
    "Fn::Join": [ "", [
      "#!/bin/bash\n",
      "/opt/aws/bin/cfn-signal -e 0
-r \"complete\" \"\",
      {
        "Ref":
"webServerWaitHandle"
      }, "\"\n"
    ]
    ]
  }
}
```

템플릿에서 출력 정의

Outputs:

- 스택의 일부로 생성된 리소스에 대한 값을 반환합니다.
- 내장 함수를 사용하여 스택의 리소스 값을 얻습니다.

```
"Outputs" : {  
  "WebsiteURL" : {  
    "Description" : "The URL of the website",  
    "Value" : { "Fn::Join" : [ "", [ "http://", {  
      "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ] } ] ] }  
    }  
  }  
}
```

추가 AWS CloudFormation 스택 옵션

다음과 같은 AWS CloudFormation 스택을 생성할 때 추가 옵션을 지정할 수 있습니다.

- 실패 시 롤백 방지
- 스택 정책을 설정하여 스택 업데이트 제어
- 종료 방지 기능 활성화

```
aws cloudformation create-stack --stack-name NewStack  
--template-body file://path/to/template.yml --on-failure DO_NOTHING
```

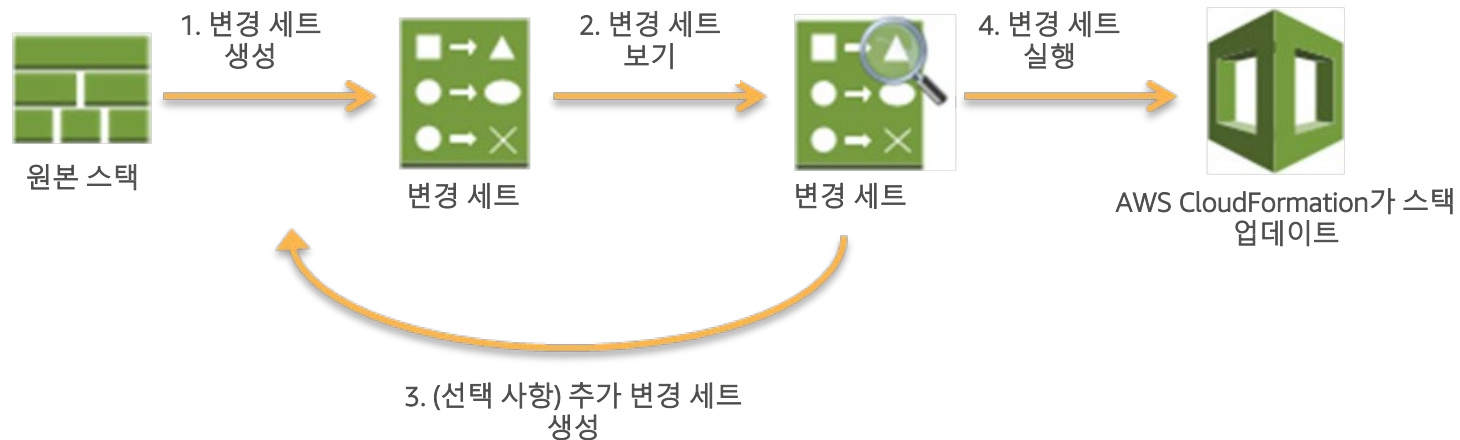

실패한 업데이트 롤백 재정의

- 롤백이 실패하고 **UPDATE_ROLLBACK_FAILED** 상태에 있어도 스택에 대한 업데이트를 계속 롤백합니다.
- 재정의 수행 –
 - 실패한 롤백의 원인을 해결합니다.
 - AWS CloudFormation에 업데이트 롤백을 계속하도록 지시합니다.

```
aws cloudformation continue-update-rollback --stack-name ExistingStack
```

변경 세트

- 변경 세트에 구현 전에 스택에 제안된 변경 사항이 나타납니다.
- AWS CloudFormation은 제출된 변경 사항을 스택과 비교합니다.
- 엔지니어는 변경 사항을 보고 어떤 리소스가 추가, 수정 또는 삭제되는지 확인할 수 있습니다.
- 그런 다음, 변경 세트를 스택에 적용하여 변경 사항을 구현할 수 있습니다.



AWS CloudFormation 모범 실무(1/3)

계획 및 정리: 수명 주기 및 소유권 기준으로 스택을 준비합니다.

템플릿을 재사용하여 여러 환경에서 스택을 복제합니다.

예제: 개발, 테스트 및 운영 환경

모든 리소스 유형의 제한을 확인합니다.

AWS CloudFormation 모범 실무(2/3)

템플릿 생성: 템플릿에 보안 인증 정보를 포함하지 않습니다.

AWS에 특정한 파라미터 유형을 사용합니다.

파라미터 제약 조건을 사용합니다.

AWS::CloudFormation::Init를 사용하여 EC2 인스턴스에 소프트웨어 애플리케이션을 배포합니다.

템플릿을 사용하기 전에 유효성을 검사합니다.

AWS CloudFormation 모범 실무(3/3)

스택 관리:

AWS CloudFormation을 통해 모든 스택 리소스를 관리합니다.

스택을 업데이트하기 전에 변경 세트를 생성합니다.

스택 정책을 사용합니다.

AWS CloudTrail을 사용하여 AWS CloudFormation 호출을 로그합니다.

코드 검토 및 수정 제어를 사용하여 템플릿을 관리합니다.

핵심 사항



- AWS CloudFormation을 사용하면 예측 가능하고 반복 가능한 방식으로 AWS 인프라 배포를 생성하고 프로비저닝할 수 있습니다.
- AWS CloudFormation의 2가지 주요 용어는 **템플릿**과 **스택**입니다.
- AWS CloudFormation 템플릿을 시작할 때 오류가 발생하면 모든 리소스가 기본적으로 롤백됩니다.
- 파라미터를 통해 스택을 생성하거나 업데이트할 때마다 템플릿에 사용자 지정 값을 입력할 수 있습니다.