**DevOps için en önemli AWS CloudFormation şablonları nelerdir?**

AWS CloudFormation, altyapıyı kod olarak yönetmeye ve otomatikleştirmeye yardımcı olan bir hizmettir. DevOps süreçlerini kolaylaştırmak için AWS CloudFormation kullanabilirsiniz. İşte DevOps için önemli AWS CloudFormation şablonlarından bazıları:

1. **VPC Oluşturma:**
   * AWS'nin Virtual Private Cloud (VPC) hizmeti, ağ altyapısını tanımlamanıza ve özelleştirmenize olanak tanır. Bir VPC oluşturma şablonu, kaynakları, alt ağları, yönlendirme tablolarını ve diğer ağ öğelerini tanımlar.
2. **EC2 Örnekleri İçin Otomatik Ölçekleme Grubu:**
   * Otomatik ölçeklendirme grupları, EC2 örneklerinin sayısını otomatik olarak ayarlamak için kullanılır. Bu, uygulamalarınıza esneklik ve yüksek erişilebilirlik sağlar.
3. **Yük Dengeleyici ve Auto Scaling Grubu İle Web Uygulaması:**
   * Web uygulaması için bir yük dengeleyici ve otomatik ölçeklendirme grubunu içeren bir şablon, uygulamanızın talep artışlarına yanıt verebilmesini sağlar.
4. **RDS Veritabanı Oluşturma:**
   * İlişkisel Veritabanı Hizmeti (RDS) kullanarak bir veritabanı oluşturan bir şablon, veritabanı kaynaklarını yönetmeyi kolaylaştırır.
5. **Lambda Fonksiyonu ve API Gateway:**
   * Serverless uygulamalar için AWS Lambda fonksiyonları ve API Gateway'i içeren bir şablon, olay tetikleyicilerini ve ilgili kaynakları hızla oluşturmanıza olanak tanır.
6. **S3 Depolama Alanı ve İzin Yönetimi:**
   * S3 (Simple Storage Service) kullanarak bir depolama alanı oluşturan ve erişim izinlerini yöneten bir şablon, veri depolama ve paylaşımını kolaylaştırır.
7. **IAM Roller ve İzinler:**
   * AWS Identity and Access Management (IAM) rollerlerini ve izinlerini tanımlayan bir şablon, güvenliği artırmanıza ve kullanıcı rollerini yönetmenize yardımcı olur.
8. **CloudWatch Alarmları ve Log Grupları:**
   * AWS CloudWatch kullanarak alarmlar ve log grupları oluşturan bir şablon, uygulama performansını izlemenize ve hata ayıklamanıza yardımcı olur.
9. **Route 53 Alan Adı ve DNS Ayarları:**
   * Domain Name System (DNS) yapılandırması ve Route 53 kullanarak bir alan adı oluşturan bir şablon, uygulamalarınızın internet erişimini yönetmenize yardımcı olur.

Bu şablonlar, temel AWS hizmetlerini ve bu hizmetler arasındaki ilişkileri otomatikleştirmek için kullanılabilir. Ancak, ihtiyaçlarınıza ve uygulamanıza bağlı olarak, daha spesifik ve karmaşık şablonlar oluşturmanız gerekebilir.

**AWS CloudFormation'da bir VPC örneği oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak basit bir VPC (Virtual Private Cloud) oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek şablon, VPC, bir ana alt ağ ve bir genel alt ağ içerir. Eksik bir AWS hesabınız varsa ve gerçek kaynakları oluşturmak istemiyorsanız, AWS Management Console'daki CloudFormation hizmetini kullanarak bu şablonu yükleyebilir ve test edebilirsiniz.

AWSTemplateFormatVersion: '2010-09-09'

Description: Basit bir VPC oluşturma örneği

Resources:

MyVPC:

Type: 'AWS::EC2::VPC'

Properties:

CidrBlock: '10.0.0.0/16'

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: MyVPC

PublicSubnet:

Type: 'AWS::EC2::Subnet'

Properties:

VpcId: !Ref MyVPC

CidrBlock: '10.0.0.0/24'

AvailabilityZone: 'us-east-1a'

MapPublicIpOnLaunch: true

Tags:

- Key: Name

Value: PublicSubnet

PrivateSubnet:

Type: 'AWS::EC2::Subnet'

Properties:

VpcId: !Ref MyVPC

CidrBlock: '10.0.1.0/24'

AvailabilityZone: 'us-east-1b'

Tags:

- Key: Name

Value: PrivateSubnet

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyVPC**: Ana VPC kaynağını temsil eder. CidrBlock, VPC'nin IP adres aralığını belirtir.
* **PublicSubnet** ve **PrivateSubnet**: Sırasıyla genel ve özel alt ağları temsil eder. CidrBlock, alt ağların IP adres aralıklarını belirtir. **MapPublicIpOnLaunch** özelliği, genel alt ağda başlatılan EC2 örneklerine otomatik olarak genel IP adresi atanmasını sağlar.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS Command Line Interface (CLI) üzerinden bir yığın (stack) oluşturabilir ve VPC'nizi inşa edebilirsiniz.

**AWS CloudFormation'da EC2 Örnekleri İçin Otomatik Ölçekleme Grubu oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak EC2 örnekleri için otomatik ölçekleme grubu oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, bir otomatik ölçekleme grubu ve bu grubun içindeki EC2 örnekleri için gerekli kaynakları tanımlar.

AWSTemplateFormatVersion: '2010-09-09'

Description: EC2 Örnekleri İçin Otomatik Ölçekleme Grubu

Resources:

MyLaunchConfiguration:

Type: 'AWS::AutoScaling::LaunchConfiguration'

Properties:

ImageId: 'ami-xxxxxxxxxxxxxxxxx' # Amazon Machine Image (AMI) ID

InstanceType: 't2.micro'

KeyName: 'your-key-pair' # EC2 Key Pair

UserData:

Fn::Base64: !Sub |

#!/bin/bash

# Your user data script here

MyAutoScalingGroup:

Type: 'AWS::AutoScaling::AutoScalingGroup'

Properties:

AvailabilityZones:

- 'us-east-1a'

- 'us-east-1b'

LaunchConfigurationName: !Ref MyLaunchConfiguration

MinSize: '2'

MaxSize: '5'

DesiredCapacity: '2'

MyScalingPolicy:

Type: 'AWS::AutoScaling::ScalingPolicy'

Properties:

AdjustmentType: 'ChangeInCapacity'

AutoScalingGroupName: !Ref MyAutoScalingGroup

Cooldown: '300'

ScalingAdjustment: '1'

MyCloudWatchAlarm:

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmDescription: 'Scale up when CPU exceeds 70%'

Namespace: 'AWS/EC2'

MetricName: 'CPUUtilization'

Dimensions:

- Name: 'AutoScalingGroupName'

Value: !Ref MyAutoScalingGroup

Statistic: 'Average'

Period: '300'

Threshold: '70'

ComparisonOperator: 'GreaterThanThreshold'

EvaluationPeriods: '2'

AlarmActions:

- !Ref MyScalingPolicy

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyLaunchConfiguration**: EC2 örneklerinin başlatma yapılandırmasını temsil eder. Bu, kullanılacak AMI, örnek türü, anahtar çifti ve kullanıcı verilerini içerir.
* **MyAutoScalingGroup**: Otomatik ölçekleme grubunu temsil eder. Bu, minimum ve maksimum ölçekleme boyutlarını, başlatma yapılandırmasını ve istenen kapasiteyi içerir.
* **MyScalingPolicy**: Otomatik ölçekleme grubunu ölçeklendirmek için bir politikayı temsil eder. Bu örnekte, CPU kullanımı %70'in üzerine çıktığında grup bir ölçü birimi (örneğin, bir EC2 örneği) ekler.
* **MyCloudWatchAlarm**: CloudWatch uyarısını temsil eder. Bu, ölçeklendirme politikasını tetikleyecek bir ölçüm (CPU kullanımı) ve eşik (70%) içerir.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve EC2 örnekleri için otomatik ölçekleme grubu oluşturabilirsiniz. Unutmayın ki gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir.

Formun Üstü

**AWS CloudFormation'da bir Yük Dengeleyici ve Auto Scaling Grubu İle Web Uygulaması oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak yük dengeleyici ve otomatik ölçekleme grubu ile bir web uygulaması oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, yük dengeleyici, otomatik ölçekleme grubu, EC2 başlatma yapılandırması ve diğer gerekli kaynakları içerir.

AWSTemplateFormatVersion: '2010-09-09'

Description: Yük Dengeleyici ve Otomatik Ölçekleme Grubu ile Web Uygulaması

Resources:

MyLaunchConfiguration:

Type: 'AWS::AutoScaling::LaunchConfiguration'

Properties:

ImageId: 'ami-xxxxxxxxxxxxxxxxx' # Amazon Machine Image (AMI) ID

InstanceType: 't2.micro'

KeyName: 'your-key-pair' # EC2 Key Pair

UserData:

Fn::Base64: !Sub |

#!/bin/bash

# Your user data script here

MyAutoScalingGroup:

Type: 'AWS::AutoScaling::AutoScalingGroup'

Properties:

AvailabilityZones:

- 'us-east-1a'

- 'us-east-1b'

LaunchConfigurationName: !Ref MyLaunchConfiguration

MinSize: '2'

MaxSize: '5'

DesiredCapacity: '2'

MyLoadBalancer:

Type: 'AWS::ElasticLoadBalancing::LoadBalancer'

Properties:

AvailabilityZones:

- 'us-east-1a'

- 'us-east-1b'

Listeners:

- LoadBalancerPort: '80'

InstancePort: '80'

Protocol: 'HTTP'

HealthCheck:

Target: 'HTTP:80/'

HealthyThreshold: '3'

UnhealthyThreshold: '5'

Interval: '30'

Timeout: '5'

Instances: !Ref MyAutoScalingGroup

MyScalingPolicy:

Type: 'AWS::AutoScaling::ScalingPolicy'

Properties:

AdjustmentType: 'ChangeInCapacity'

AutoScalingGroupName: !Ref MyAutoScalingGroup

Cooldown: '300'

ScalingAdjustment: '1'

MyCloudWatchAlarm:

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmDescription: 'Scale up when CPU exceeds 70%'

Namespace: 'AWS/EC2'

MetricName: 'CPUUtilization'

Dimensions:

- Name: 'AutoScalingGroupName'

Value: !Ref MyAutoScalingGroup

Statistic: 'Average'

Period: '300'

Threshold: '70'

ComparisonOperator: 'GreaterThanThreshold'

EvaluationPeriods: '2'

AlarmActions:

- !Ref MyScalingPolicy

Unit: 'Percent'

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyLaunchConfiguration**: EC2 örneklerinin başlatma yapılandırmasını temsil eder.
* **MyAutoScalingGroup**: Otomatik ölçekleme grubunu temsil eder.
* **MyLoadBalancer**: Yük dengeleyiciyi temsil eder ve web uygulamasının trafiğini ölçeklendirilmiş EC2 örneklerine yönlendirir.
* **MyScalingPolicy**: Otomatik ölçekleme grubunu ölçeklendirmek için bir politikayı temsil eder.
* **MyCloudWatchAlarm**: CloudWatch uyarısını temsil eder. Bu, ölçeklendirme politikasını tetikleyecek bir ölçüm (CPU kullanımı) ve eşik (70%) içerir.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve yük dengeleyici ve otomatik ölçekleme grubu ile bir web uygulaması oluşturabilirsiniz. Unutmayın ki gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir.

**AWS CloudFormation'da bir RDS Veritabanı Oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak bir RDS (Relational Database Service) veritabanı oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, bir RDS veritabanı örneği, veritabanı kullanıcısı ve diğer gerekli kaynakları içerir.

AWSTemplateFormatVersion: '2010-09-09'

Description: RDS Veritabanı Oluştur

Parameters:

DBName:

Type: String

Description: Veritabanı adı

Default: MyDB

DBUsername:

Type: String

Description: Veritabanı kullanıcı adı

Default: admin

DBPassword:

Type: String

NoEcho: true

Description: Veritabanı şifresi

Default: Password123

Resources:

MyDBInstance:

Type: 'AWS::RDS::DBInstance'

Properties:

Engine: 'mysql' # RDS veritabanı motoru

DBInstanceIdentifier: !Ref 'AWS::StackName'

MasterUsername: !Ref DBUsername

MasterUserPassword: !Ref DBPassword

AllocatedStorage: '20'

DBInstanceClass: 'db.t2.micro'

VPCSecurityGroups:

- !Ref MyDBSecurityGroup

AvailabilityZone: 'us-east-1a'

MyDBSecurityGroup:

Type: 'AWS::RDS::DBSecurityGroup'

Properties:

DBSecurityGroupIngress:

- EC2SecurityGroupName: 'default'

EC2SecurityGroupId: !GetAtt 'MyEC2Instance.SecurityGroups.0.GroupId'

Outputs:

EndpointAddress:

Description: RDS veritabanı son nokta adresi

Value: !GetAtt MyDBInstance.Endpoint.Address

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyDBInstance**: RDS veritabanı örneğini temsil eder. Bu örnekte MySQL veritabanı motorunu kullanıyoruz, ancak ihtiyacınıza göre değiştirebilirsiniz.
* **MyDBSecurityGroup**: RDS veritabanı için güvenlik grubunu temsil eder. Bu güvenlik grubu, veritabanına erişim izinlerini kontrol eder.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve RDS veritabanınızı oluşturabilirsiniz. Parametreler bölümünde, veritabanı adı, kullanıcı adı ve şifre gibi değişkenleri özelleştirebilirsiniz.

**AWS CloudFormation'da bir Lambda Fonksiyonu ve API Gateway oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak bir Lambda fonksiyonu ve API Gateway oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, bir Lambda fonksiyonu, API Gateway ve bu iki kaynağın arasındaki ilişkiyi içerir.

AWSTemplateFormatVersion: '2010-09-09'

Description: Lambda Fonksiyonu ve API Gateway Oluştur

Resources:

MyLambdaFunction:

Type: 'AWS::Lambda::Function'

Properties:

FunctionName: MyLambdaFunction

Handler: index.handler

Role: !GetAtt MyLambdaExecutionRole.Arn

FunctionName: MyLambdaFunction

Runtime: nodejs14.x

Timeout: 30

Code:

S3Bucket: your-s3-bucket

S3Key: your-lambda-code.zip

MyLambdaExecutionRole:

Type: 'AWS::IAM::Role'

Properties:

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Principal:

Service: lambda.amazonaws.com

Action: 'sts:AssumeRole'

Policies:

- PolicyName: LambdaExecutionPolicy

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Action:

- 'logs:CreateLogGroup'

- 'logs:CreateLogStream'

- 'logs:PutLogEvents'

Resource: 'arn:aws:logs:\*:\*:\*'

MyApiGateway:

Type: 'AWS::ApiGateway::RestApi'

Properties:

Name: MyApi

MyApiGatewayResource:

Type: 'AWS::ApiGateway::Resource'

Properties:

RestApiId: !Ref MyApiGateway

ParentId: !GetAtt MyApiGateway.RootResourceId

PathPart: 'myresource'

MyApiGatewayMethod:

Type: 'AWS::ApiGateway::Method'

Properties:

AuthorizationType: NONE

HttpMethod: POST

ResourceId: !Ref MyApiGatewayResource

RestApiId: !Ref MyApiGateway

Integration:

IntegrationHttpMethod: POST

Type: AWS\_PROXY

Uri: !Sub arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${MyLambdaFunction.Arn}/invocations

Outputs:

ApiEndpoint:

Description: API Gateway Endpoint

Value: !Sub 'https://${MyApiGateway}.execute-api.${AWS::Region}.amazonaws.com'

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyLambdaFunction**: Lambda fonksiyonunu temsil eder. Bu örnekte, Node.js 14.x sürümü kullanılıyor ancak ihtiyacınıza göre değiştirebilirsiniz. Lambda fonksiyonunun kodu S3'den alınır.
* **MyLambdaExecutionRole**: Lambda fonksiyonunun çalışma izinlerini temsil eder. Bu rol, CloudWatch Logs'a yazma yetkilerini içerir.
* **MyApiGateway**: API Gateway'yi temsil eder.
* **MyApiGatewayResource**: API Gateway kaynağını temsil eder.
* **MyApiGatewayMethod**: API Gateway yöntemini temsil eder. Bu, POST isteklerini Lambda fonksiyonuna yönlendiren bir entegrasyon içerir.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve Lambda fonksiyonu ile birlikte API Gateway'ı oluşturabilirsiniz. Unutmayın ki gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir.

**AWS CloudFormation'da bir S3 Depolama Alanı ve İzin Yönetimi oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak bir S3 depolama alanı (bucket) ve izin yönetimi oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, bir S3 depolama alanı, erişim kontrol listesi (ACL) ve bucket politikası içerir.

AWSTemplateFormatVersion: '2010-09-09'

Description: S3 Depolama Alanı ve İzin Yönetimi

Resources:

MyS3Bucket:

Type: 'AWS::S3::Bucket'

Properties:

BucketName: my-s3-bucket

AccessControl: Private

Tags:

- Key: Name

Value: MyS3Bucket

MyBucketPolicy:

Type: 'AWS::S3::BucketPolicy'

Properties:

Bucket: !Ref MyS3Bucket

PolicyDocument:

Statement:

- Effect: Allow

Principal: '\*'

Action: 's3:GetObject'

Resource: !Sub 'arn:aws:s3:::${MyS3Bucket}/\*'

Outputs:

S3BucketName:

Description: S3 Depolama Alanı Adı

Value: !Ref MyS3Bucket

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyS3Bucket**: S3 depolama alanını temsil eder. Bu örnekte, depolama alanının adını, erişim kontrol listesini (Private olarak ayarlanmıştır) ve etiketlerini belirtiyoruz.
* **MyBucketPolicy**: S3 depolama alanı için bir bucket politikasını temsil eder. Bu politika, tüm kullanıcılara (**Principal: '\*'**) depolama alanındaki nesneleri alabilme yetkisi verir.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve S3 depolama alanınızı oluşturabilirsiniz. Unutmayın ki gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir. Özellikle, güvenlik gereksinimlerinize ve uygulama tasarımınıza uygun bir erişim kontrol politikası belirlemeniz önemlidir.

AWS CloudFormation'da bir IAM Roller ve İzinleri oluştur.

Aşağıda, AWS CloudFormation şablonu kullanarak bir IAM rolü ve izinlerini oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, bir IAM rolü, bir izin politikası ve kullanıcılara rol atamasını içerir.

AWSTemplateFormatVersion: '2010-09-09'

Description: IAM Rolü ve İzinleri

Resources:

MyIAMRole:

Type: 'AWS::IAM::Role'

Properties:

RoleName: MyIAMRole

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Principal:

Service: lambda.amazonaws.com

Action: 'sts:AssumeRole'

MyIAMRolePolicy:

Type: 'AWS::IAM::Policy'

Properties:

PolicyName: MyIAMRolePolicy

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Action:

- 's3:GetObject'

- 's3:PutObject'

Resource: 'arn:aws:s3:::my-s3-bucket/\*' # S3 bucket izinleri

Roles:

- !Ref MyIAMRole

Outputs:

IAMRoleArn:

Description: IAM Rolü ARN

Value: !GetAtt MyIAMRole.Arn

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyIAMRole**: IAM rolünü temsil eder. Bu örnekte, Lambda fonksiyonları tarafından kullanılan bir rol oluşturuyoruz. **AssumeRolePolicyDocument** alanında, bu rola erişim izni verilen AWS hizmetini belirtiyoruz.
* **MyIAMRolePolicy**: IAM rolüne eklenen bir izin politikasını temsil eder. Bu örnekte, S3 depolama alanına erişim izni ekliyoruz.
* **IAMRoleArn**: Oluşturulan IAM rolünün ARN'ini gösteren bir çıktı.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve IAM rolünüzü ve ilişkili izinleri oluşturabilirsiniz. Unutmayın ki gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir. Özellikle, IAM rolleri ve izin politikaları, güvenlik ve uyumluluk gereksinimlerinize uyacak şekilde dikkatlice tasarlanmalıdır.

**AWS CloudFormation'da bir CloudWatch Alarmları ve Log Grupları oluştur.**

Aşağıda, AWS CloudFormation şablonu kullanarak bir CloudWatch Alarmları ve Log Grupları oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnek, bir CloudWatch Log Grubu ve bu log grubu üzerinde bir CloudWatch Alarm'ı içerir.

AWSTemplateFormatVersion: '2010-09-09'

Description: CloudWatch Alarmları ve Log Grupları

Resources:

MyLogGroup:

Type: 'AWS::Logs::LogGroup'

Properties:

LogGroupName: MyLogGroup

MyCloudWatchAlarm:

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmName: MyCloudWatchAlarm

ComparisonOperator: GreaterThanThreshold

EvaluationPeriods: 1

MetricName: 'IncomingLogEvents'

Namespace: 'AWS/Logs'

Period: 60

Statistic: Sum

Threshold: 10

ActionsEnabled: true

AlarmActions:

- !Ref MySnsTopic

Dimensions:

- Name: LogGroupName

Value: !Ref MyLogGroup

TreatMissingData: notBreaching

MySnsTopic:

Type: 'AWS::SNS::Topic'

Properties:

DisplayName: 'MySnsTopic'

Outputs:

LogGroupName:

Description: CloudWatch Log Grubu Adı

Value: !Ref MyLogGroup

CloudWatchAlarmName:

Description: CloudWatch Alarm Adı

Value: !Ref MyCloudWatchAlarm

SnsTopicArn:

Description: SNS Topic ARN

Value: !Ref MySnsTopic

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyLogGroup**: CloudWatch Log Grubunu temsil eder. Bu örnekte, bir log grubu oluşturuyoruz.
* **MyCloudWatchAlarm**: CloudWatch Alarm'ını temsil eder. Bu örnekte, belirli bir log grubu üzerindeki gelen log olaylarına dayalı bir alarm oluşturuyoruz. Alarm, 60 saniyelik bir değerlendirme süresi içinde 10'dan fazla gelen log olayını algıladığında tetiklenecek şekilde ayarlanmıştır.
* **MySnsTopic**: SNS (Simple Notification Service) konusunu temsil eder. Alarm durumu değiştiğinde, bu konu üzerindeki abonelere bir bildirim göndermek için kullanılır.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve CloudWatch Log Grubu ve Alarm'ınızı oluşturabilirsiniz. Unutmayın ki gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir.

AWS CloudFormation'da bir Route 53 Alan Adı ve DNS Ayarları oluştur.

Aşağıda, AWS CloudFormation şablonu kullanarak bir Route 53 alan adı ve DNS ayarlarını oluşturmanın örnek bir YAML kodu bulunmaktadır. Bu örnekte, bir alan adı ve bu alan adına bağlı bir kayıt kümesi (record set) oluşturuyoruz.

AWSTemplateFormatVersion: '2010-09-09'

Description: Route 53 Alan Adı ve DNS Ayarları

Parameters:

DomainName:

Type: String

Description: "Alan adı"

Default: mydomain.com

Resources:

MyHostedZone:

Type: 'AWS::Route53::HostedZone'

Properties:

Name: !Ref DomainName

MyRecordSet:

Type: 'AWS::Route53::RecordSet'

Properties:

HostedZoneId: !Ref MyHostedZone

Name: !Ref DomainName

Type: A

TTL: '300'

ResourceRecords:

- '1.2.3.4' # Bu, alan adınızın yönlendirileceği IP adresini temsil eder.

Outputs:

HostedZoneId:

Description: Route 53 Hosted Zone ID

Value: !Ref MyHostedZone

Bu şablon aşağıdaki temel öğeleri içerir:

* **MyHostedZone**: Route 53 hosted zone'nu temsil eder. Bu, alan adınızın yönetildiği yerdir.
* **MyRecordSet**: Route 53 record set'i temsil eder. Bu örnekte, bir "A" tipi kayıt kümesi oluşturuyoruz ve bu kayıt kümesi, belirtilen IP adresine yönlendirilmiş bir alan adını temsil eder.

Bu şablonu kullanarak CloudFormation konsolundan veya AWS CLI üzerinden bir yığın (stack) oluşturabilir ve Route 53 alan adınızı ve DNS ayarlarınızı oluşturabilirsiniz. "DomainName" parametresini belirleyerek farklı alan adları kullanabilirsiniz. Gerçek kullanım senaryolarınıza uyacak şekilde özelleştirmeler yapmanız gerekebilir.