Deploying an EC2 Instance Using AWS CloudFormation and AWS CLI

Overview

This document provides step-by-step instructions to launch an Amazon EC2 instance using an AWS CloudFormation template via the AWS CLI.

Prerequisites

- AWS CLI installed and configured.
- An AWS account with necessary IAM permissions to create EC2 instances and CloudFormation stacks.
- A valid CloudFormation YAML template defining the EC2 instance.

Step 1: Create the CloudFormation YAML Template

Create a YAML file (awslinuxid.yaml) with the following content to launch an Amazon EC2 instance using a predefined Amazon Linux AMI:

Users > Harshitha Basavaraju > aws-linux > ! awslinuxid.yaml

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Create an EC2 instance with a Security Group

Resources:
  # Security Group
  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow SSH and HTTP access
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0  # Allow SSH from anywhere (Modify for security)
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0  # Allow HTTP from anywhere
      Tags:
        - Key: Name
          Value: MySecurityGroup

  # EC2 Instance
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-087f352c165340ea1 # Replace with a valid AMI ID
      InstanceType: t2.micro
      SecurityGroupIds:
        - !Ref MySecurityGroup
      KeyName: myec2instance  # Replace with your EC2 key pair
      Tags:
        - Key: Name
          Value: MyEC2Instance
```

Step 2: Save the Template Locally

Save the YAML file in a directory on your system

Step 3: Configure AWS CLI

Run the following command to configure AWS CLI with your credentials: **aws configure**

Enter your:

- AWS Access Key ID
- AWS Secret Access Key
- Default region (e.g., us-west-2)
- Default output format (json or text)

Step 4: Verify AWS CLI Connection

Run the following command to verify your AWS identity**: aws sts get-caller-identity**

This should return details about your AWS IAM user or role.

```
C:\Users\Harshitha Basavaraju>aws sts get-caller-identity
Account: '949847155882'
Arn: arn:aws:iam::949847155882:user/Harshitha
UserId: AIDA52J2NMSVOQWMYRXE7
```

Step 5: Deploy the CloudFormation Stack

Use the following command to create the CloudFormation stack:

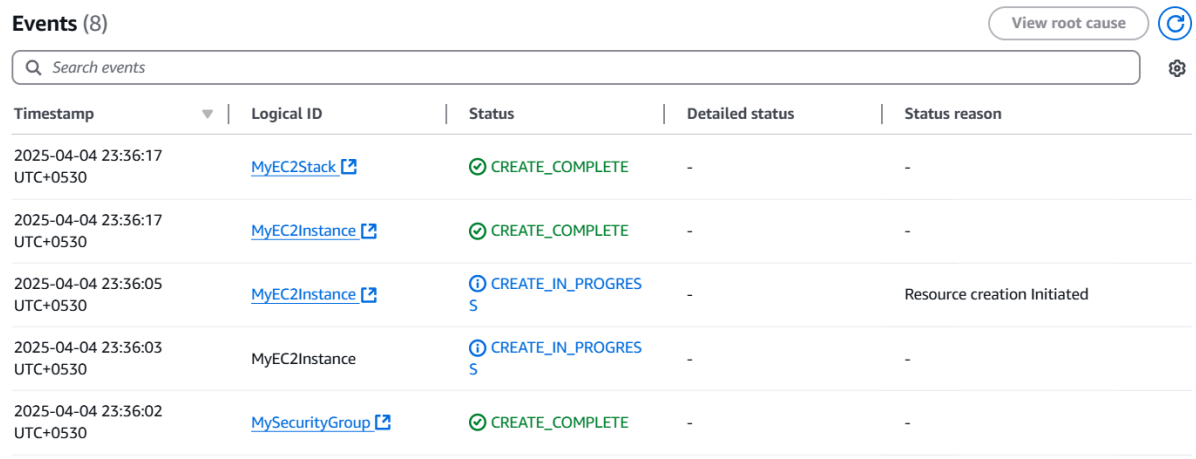
aws cloudformation create-stack –stack-name MyEC2Stack –template-body file://"C:/Users/Filepath" –capabilities CAPABILITY_NAMED_IAM

```
C:\Users\Harshitha Basavaraju>aws cloudformation create-stack --stack-name MyEC2Stack --template-body file://"C:\Users\Harshitha Basavaraju\aws-linux\awslin
uxid.yaml" --capabilities CAPABILITY_NAMED_IAM
StackId: arn:aws:cloudformation:us-west-2:949847155882:stack/MyEC2Stack/71e6b830-117f-11f0-aab7-0a85c3e2c5b7
```

Possible statuses:

- CREATE_IN_PROGRESS: Stack is being created.
- CREATE_COMPLETE: Stack has been successfully deployed.
- ROLLBACK_IN_PROGRESS: Stack creation failed, and AWS is rolling back changes

By following these steps, you have successfully deployed an EC2 instance using AWS CloudFormation via AWS CLI.



Step 6 : Verification through command line

RUN: **aws cloudformation describe-stacks --stack-name MyEC2Stack --query"Stacks[0].StackStatus"**

```
C:\Users\Harshitha Basavaraju>aws cloudformation describe-stacks --stack-name MyEC2Stack --query "Stacks[0].StackStatus"
"CREATE_COMPLETE"
```