Project

Setup a three stages pipeline to deploy a given Application using CloudFormation as a deploy provider.

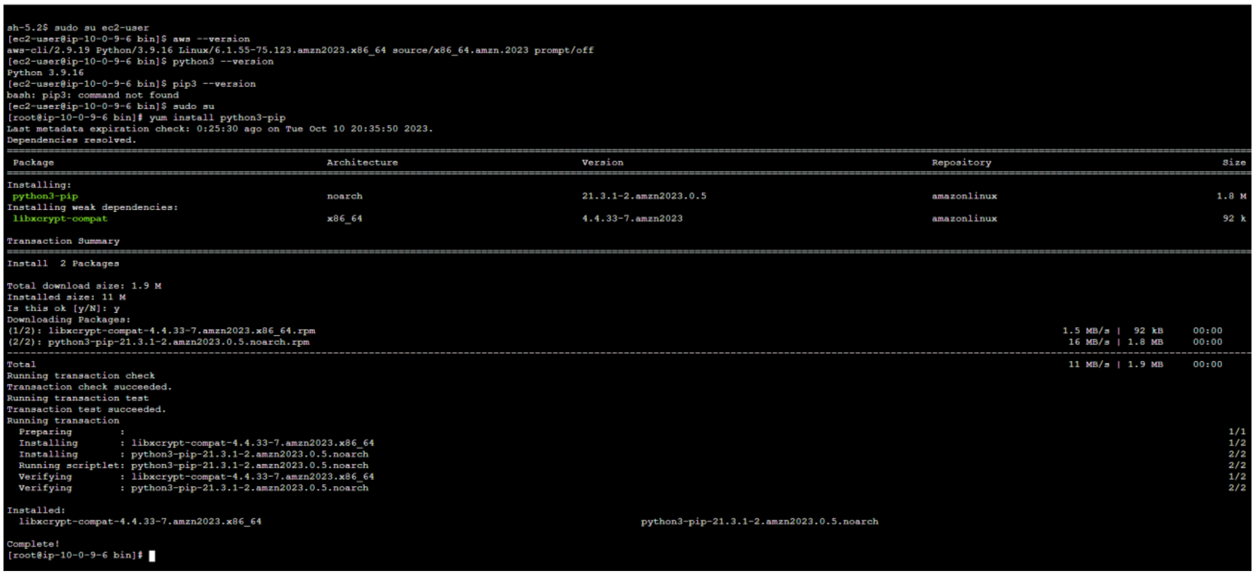
By

Aghogho Dre Aruoture

Best Practice as a DevOps engineer is to **always test your tools and codes manually before any automation.**

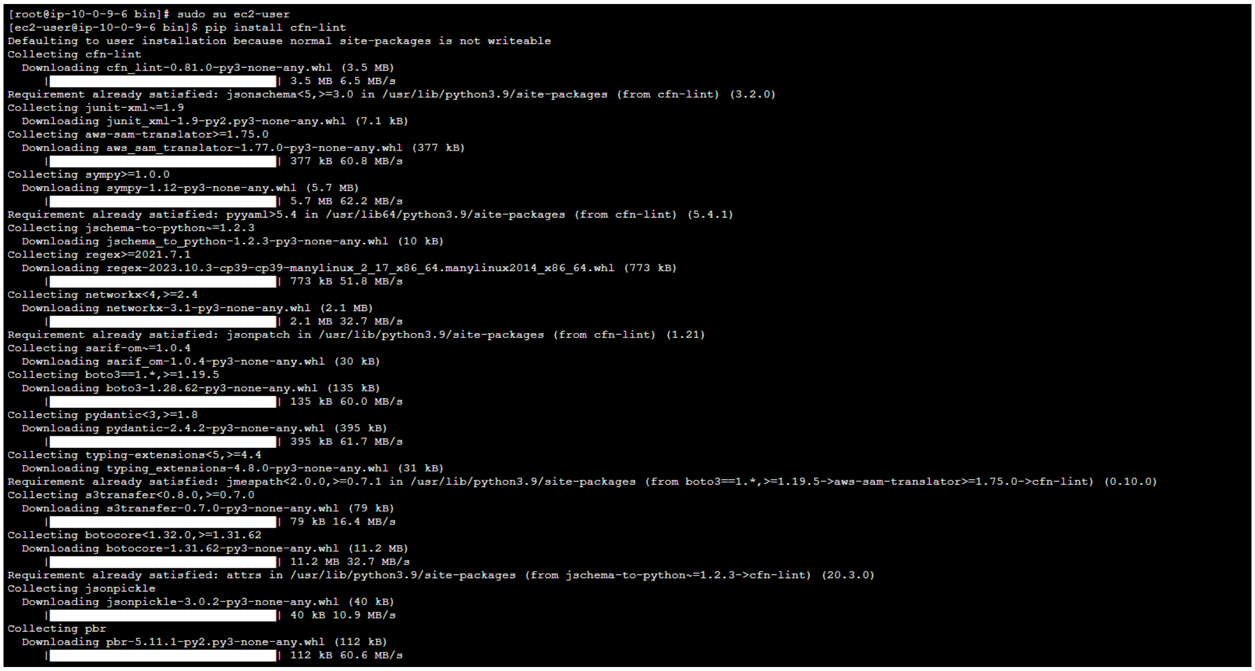
In this project, I used cfn-lint testing tool to integrate as part of my DevOps testing tool. For me to be sure cfn-lint tool works fine, I had to first of all, test it locally. (You can test it locally from your PC if your PC has the tools, you need or can just simply create an EC2 instance, ssh into it and test it there). In this case, I will be launching an EC2 instance.

It should be noted that cfn-lint needs python and pip3 installed. More info about cfn-lint can be found in our grand father google 😉. (https://github.com/aws-cloudformation/cfn-lint)

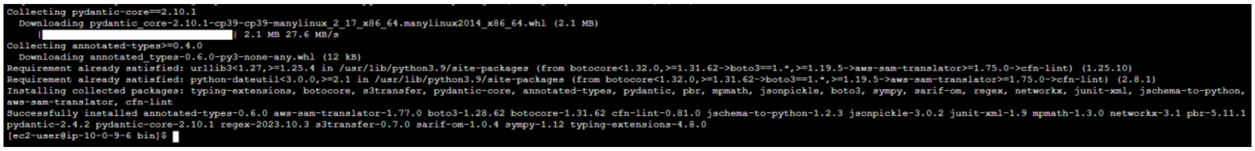


```
sh-5.2$ sudo su ec2-user
[ec2-user@ip-10-0-9-6 bin]$ aws --version
aws-cli/2.9.19 Python/3.9.16 Linux/6.1.55-75.123.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
[ec2-user@ip-10-0-9-6 bin]$ python3 --version
Python 3.9.16
[ec2-user@ip-10-0-9-6 bin]$ pip3 --version
bash: pip3: command not found
[ec2-user@ip-10-0-9-6 bin]$ sudo su
[root@ip-10-0-9-6 bin]# yum install python3-pip
Last metadata expiration check: 0:25:30 ago on Tue Oct 10 20:35:50 2023.
Dependencies resolved.
==================================================================================================================================================================================================
 Package                                          Architecture                            Version                                                Repository                                   Size
==================================================================================================================================================================================================
Installing:
 python3-pip                                      noarch                                  21.3.1-2.amzn2023.0.5                                  amazonlinux                                 1.8 M
Installing weak dependencies:
 libxcrypt-compat                                 x86_64                                  4.4.33-7.amzn2023                                      amazonlinux                                  92 k

Transaction Summary
==================================================================================================================================================================================================
Install  2 Packages

Total download size: 1.9 M
Installed size: 11 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): libxcrypt-compat-4.4.33-7.amzn2023.x86_64.rpm                                                                                                 1.5 MB/s |  92 kB     00:00
(2/2): python3-pip-21.3.1-2.amzn2023.0.5.noarch.rpm                                                                                                   16 MB/s | 1.8 MB     00:00
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                                                                                 11 MB/s | 1.9 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                                                                                           1/1
  Installing       : libxcrypt-compat-4.4.33-7.amzn2023.x86_64                                                                                                                                 1/2
  Installing       : python3-pip-21.3.1-2.amzn2023.0.5.noarch                                                                                                                                  2/2
  Running scriptlet: python3-pip-21.3.1-2.amzn2023.0.5.noarch                                                                                                                                  2/2
  Verifying        : libxcrypt-compat-4.4.33-7.amzn2023.x86_64                                                                                                                                 1/2
  Verifying        : python3-pip-21.3.1-2.amzn2023.0.5.noarch                                                                                                                                  2/2

Installed:
  libxcrypt-compat-4.4.33-7.amzn2023.x86_64                                                        python3-pip-21.3.1-2.amzn2023.0.5.noarch

Complete!
[root@ip-10-0-9-6 bin]#
```

To install pip, run the cmd pip install cfn-lint

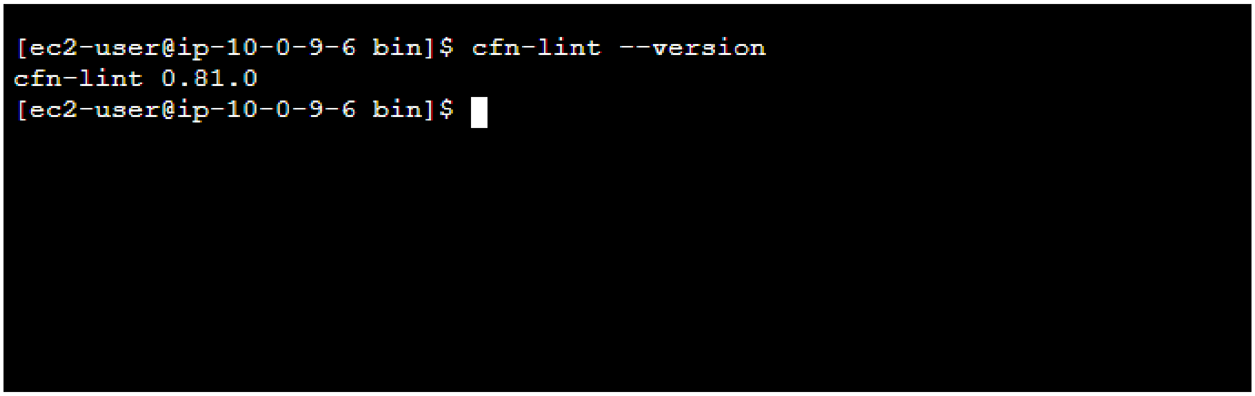


```
[root@ip-10-0-9-6 bin]# sudo su ec2-user
[ec2-user@ip-10-0-9-6 bin]$ pip install cfn-lint
Defaulting to user installation because normal site-packages is not writeable
Collecting cfn-lint
  Downloading cfn_lint-0.81.0-py3-none-any.whl (3.5 MB)
     |████████████████████████████████| 3.5 MB 6.5 MB/s
Requirement already satisfied: jsonschema<5,>=3.0 in /usr/lib/python3.9/site-packages (from cfn-lint) (3.2.0)
Collecting junit-xml~=1.9
  Downloading junit_xml-1.9-py2.py3-none-any.whl (7.1 kB)
Collecting aws-sam-translator>=1.75.0
  Downloading aws_sam_translator-1.77.0-py3-none-any.whl (377 kB)
     |████████████████████████████████| 377 kB 60.8 MB/s
Collecting sympy>=1.0.0
  Downloading sympy-1.12-py3-none-any.whl (5.7 MB)
     |████████████████████████████████| 5.7 MB 62.2 MB/s
Requirement already satisfied: pyyaml>5.4 in /usr/lib64/python3.9/site-packages (from cfn-lint) (5.4.1)
Collecting jschema-to-python~=1.2.3
  Downloading jschema_to_python-1.2.3-py3-none-any.whl (10 kB)
Collecting regex>=2021.7.1
  Downloading regex-2023.10.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (773 kB)
     |████████████████████████████████| 773 kB 51.8 MB/s
Collecting networkx<4,>=2.4
  Downloading networkx-3.1-py3-none-any.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 32.7 MB/s
Requirement already satisfied: jsonpatch in /usr/lib/python3.9/site-packages (from cfn-lint) (1.21)
Collecting sarif-om~=1.0.4
  Downloading sarif_om-1.0.4-py3-none-any.whl (30 kB)
Collecting boto3==1.*,>=1.19.5
  Downloading boto3-1.28.62-py3-none-any.whl (135 kB)
     |████████████████████████████████| 135 kB 60.0 MB/s
Collecting pydantic<3,>=1.8
  Downloading pydantic-2.4.2-py3-none-any.whl (395 kB)
     |████████████████████████████████| 395 kB 61.7 MB/s
Collecting typing-extensions<5,>=4.4
  Downloading typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/lib/python3.9/site-packages (from boto3==1.*,>=1.19.5->aws-sam-translator>=1.75.0->cfn-lint) (0.10.0)
Collecting s3transfer<0.8.0,>=0.7.0
  Downloading s3transfer-0.7.0-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 16.4 MB/s
Collecting botocore<1.32.0,>=1.31.62
  Downloading botocore-1.31.62-py3-none-any.whl (11.2 MB)
     |████████████████████████████████| 11.2 MB 32.7 MB/s
Requirement already satisfied: attrs in /usr/lib/python3.9/site-packages (from jschema-to-python~=1.2.3->cfn-lint) (20.3.0)
Collecting jsonpickle
  Downloading jsonpickle-3.0.2-py3-none-any.whl (40 kB)
     |████████████████████████████████| 40 kB 10.9 MB/s
Collecting pbr
  Downloading pbr-5.11.1-py2.py3-none-any.whl (112 kB)
     |████████████████████████████████| 112 kB 60.6 MB/s
```



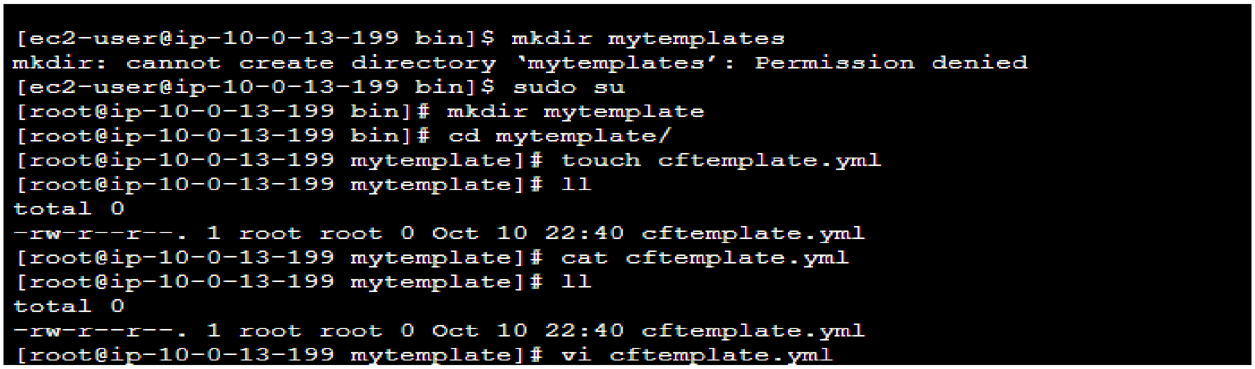
```
Collecting pydantic-core==2.10.1
  Downloading pydantic_core-2.10.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 27.6 MB/s
Collecting annotated-types>=0.4.0
  Downloading annotated_types-0.6.0-py3-none-any.whl (12 kB)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/lib/python3.9/site-packages (from botocore<1.32.0,>=1.31.62->boto3==1.*,>=1.19.5->aws-sam-translator>=1.75.0->cfn-lint) (1.25.10)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python3.9/site-packages (from botocore<1.32.0,>=1.31.62->boto3==1.*,>=1.19.5->aws-sam-translator>=1.75.0->cfn-lint) (2.8.1)
Installing collected packages: typing-extensions, botocore, s3transfer, pydantic-core, annotated-types, pydantic, pbr, mpmath, jsonpickle, boto3, sympy, sarif-om, regex, networkx, junit-xml, jschema-to-python,
aws-sam-translator, cfn-lint
Successfully installed annotated-types-0.6.0 aws-sam-translator-1.77.0 boto3-1.28.62 botocore-1.31.62 cfn-lint-0.81.0 jschema-to-python-1.2.3 jsonpickle-3.0.2 junit-xml-1.9 mpmath-1.3.0 networkx-3.1 pbr-5.11.1
pydantic-2.4.2 pydantic-core-2.10.1 regex-2023.10.3 s3transfer-0.7.0 sarif-om-1.0.4 sympy-1.12 typing-extensions-4.8.0
[ec2-user@ip-10-0-9-6 bin]$ █
```

The cmd cfn-lint –version is used to verify if cfn-lint was succesfully installed.

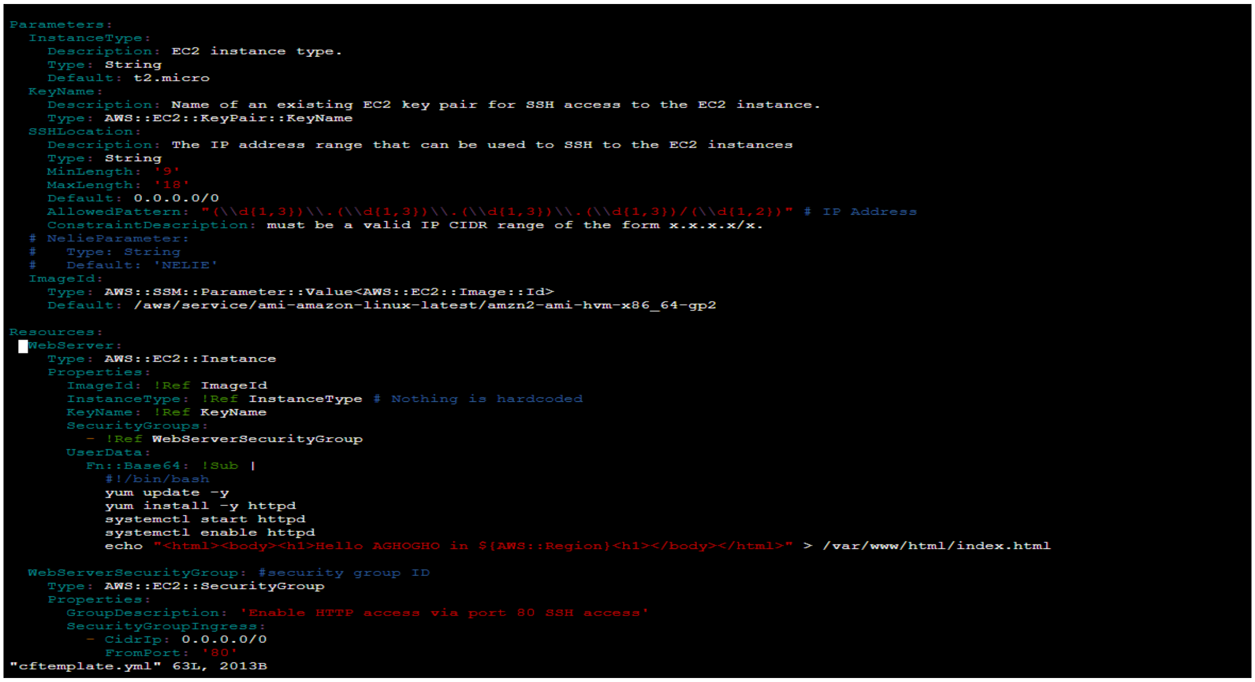


```
[ec2-user@ip-10-0-9-6 bin]$ cfn-lint --version
cfn-lint 0.81.0
[ec2-user@ip-10-0-9-6 bin]$ █
```

As a root user, I made a folder and added my cloudformation template with the cmd vi cftemplate.yml



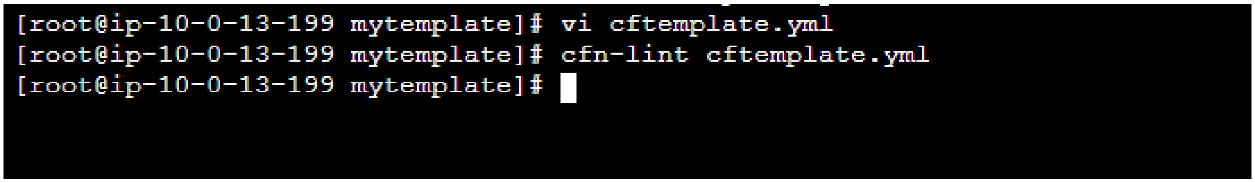
```
[ec2-user@ip-10-0-13-199 bin]$ mkdir mytemplates
mkdir: cannot create directory 'mytemplates': Permission denied
[ec2-user@ip-10-0-13-199 bin]$ sudo su
[root@ip-10-0-13-199 bin]# mkdir mytemplate
[root@ip-10-0-13-199 bin]# cd mytemplate/
[root@ip-10-0-13-199 mytemplate]# touch cftemplate.yml
[root@ip-10-0-13-199 mytemplate]# ll
total 0
-rw-r--r--. 1 root root 0 Oct 10 22:40 cftemplate.yml
[root@ip-10-0-13-199 mytemplate]# cat cftemplate.yml
[root@ip-10-0-13-199 mytemplate]# ll
total 0
-rw-r--r--. 1 root root 0 Oct 10 22:40 cftemplate.yml
[root@ip-10-0-13-199 mytemplate]# vi cftemplate.yml
```



```
Parameters:
  InstanceType:
    Description: EC2 instance type.
    Type: String
    Default: t2.micro
  KeyName:
    Description: Name of an existing EC2 key pair for SSH access to the EC2 instance.
    Type: AWS::EC2::KeyPair::KeyName
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})" # IP Address
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  # NelieParameter:
  #   Type: String
  #   Default: 'NELIE'
  ImageId:
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  WebServer:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref ImageId
      InstanceType: !Ref InstanceType # Nothing is hardcoded
      KeyName: !Ref KeyName
      SecurityGroups:
        - !Ref WebServerSecurityGroup
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          systemctl start httpd
          systemctl enable httpd
          echo "<html><body><h1>Hello AGHOGHO in ${AWS::Region}<h1></body></html>" > /var/www/html/index.html

  WebServerSecurityGroup: #security group ID
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: 'Enable HTTP access via port 80 SSH access'
      SecurityGroupIngress:
        - CidrIp: 0.0.0.0/0
          FromPort: '80'
"cftemplate.yml" 63L, 2013B
```

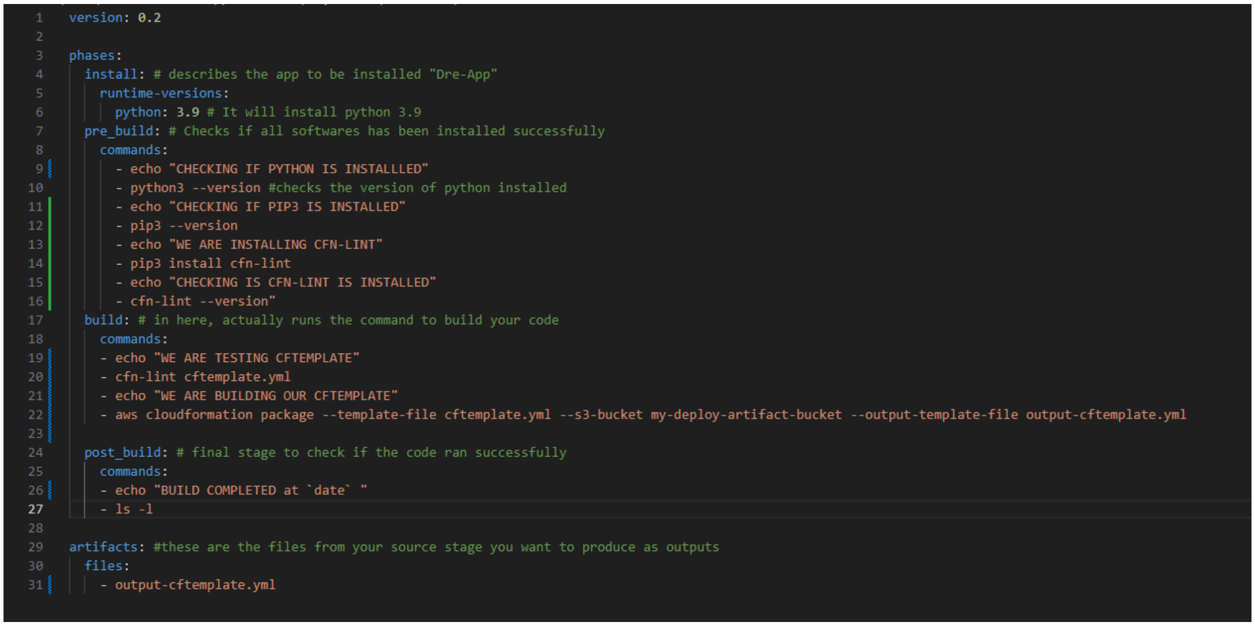
To test the cftemplate.yml using the cfn-lint tool we just installed, we run the cmd cfn-lint cftemplate.yml

If there is no error response, it means the template is good without errors as seen below.



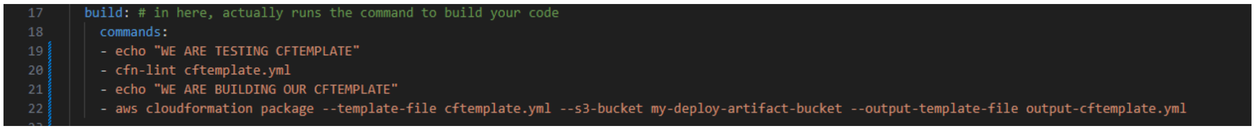
```
[root@ip-10-0-13-199 mytemplate]# vi cftemplate.yml
[root@ip-10-0-13-199 mytemplate]# cfn-lint cftemplate.yml
[root@ip-10-0-13-199 mytemplate]# 
```

Now that I have tested my testing tool “cfn-lint” locally, I can now proceed to the automation. So, the next step is to write my buildspec.yml



```yaml
version: 0.2

phases:
  install: # describes the app to be installed "Dre-App"
    runtime-versions:
      python: 3.9 # It will install python 3.9
  pre_build: # Checks if all softwares has been installed successfully
    commands:
      - echo "CHECKING IF PYTHON IS INSTALLLED"
      - python3 --version #checks the version of python installed
      - echo "CHECKING IF PIP3 IS INSTALLED"
      - pip3 --version
      - echo "WE ARE INSTALLING CFN-LINT"
      - pip3 install cfn-lint
      - echo "CHECKING IS CFN-LINT IS INSTALLED"
      - cfn-lint --version"
  build: # in here, actually runs the command to build your code
    commands:
    - echo "WE ARE TESTING CFTEMPLATE"
    - cfn-lint cftemplate.yml
    - echo "WE ARE BUILDING OUR CFTEMPLATE"
    - aws cloudformation package --template-file cftemplate.yml --s3-bucket my-deploy-artifact-bucket --output-template-file output-cftemplate.yml

  post_build: # final stage to check if the code ran successfully
    commands:
    - echo "BUILD COMPLETED at `date` "
    - ls -l

artifacts: #these are the files from your source stage you want to produce as outputs
  files:
    - output-cftemplate.yml
```

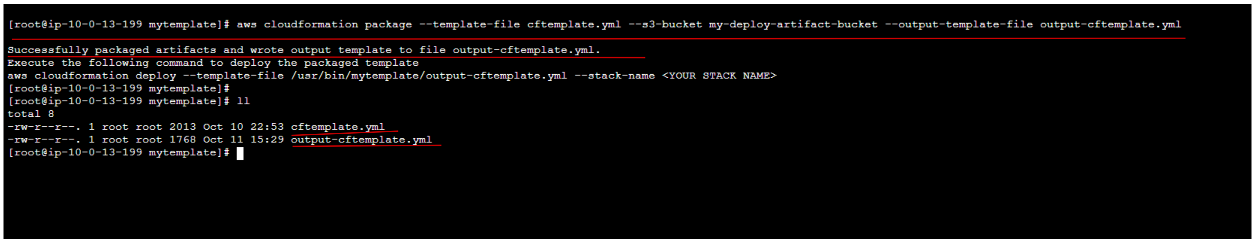
As seen on the build section;



```yaml
  build: # in here, actually runs the command to build your code
    commands:
    - echo "WE ARE TESTING CFTEMPLATE"
    - cfn-lint cftemplate.yml
    - echo "WE ARE BUILDING OUR CFTEMPLATE"
    - aws cloudformation package --template-file cftemplate.yml --s3-bucket my-deploy-artifact-bucket --output-template-file output-cftemplate.yml
```

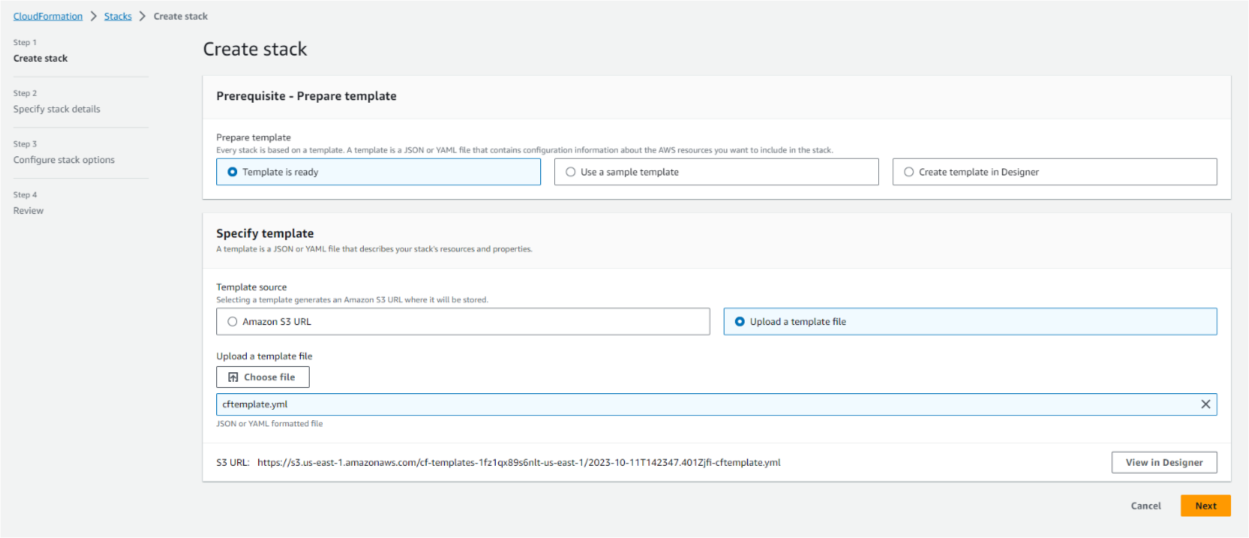
The build stage will **test** the cftemplate with the cfn-lint and will **build** the cftemplate with the cmd aws cloudformation package. Aws cloudformation package will build, clean-up the cftemplate and package it as an artifact and store on an s3-bucket u define and to your local file. (I created an S3 bucket “my-deploy-artifact-bucket).

As best practice, It should be noted to test the cftemplate manually on the aws console before automating its testing with cfn-lin. Also, before adding the aws cloudformation package tool, its best practice to test it locally like I did from my ec2 virtual server below.



```
[root@ip-10-0-13-199 mytemplate]# aws cloudformation package --template-file cftemplate.yml --s3-bucket my-deploy-artifact-bucket --output-template-file output-cftemplate.yml

Successfully packaged artifacts and wrote output template to file output-cftemplate.yml.
Execute the following command to deploy the packaged template
aws cloudformation deploy --template-file /usr/bin/mytemplate/output-cftemplate.yml --stack-name <YOUR STACK NAME>
[root@ip-10-0-13-199 mytemplate]#
[root@ip-10-0-13-199 mytemplate]# ll
total 8
-rw-r--r--. 1 root root 2013 Oct 10 22:53 cftemplate.yml
-rw-r--r--. 1 root root 1768 Oct 11 15:29 output-cftemplate.yml
[root@ip-10-0-13-199 mytemplate]#
```

Testing the cftemplate manually on the aws console, I just simply deployed the it on CloudFormation in the console as shown below.



Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

● Template is ready ○ Use a sample template ○ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

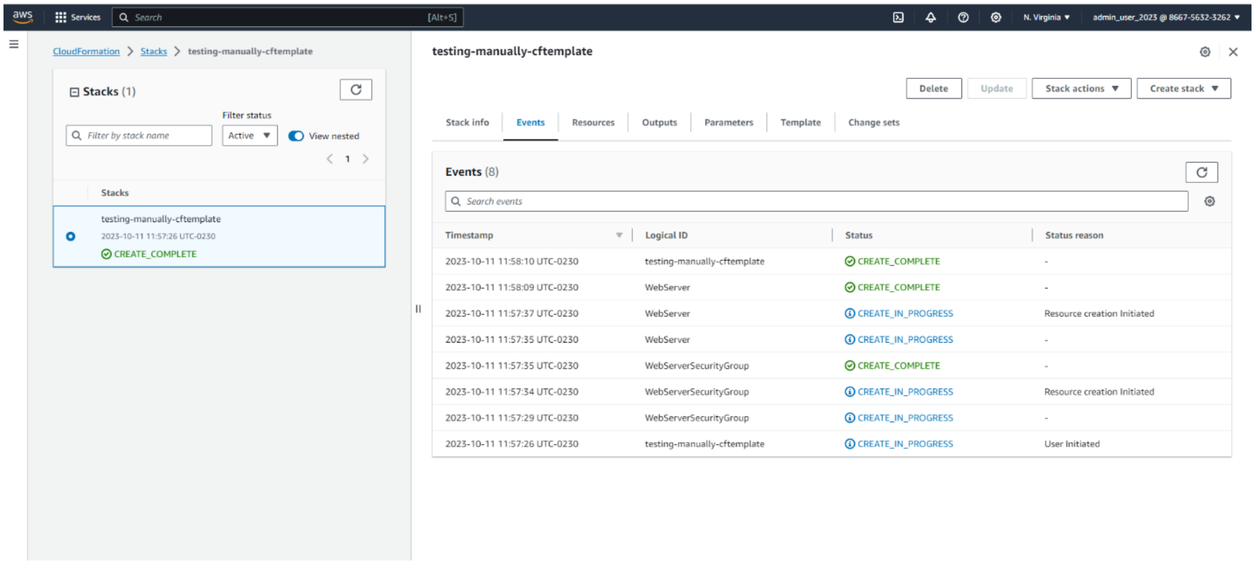
○ Amazon S3 URL ● Upload a template file

Upload a template file

cftemplate.yml

JSON or YAML formatted file

S3 URL: https://s3.us-east-1.amazonaws.com/cf-templates-1fz1qx89s6nlt-us-east-1/2023-10-11T142347.401Zjfi-cftemplate.yml

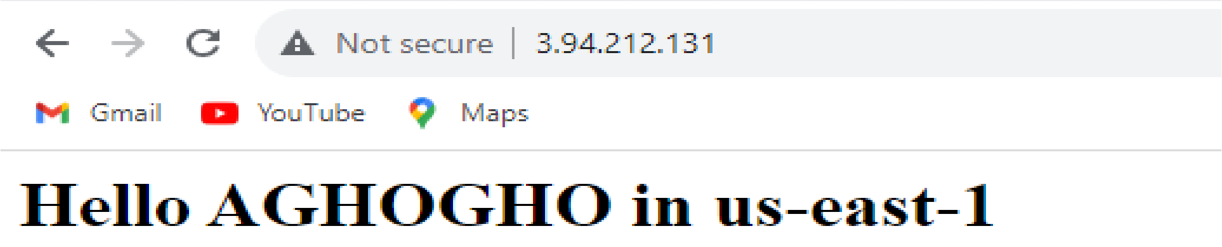


Stacks
testing-manually-cftemplate 2023-10-11 11:57:26 UTC-0230 CREATE_COMPLETE

testing-manually-cftemplate

Events (8)

Timestamp	Logical ID	Status	Status reason
2023-10-11 11:58:10 UTC-0230	testing-manually-cftemplate	CREATE_COMPLETE	-
2023-10-11 11:58:09 UTC-0230	WebServer	CREATE_COMPLETE	-
2023-10-11 11:57:37 UTC-0230	WebServer	CREATE_IN_PROGRESS	Resource creation Initiated
2023-10-11 11:57:35 UTC-0230	WebServer	CREATE_IN_PROGRESS	-
2023-10-11 11:57:35 UTC-0230	WebServerSecurityGroup	CREATE_COMPLETE	-
2023-10-11 11:57:34 UTC-0230	WebServerSecurityGroup	CREATE_IN_PROGRESS	Resource creation Initiated
2023-10-11 11:57:29 UTC-0230	WebServerSecurityGroup	CREATE_IN_PROGRESS	-
2023-10-11 11:57:26 UTC-0230	testing-manually-cftemplate	CREATE_IN_PROGRESS	User Initiated



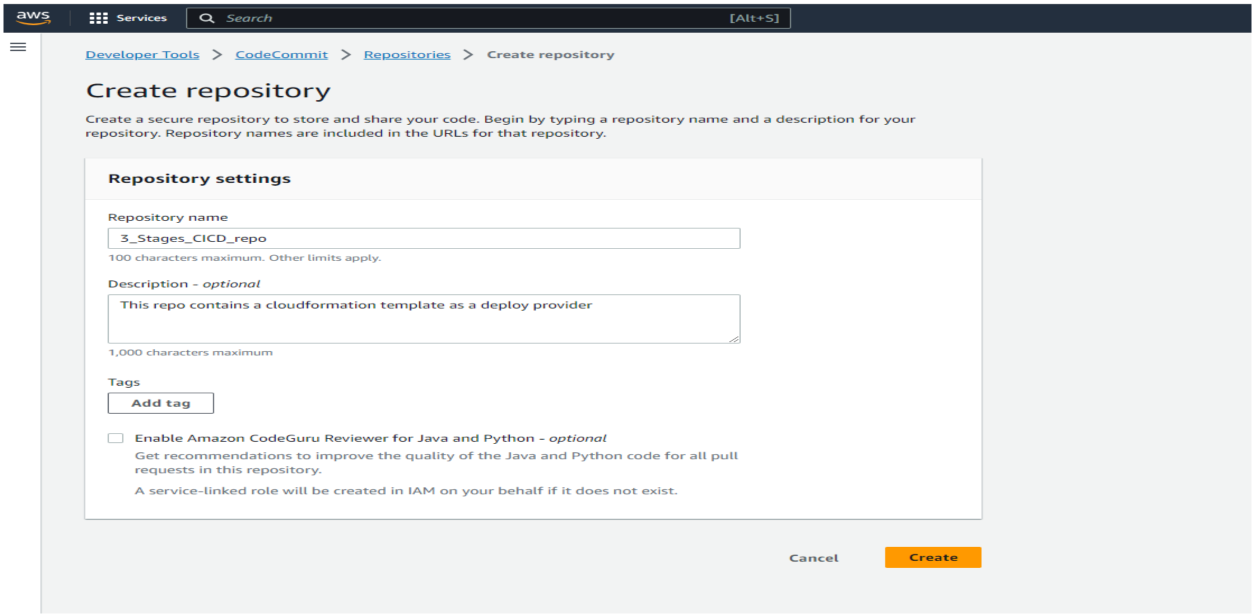
Hello AGHOGHO in us-east-1

From manual testing view, the Cftemplate works.

With my buildspec well written above, I am confident in proceeding with automation. For this Project Road Map Schematics of this project the road map is

Local Testing → Write Buildspec → Source Stage (CodeCommit) → Build Stage (CodeBuild) → Deploy Stage → CI/CD

The Next Step is creating the source stage, and I used CodeCommit.



Developer Tools > CodeCommit > Repositories > Create repository

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name: 3_Stages_CICD_repo

100 characters maximum. Other limits apply.

Description - *optional*: This repo contains a cloudformation template as a deploy provider

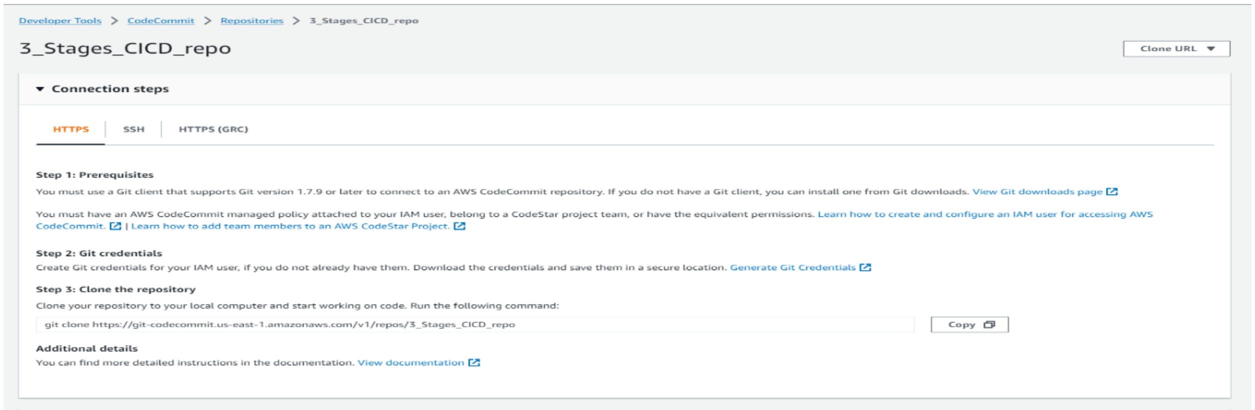
1,000 characters maximum

Tags

□ Enable Amazon CodeGuru Reviewer for Java and Python - *optional*

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.


Developer Tools > CodeCommit > Repositories > 3_Stages_CICD_repo

3_Stages_CICD_repo

Step 1: Prerequisites

You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. View Git downloads page

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. | Learn how to add team members to an AWS CodeStar Project.

Step 2: Git credentials

Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. Generate Git Credentials

Step 3: Clone the repository

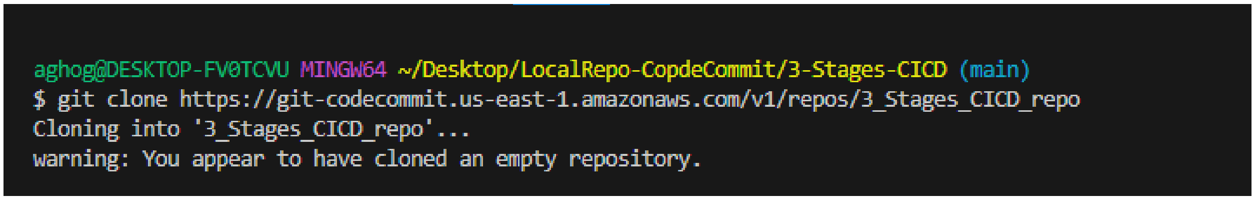
Clone your repository to your local computer and start working on code. Run the following command:

```
git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/3_Stages_CICD_repo
```

Additional details

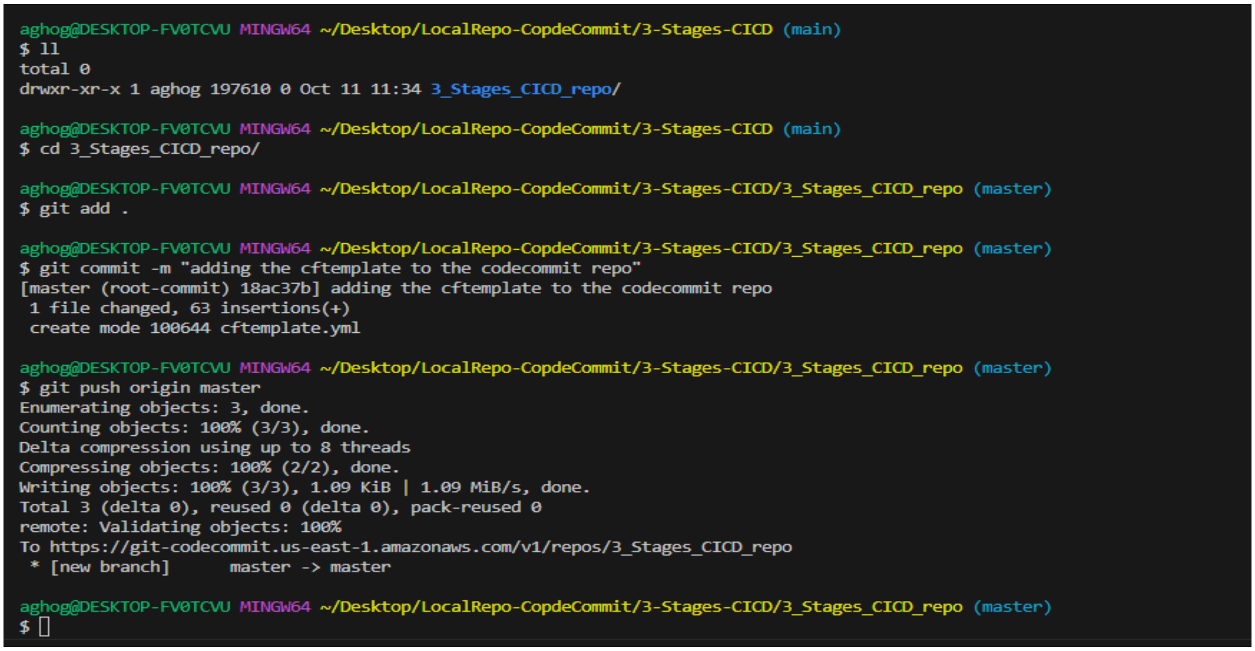
You can find more detailed instructions in the documentation. View documentation

I cloned the code commit repository.



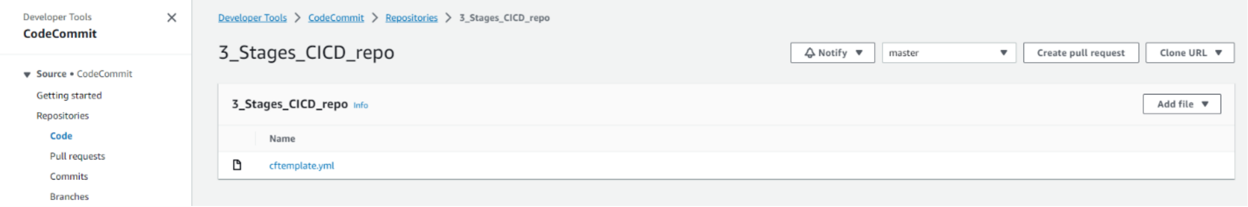
```
aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD (main)
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/3_Stages_CICD_repo
Cloning into '3_Stages_CICD_repo'...
warning: You appear to have cloned an empty repository.
```

Next I pushed the cftemplate to the codecommit repository “3-Stages-CICD_repo”



```
aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD (main)
$ ll
total 0
drwxr-xr-x 1 aghog 197610 0 Oct 11 11:34 3_Stages_CICD_repo/

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD (main)
$ cd 3_Stages_CICD_repo/

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ git add .

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ git commit -m "adding the cftemplate to the codecommit repo"
[master (root-commit) 18ac37b] adding the cftemplate to the codecommit repo
 1 file changed, 63 insertions(+)
 create mode 100644 cftemplate.yml

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.09 KiB | 1.09 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/3_Stages_CICD_repo
 * [new branch]      master -> master

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ 
```

The cftemplate reflects on the 3-Stages-CICD_repo

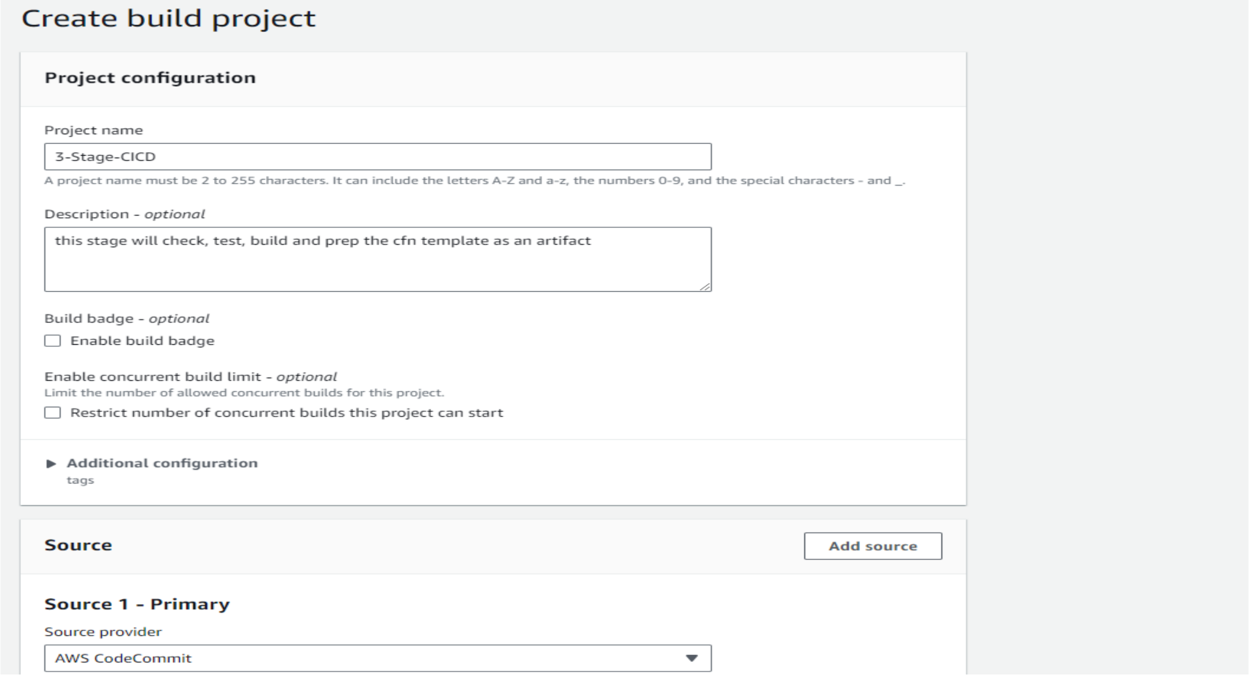


3_Stages_CICD_repo

3_Stages_CICD_repo

Name
cftemplate.yml

Next stage is the Build stage, and I used CodeBuild.



Create build project

Project configuration

Project name

3-Stage-CICD

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Description - *optional*

this stage will check, test, build and prep the cfn template as an artifact

Build badge - *optional*

□ Enable build badge

Enable concurrent build limit - *optional*

Limit the number of allowed concurrent builds for this project.

□ Restrict number of concurrent builds this project can start

▶ Additional configuration

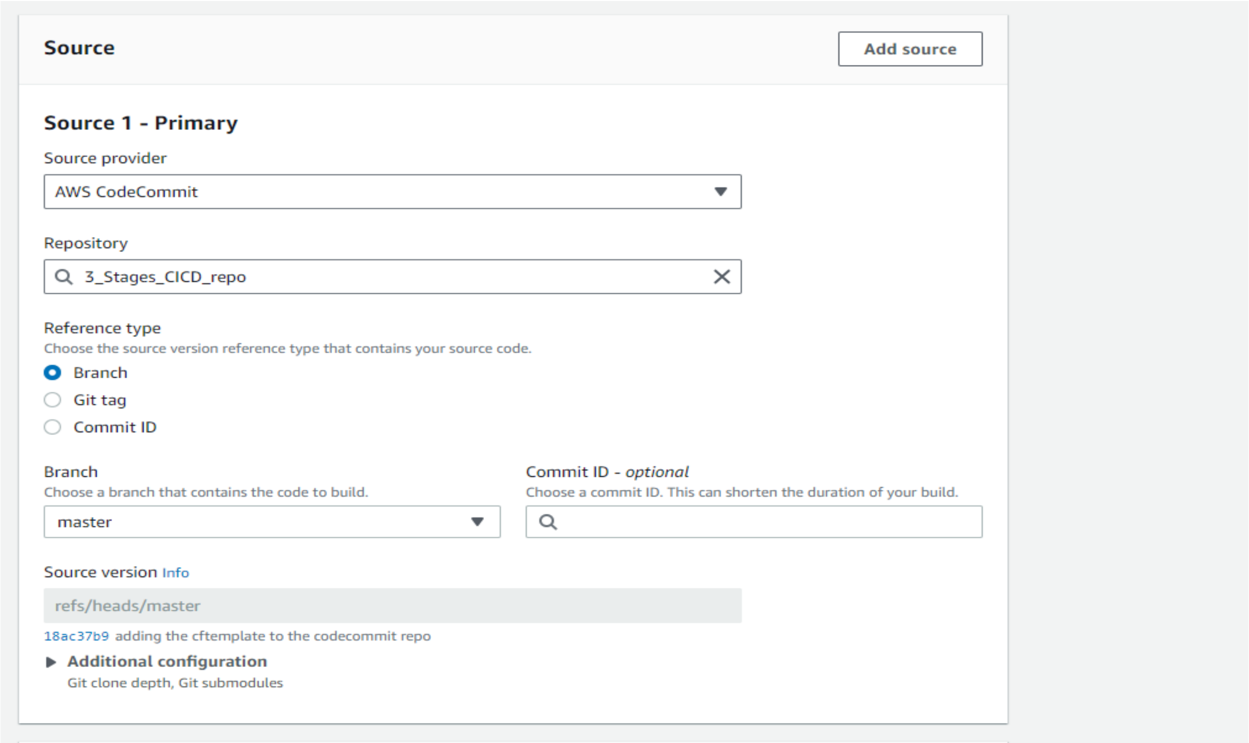
tags

Source

Source 1 - Primary

Source provider

AWS CodeCommit



Source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

3_Stages_CICD_repo

Reference type

Choose the source version reference type that contains your source code.

● Branch

○ Git tag

○ Commit ID

Branch

Choose a branch that contains the code to build.

master

Commit ID - *optional*

Choose a commit ID. This can shorten the duration of your build.

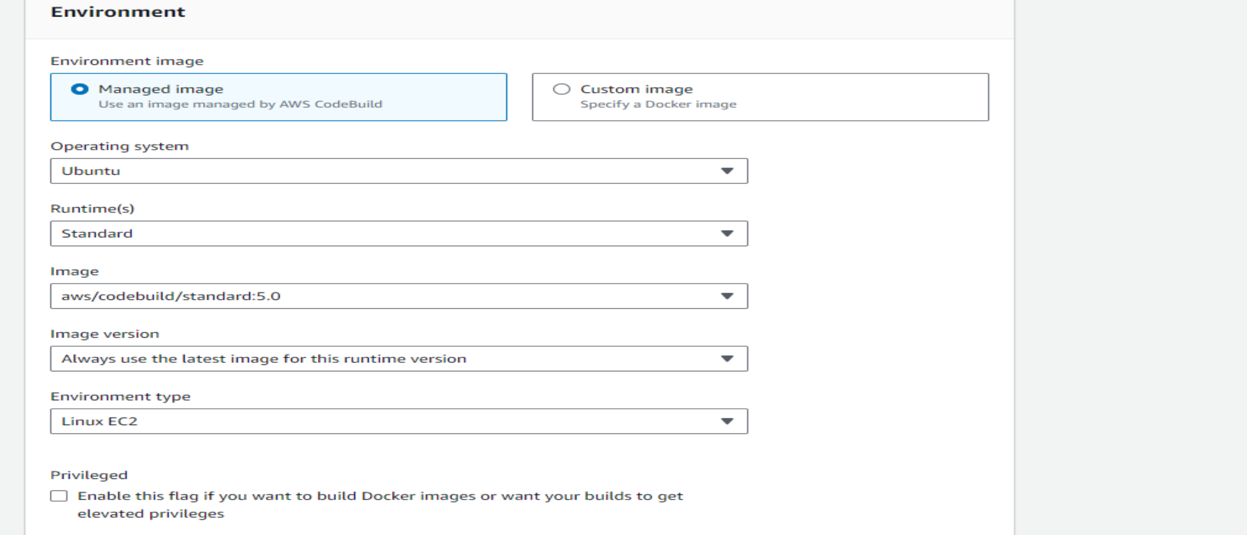
Source version Info

refs/heads/master

18ac37b9 adding the cftemplate to the codecommit repo

▶ Additional configuration

Git clone depth, Git submodules



Environment

Environment image

● Managed image
Use an image managed by AWS CodeBuild

○ Custom image
Specify a Docker image

Operating system
Ubuntu

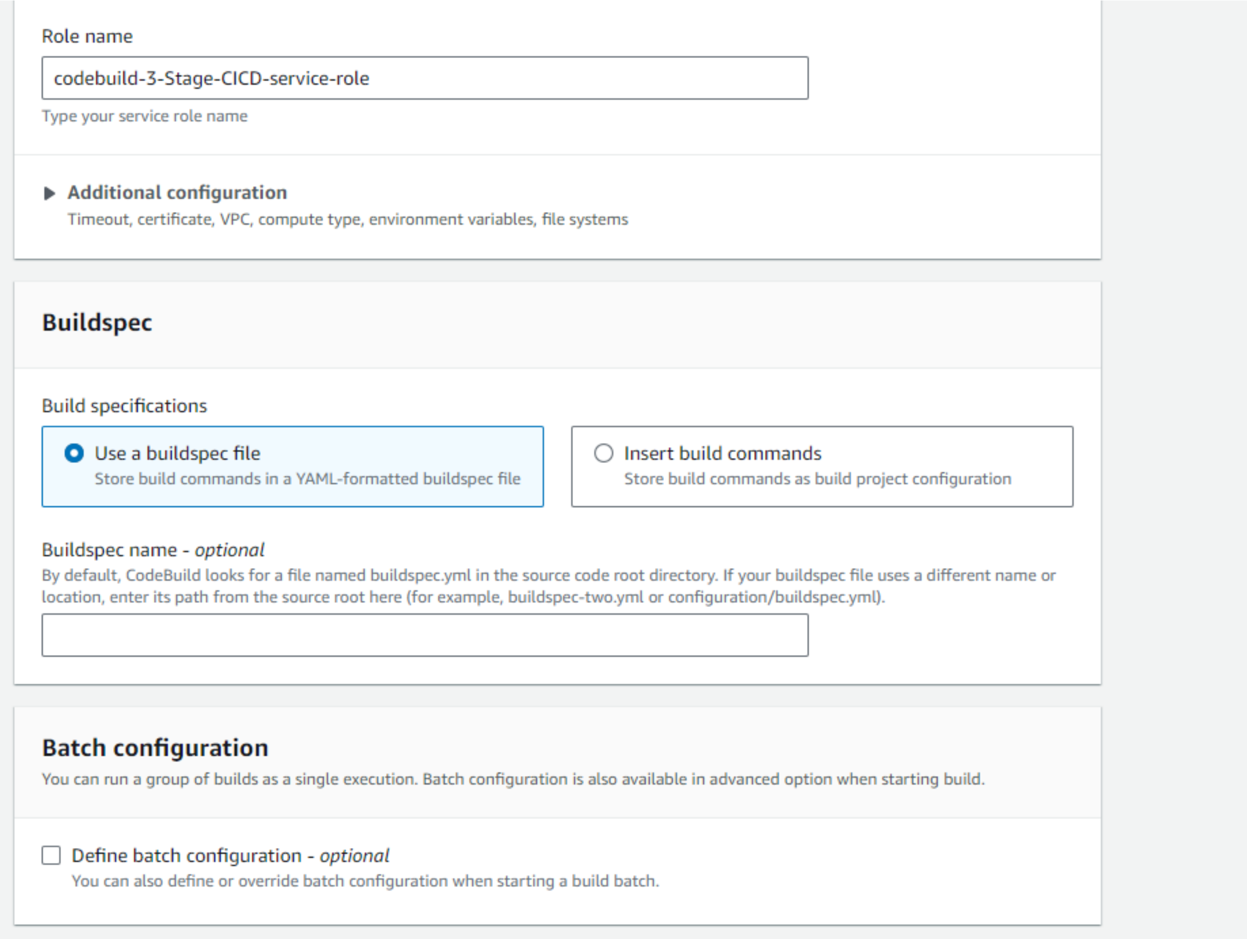
Runtime(s)
Standard

Image
aws/codebuild/standard:5.0

Image version
Always use the latest image for this runtime version

Environment type
Linux EC2

Privileged
□ Enable this flag if you want to build Docker images or want your builds to get elevated privileges



Role name
codebuild-3-Stage-CICD-service-role
Type your service role name

▶ **Additional configuration**
Timeout, certificate, VPC, compute type, environment variables, file systems

Buildspec

Build specifications

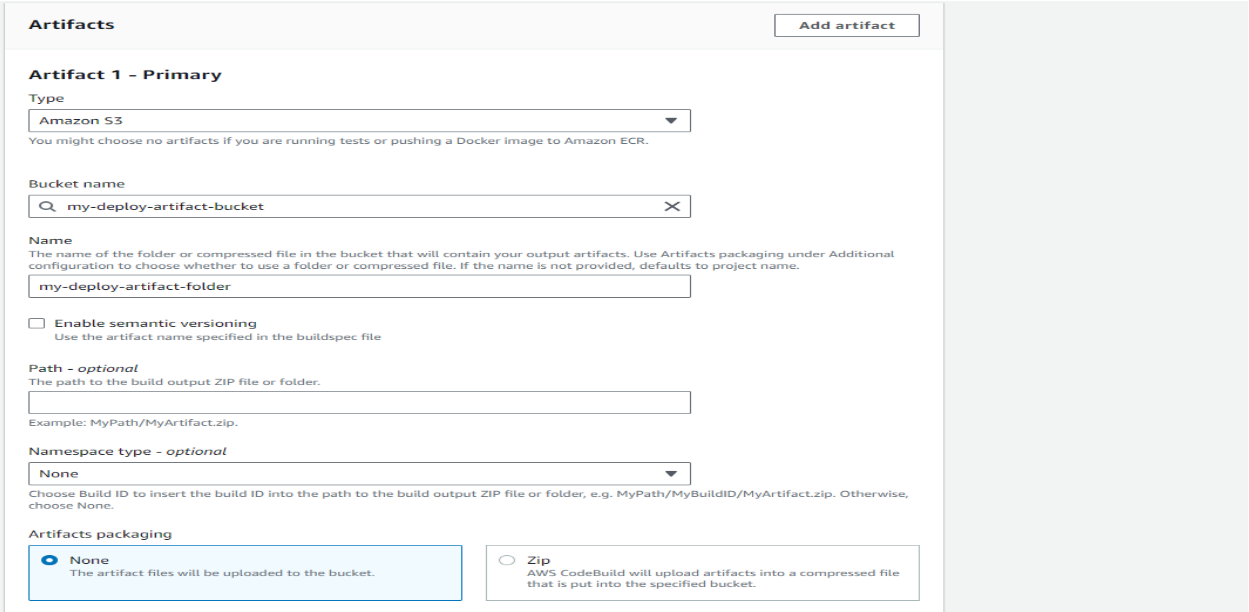
● Use a buildspec file
Store build commands in a YAML-formatted buildspec file

○ Insert build commands
Store build commands as build project configuration

Buildspec name - *optional*
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

Batch configuration
You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.

□ Define batch configuration - *optional*
You can also define or override batch configuration when starting a build batch.



Artifacts

Artifact 1 - Primary

Type

Amazon S3 ▾

You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

Bucket name

my-deploy-artifact-bucket

Name

The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.

my-deploy-artifact-folder

□ Enable semantic versioning
Use the artifact name specified in the buildspec file

Path - *optional*
The path to the build output ZIP file or folder.

Example: MyPath/MyArtifact.zip.

Namespace type - *optional*

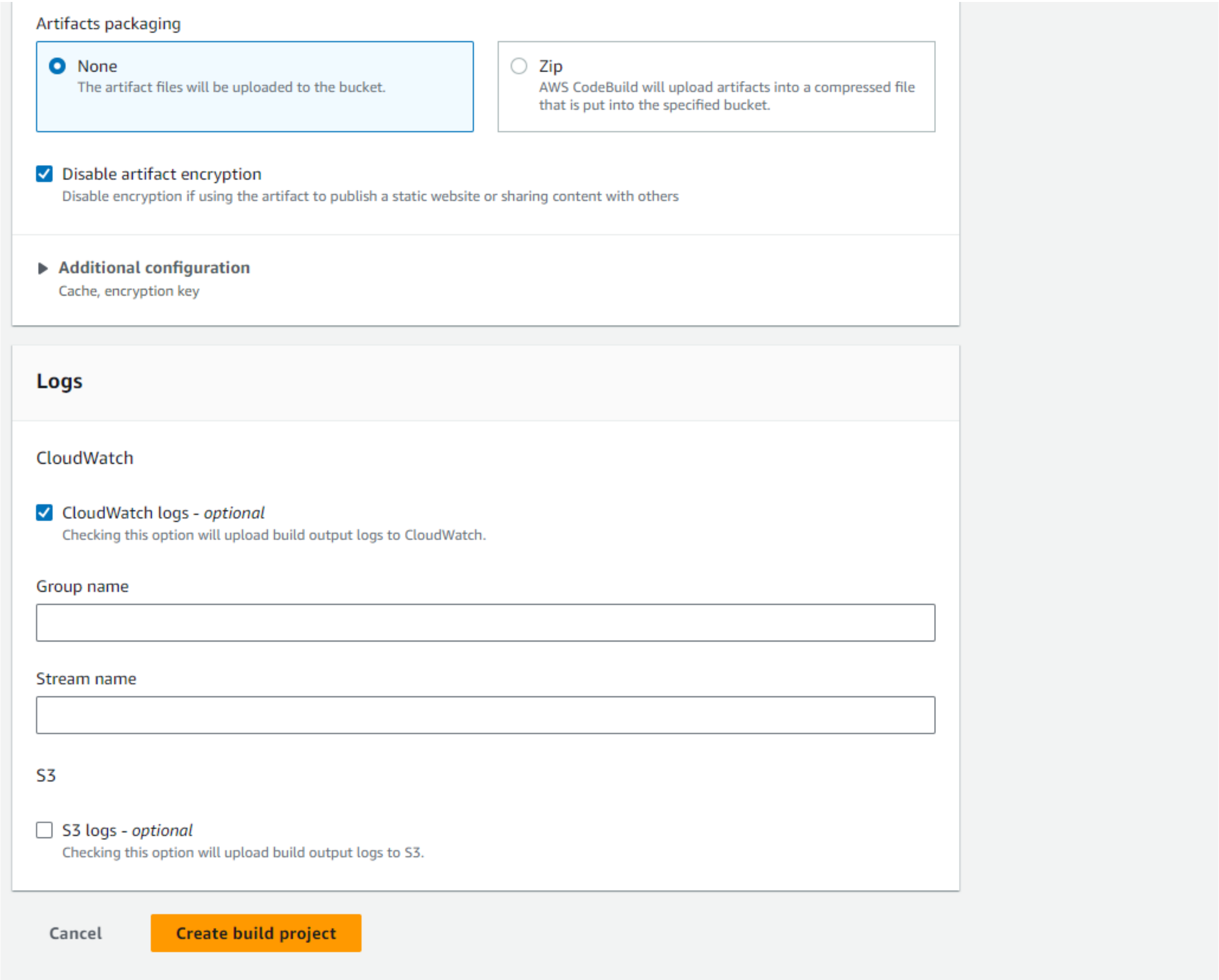
None ▾

Choose Build ID to insert the build ID into the path to the build output ZIP file or folder, e.g. MyPath/MyBuildID/MyArtifact.zip. Otherwise, choose None.

Artifacts packaging

● None
The artifact files will be uploaded to the bucket.

○ Zip
AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.



Artifacts packaging

● None
The artifact files will be uploaded to the bucket.

○ Zip
AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.

☑ Disable artifact encryption
Disable encryption if using the artifact to publish a static website or sharing content with others

▶ Additional configuration
Cache, encryption key

Logs

CloudWatch

☑ CloudWatch logs - *optional*
Checking this option will upload build output logs to CloudWatch.

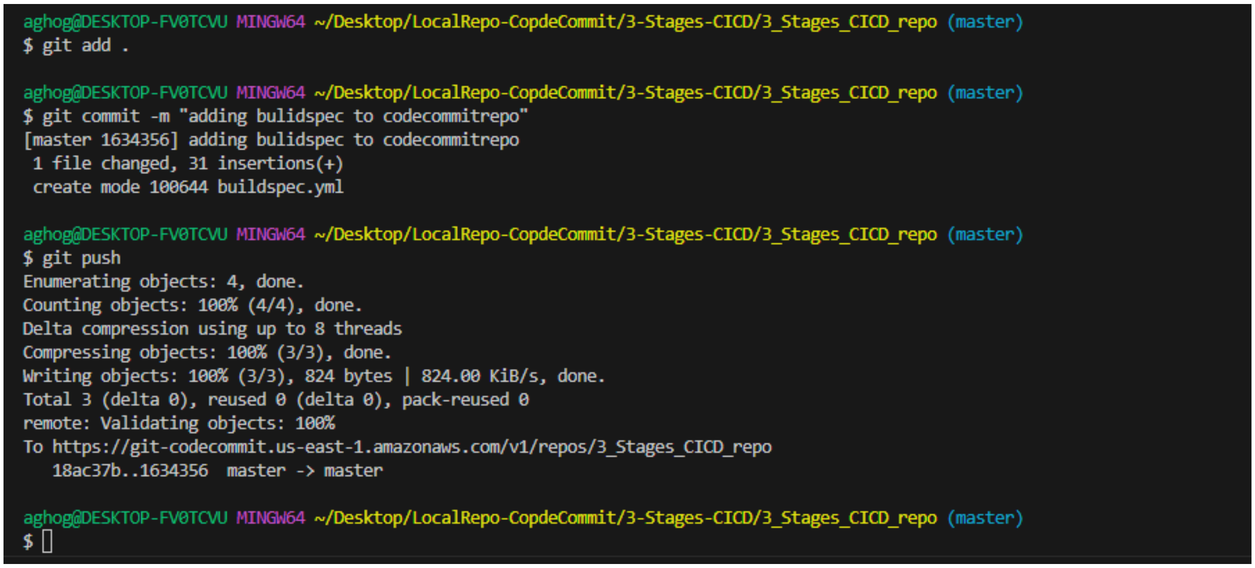
Group name

Stream name

S3

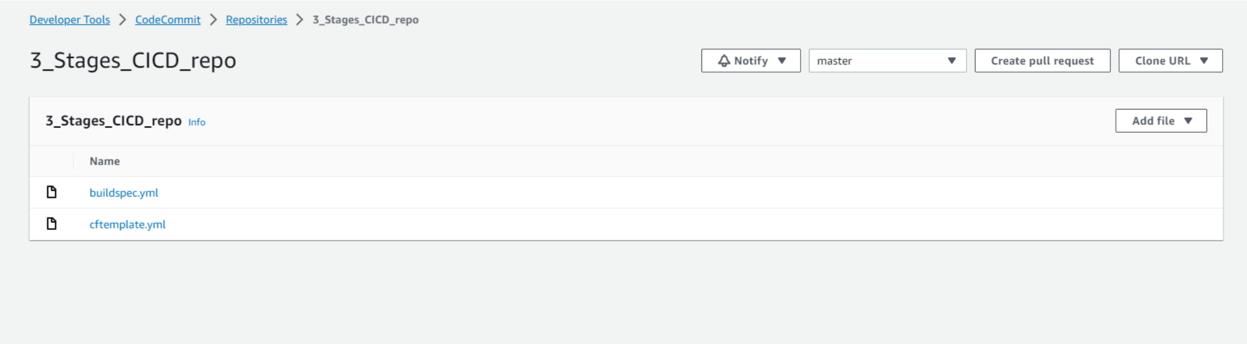
□ S3 logs - *optional*
Checking this option will upload build output logs to S3.

I pushed my buildspec.yml to my codecommit repository “3_Stages_CICD_repo



```
aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ git add .

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ git commit -m "adding bulidspec to codecommitrepo"
[master 1634356] adding bulidspec to codecommitrepo
 1 file changed, 31 insertions(+)
 create mode 100644 buildspec.yml

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 824 bytes | 824.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/3_Stages_CICD_repo
   18ac37b..1634356  master -> master

aghog@DESKTOP-FV0TCVU MINGW64 ~/Desktop/LocalRepo-CopdeCommit/3-Stages-CICD/3_Stages_CICD_repo (master)
$ 
```

Confirming the push update on the console



Developer Tools > CodeCommit > Repositories > 3_Stages_CICD_repo
Notify ▼
master ▼
Create pull request
Clone URL ▼
Info
Add file ▼

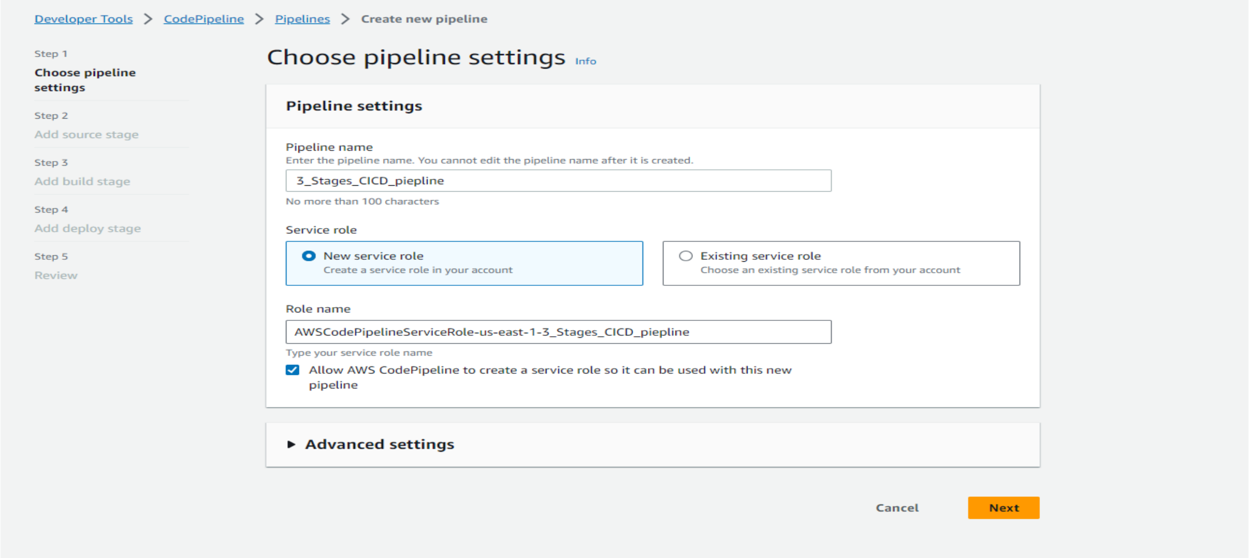
3_Stages_CICD_repo

3_Stages_CICD_repo

Name
buildspec.yml
cftemplate.yml

Next stage is building a CI/CD Pipeline.

Pipeline Settings:



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Choose pipeline settings Info

Pipeline settings

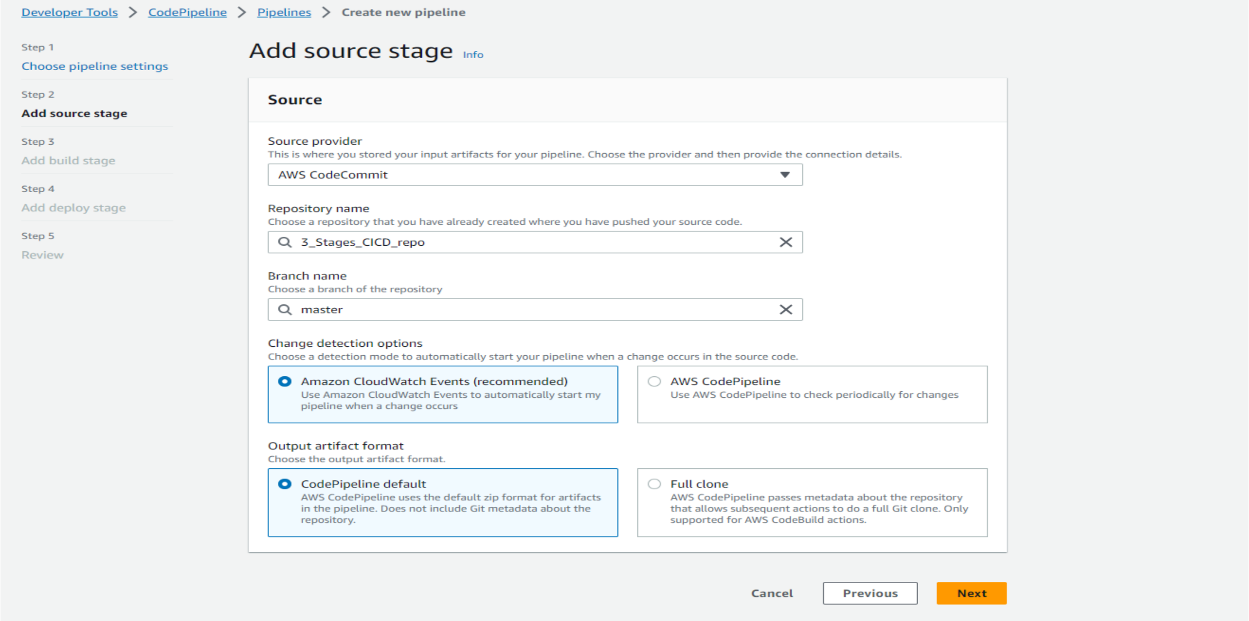
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
3_Stages_CICD_piepline
No more than 100 characters

Service role
● New service role
Create a service role in your account
○ Existing service role
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-us-east-1-3_Stages_CICD_piepline
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

▸ **Advanced settings**

Source Stage:



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add source stage Info

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
AWS CodeCommit

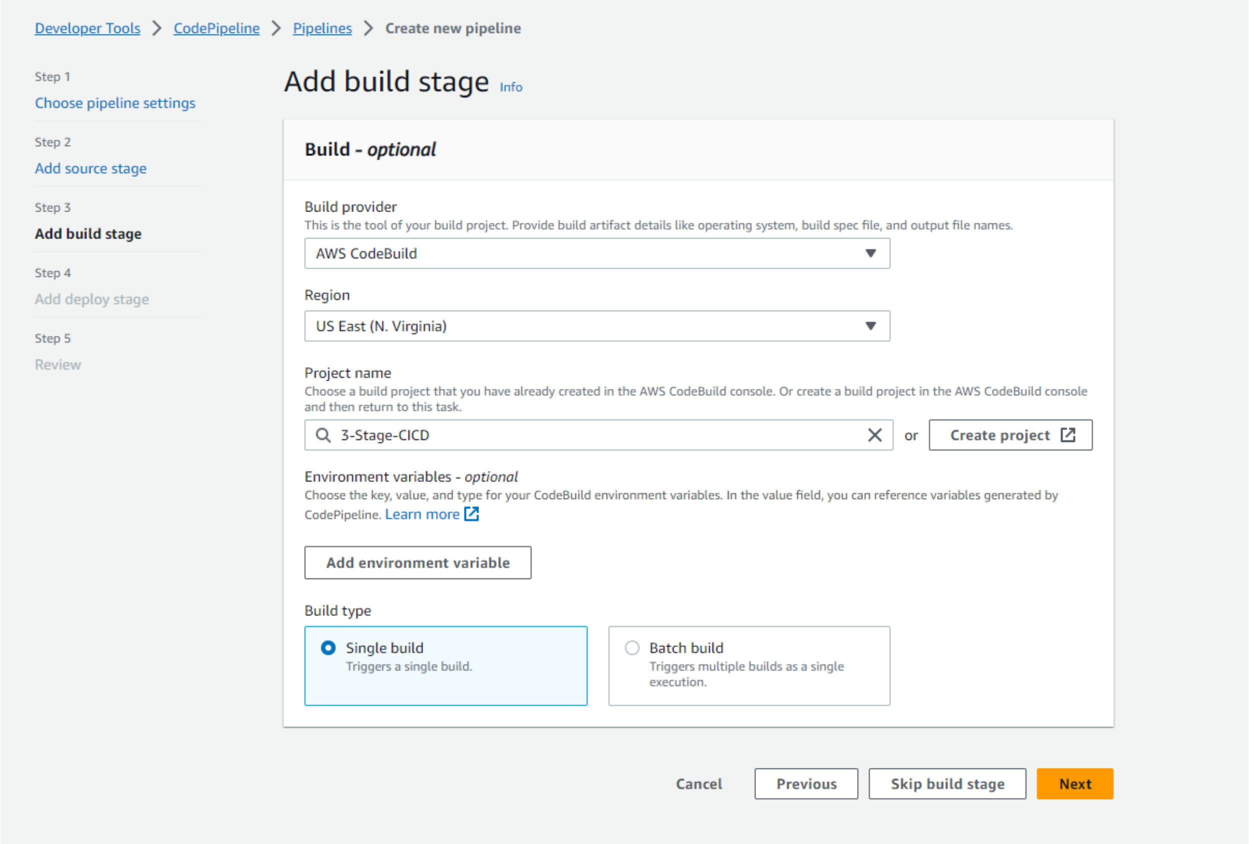
Repository name
Choose a repository that you have already created where you have pushed your source code.
3_Stages_CICD_repo

Branch name
Choose a branch of the repository
master

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
● Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Output artifact format
Choose the output artifact format.
● CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
○ Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Build Stage:



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add build stage Info

Build - *optional*

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild ▼

Region
US East (N. Virginia) ▼

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
3-Stage-CICD ✕ or Create project

Environment variables - *optional*
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

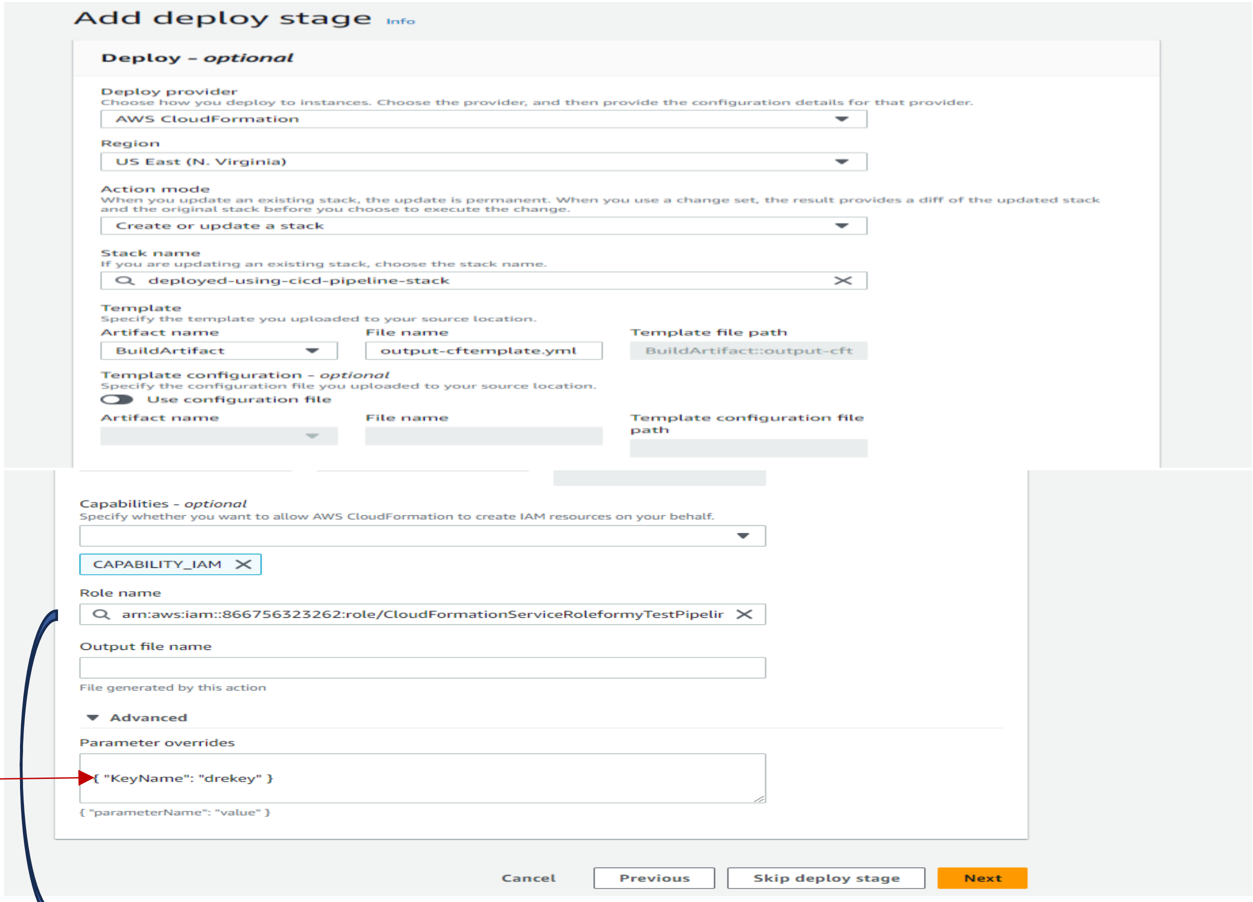
Add environment variable

Build type

● Single build
Triggers a single build.

○ Batch build
Triggers multiple builds as a single execution.

Deploy Stage: First Action is the (DeployToTest) Environment



Add deploy stage Info

Deploy – *optional*

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CloudFormation

Region
US East (N. Virginia)

Action mode
When you update an existing stack, the update is permanent. When you use a change set, the result provides a diff of the updated stack and the original stack before you choose to execute the change.
Create or update a stack

Stack name
If you are updating an existing stack, choose the stack name.
deployed-using-cicd-pipeline-stack

Template
Specify the template you uploaded to your source location.

Artifact name	File name	Template file path
BuildArtifact	output-cftemplate.yml	BuildArtifact::output-cft

Template configuration - *optional*
Specify the configuration file you uploaded to your source location.
Use configuration file

Artifact name	File name	Template configuration file path

Capabilities - *optional*
Specify whether you want to allow AWS CloudFormation to create IAM resources on your behalf.
CAPABILITY_IAM

Role name
arn:aws:iam::866756323262:role/CloudFormationServiceRoleformyTestPipelir

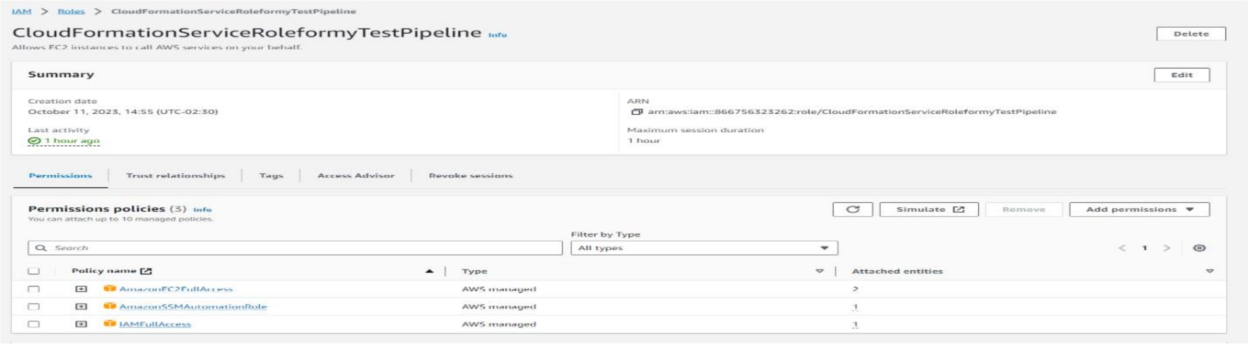
Output file name

File generated by this action

Advanced

Parameter overrides
{ "KeyName": "drekey" }
{ "parameterName": "value" }

An IAM Role was created as shown below.



IAM > Roles > CloudFormationServiceRoleformyTestPipeline

CloudFormationServiceRoleformyTestPipeline Info
Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
October 11, 2023, 14:55 (UTC-02:30)

Last activity
1 hour ago

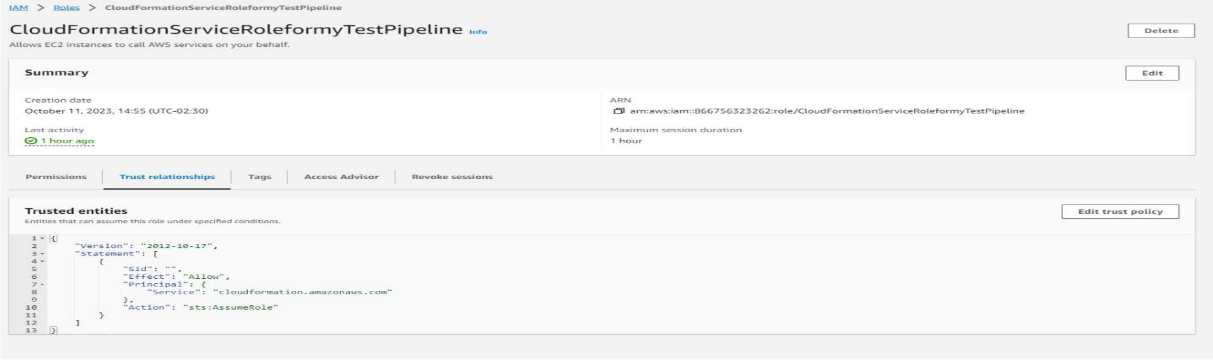
ARN
arn:aws:iam::866756323262:role/CloudFormationServiceRoleformyTestPipeline

Maximum session duration
1 hour

Permissions policies (3) Info
You can attach up to 10 managed policies.

Filter by Type
All types

Policy name	Type	Attached entities
AmazonEC2FullAccess	AWS managed	2
AmazonSSMAutomationRole	AWS managed	1
IAMFullAccess	AWS managed	1



CloudFormationServiceRoleformyTestPipeline

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
October 11, 2023, 14:55 (UTC-02:30)

Last activity
1 hour ago

ARN
arn:aws:iam::866756323262:role/CloudFormationServiceRoleformyTestPipeline

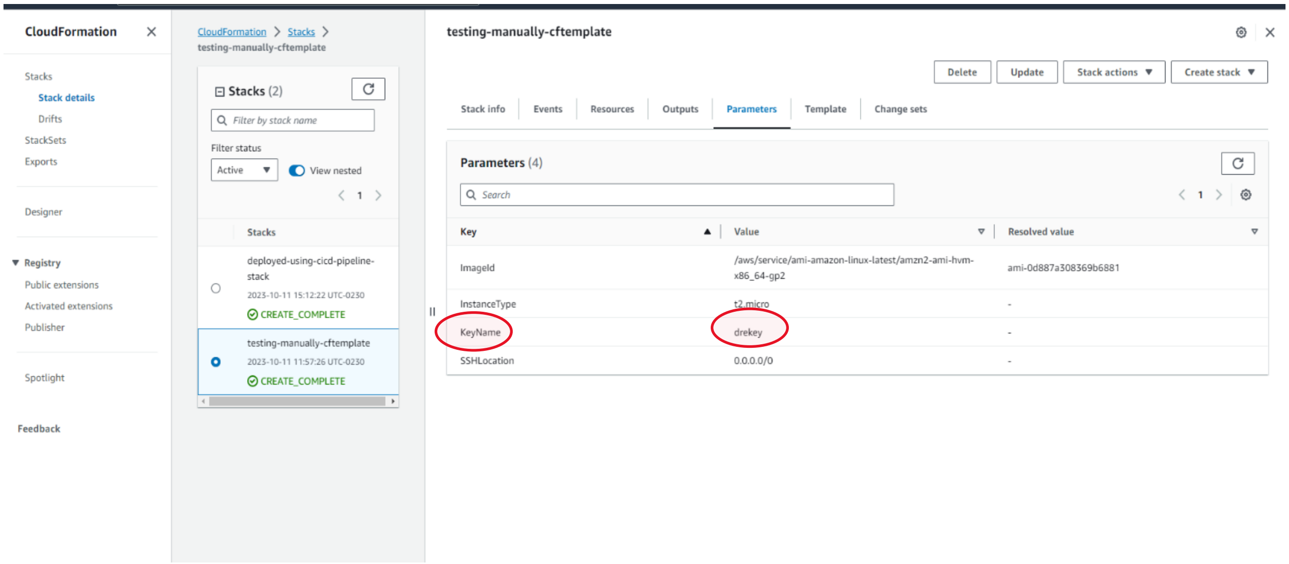
Maximum session duration
1 hour

Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

For the Parameter Overrides from the settings above, I copied the key and value on the cftemplate.



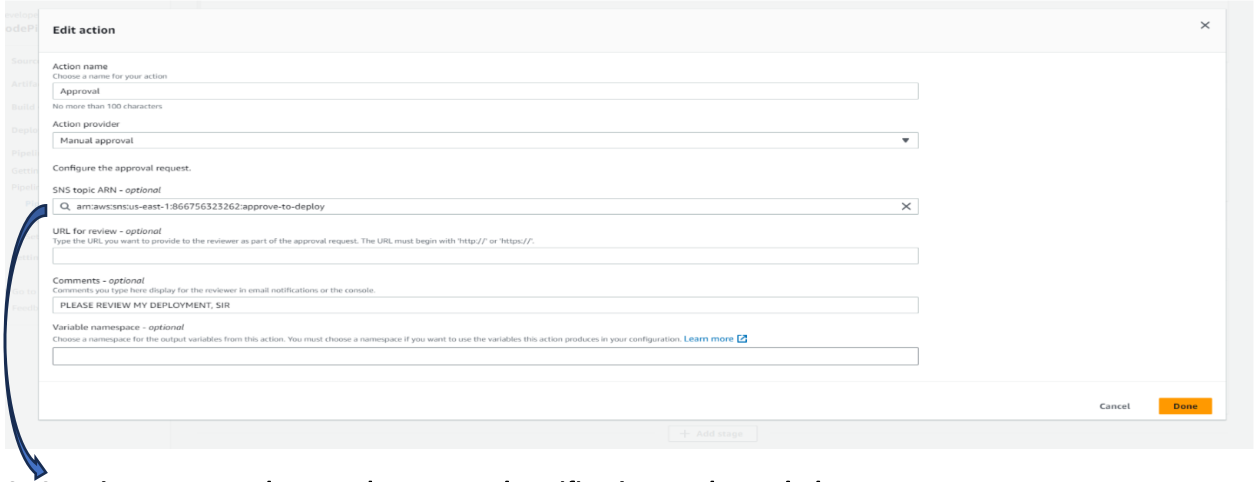
Stacks
deployed-using-cicd-pipeline-stack 2023-10-11 15:12:22 UTC-0230 CREATE_COMPLETE
testing-manually-cftemplate 2023-10-11 11:57:26 UTC-0230 CREATE_COMPLETE

testing-manually-cftemplate

Parameters (4)

Key	Value	Resolved value
ImageId	/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2	ami-0d887a308369b6881
InstanceType	t2.micro	-
KeyName	drekey	-
SSHLocation	0.0.0.0/0	-

Furthermore, on the Deploy stage, the deploy stage was edited to add an “action” for manual approval as shown below.



Edit action

Action name
Choose a name for your action
Approval
No more than 100 characters

Action provider
Manual approval

Configure the approval request.

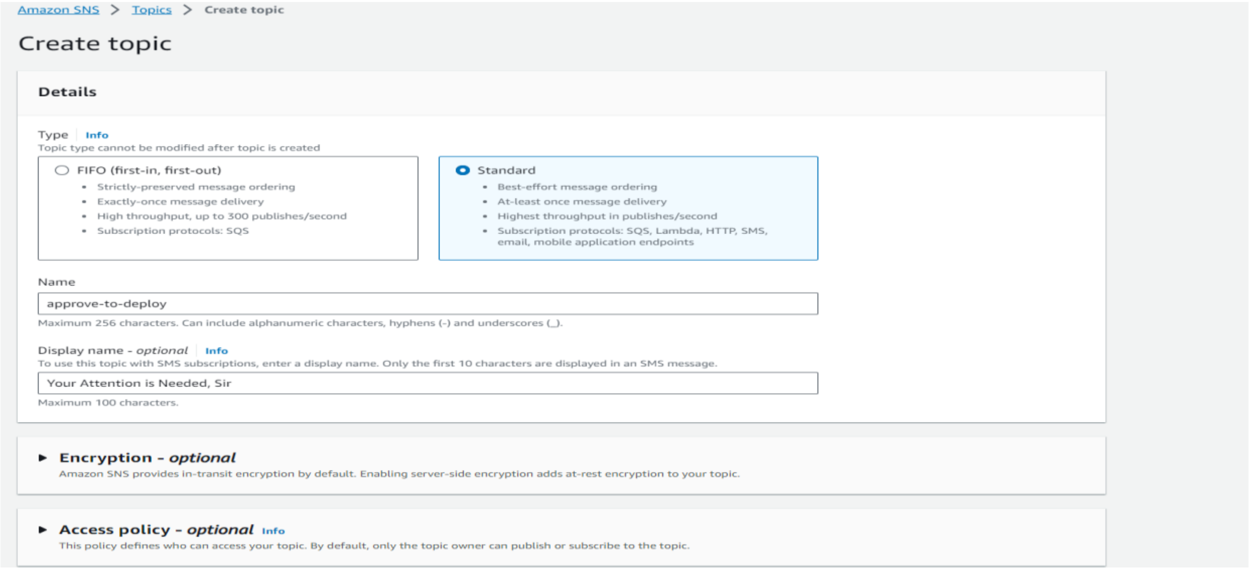
SNS topic ARN - *optional*
arn:aws:sns:us-east-1:866756323262:approve-to-deploy

URL for review - *optional*
Type the URL you want to provide to the reviewer as part of the approval request. The URL must begin with 'http://' or 'https://'.

Comments - *optional*
Comments you type here display for the reviewer in email notifications or the console.
PLEASE REVIEW MY DEPLOYMENT, SIR

Variable namespace - *optional*
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more

SNS topic was created to get the approval notification as shown below.



Amazon SNS > Topics > Create topic

Create topic

Details

Type Info
Topic type cannot be modified after topic is created

○ FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

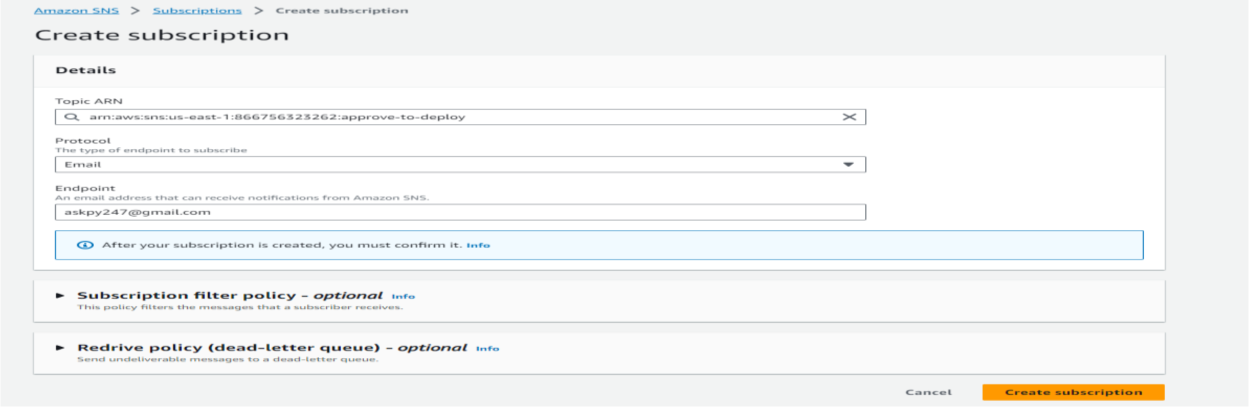
● Standard
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name
approve-to-deploy
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* Info
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.
Your Attention is Needed, Sir
Maximum 100 characters.

▸ Encryption - *optional*
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

▸ Access policy - *optional* Info
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.



Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN
arn:aws:sns:us-east-1:866756323262:approve-to-deploy

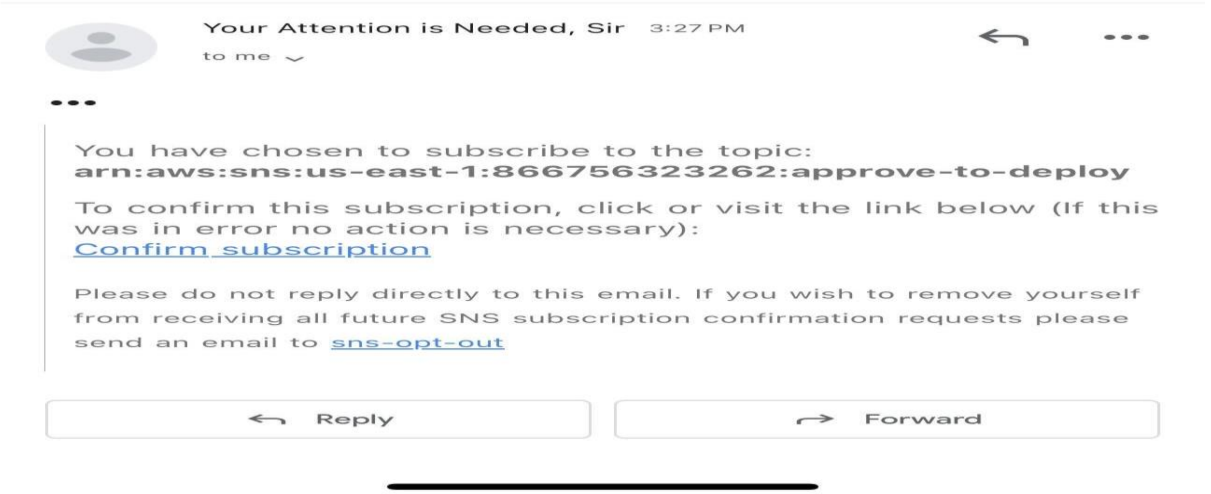
Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
askpy247@gmail.com

After your subscription is created, you must confirm it. Info

▸ Subscription filter policy - *optional* Info
This policy filters the messages that a subscriber receives.

▸ Redrive policy (dead-letter queue) - *optional* Info
Send undeliverable messages to a dead-letter queue.



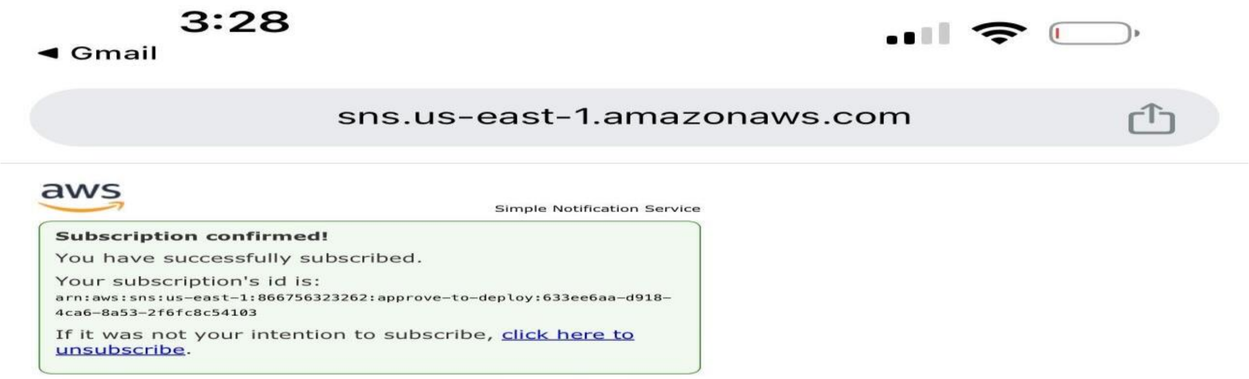
Your Attention is Needed, Sir 3:27 PM

to me

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:866756323262:approve-to-deploy

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out

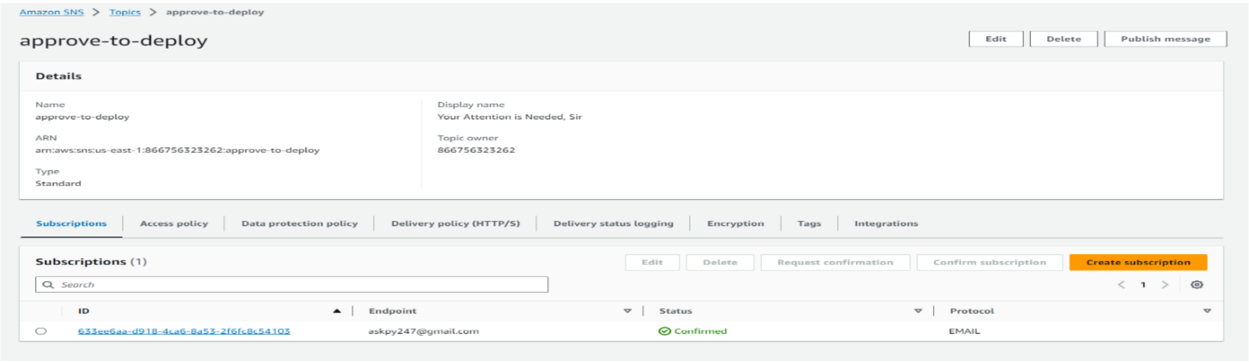


Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-east-1:866756323262:approve-to-deploy:633ee6aa-d918-4ca6-8a53-2f6fc8c54103

If it was not your intention to subscribe, click here to unsubscribe.



approve-to-deploy

Details

Name
approve-to-deploy

ARN
arn:aws:sns:us-east-1:866756323262:approve-to-deploy

Type
Standard

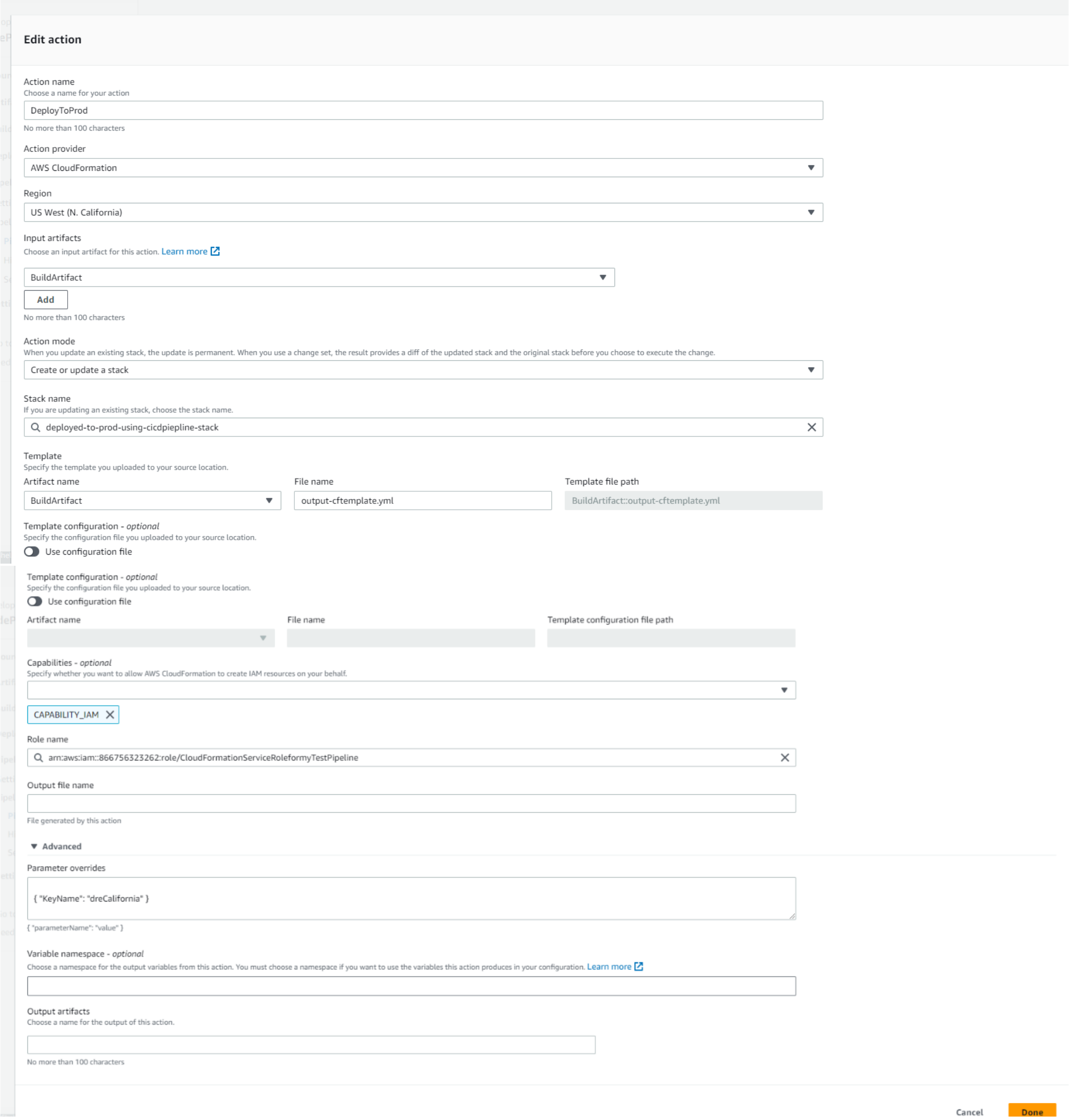
Display name
Your Attention is Needed, Sir

Topic owner
866756323262

Subscriptions (1)

	ID	Endpoint	Status	Protocol
○	633ee6aa-d918-4ca6-8a53-2f6fc8c54103	askpy247@gmail.com	✓ Confirmed	EMAIL

Another "action" (DeployToProd) was added to the Deploy stage for the Production Environment.



Edit action

Action name
Choose a name for your action
DeployToProd
No more than 100 characters

Action provider
AWS CloudFormation

Region
US West (N. California)

Input artifacts
Choose an input artifact for this action. Learn more
BuildArtifact
Add
No more than 100 characters

Action mode
When you update an existing stack, the update is permanent. When you use a change set, the result provides a diff of the updated stack and the original stack before you choose to execute the change.
Create or update a stack

Stack name
If you are updating an existing stack, choose the stack name.
deployed-to-prod-using-cicdpiepline-stack

Template
Specify the template you uploaded to your source location.

Artifact name	File name	Template file path
BuildArtifact	output-cftemplate.yml	BuildArtifact::output-cftemplate.yml

Template configuration - *optional*
Specify the configuration file you uploaded to your source location.
Use configuration file

Template configuration - *optional*
Specify the configuration file you uploaded to your source location.
Use configuration file

Artifact name	File name	Template configuration file path

Capabilities - *optional*
Specify whether you want to allow AWS CloudFormation to create IAM resources on your behalf.
CAPABILITY_IAM

Role name
arn:aws:iam::866756323262:role/CloudFormationServiceRoleformyTestPipeline

Output file name

File generated by this action

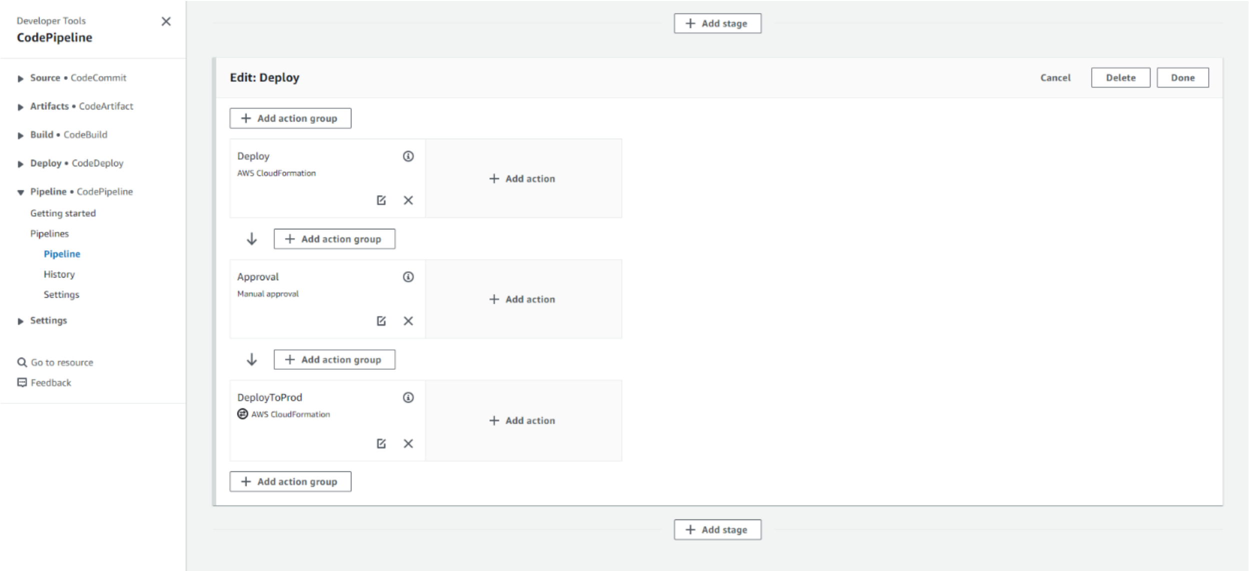
Advanced

Parameter overrides
{ "KeyName": "dreCalifornia" }
{ "parameterName": "value" }

Variable namespace - *optional*
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more

Output artifacts
Choose a name for the output of this action.

No more than 100 characters



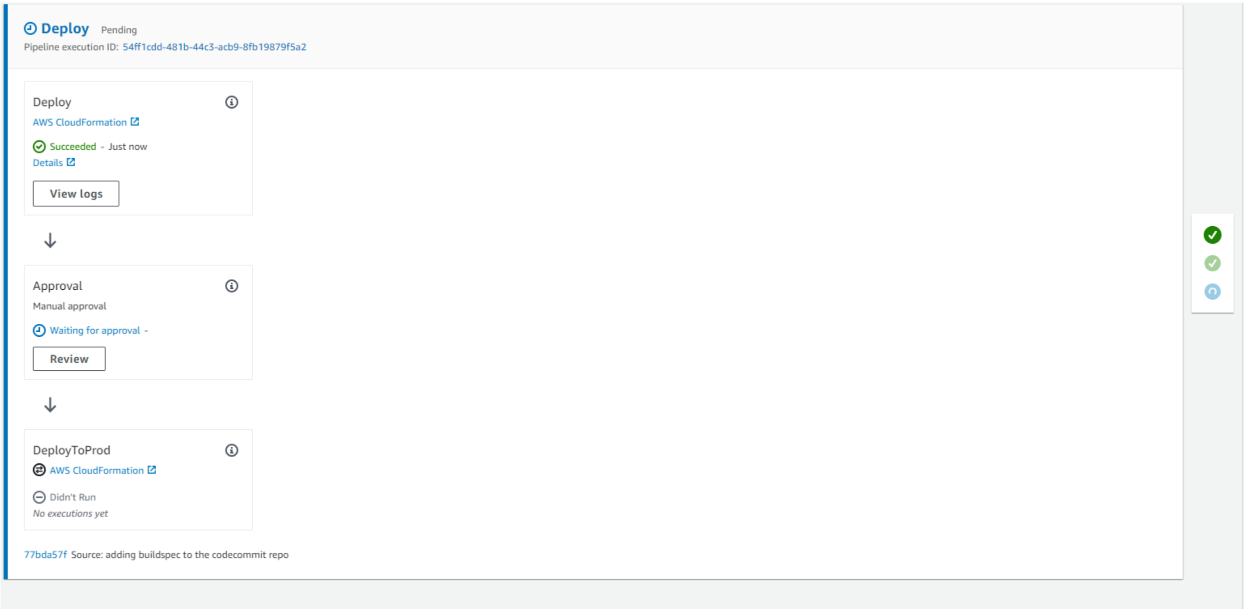
Edit: Deploy

Deploy
AWS CloudFormation

Approval
Manual approval

DeployToProd
AWS CloudFormation

I saved the actions above and released change.



Deploy Pending
Pipeline execution ID: 54ff1cdd-481b-44c3-acb9-8fb19879f5a2

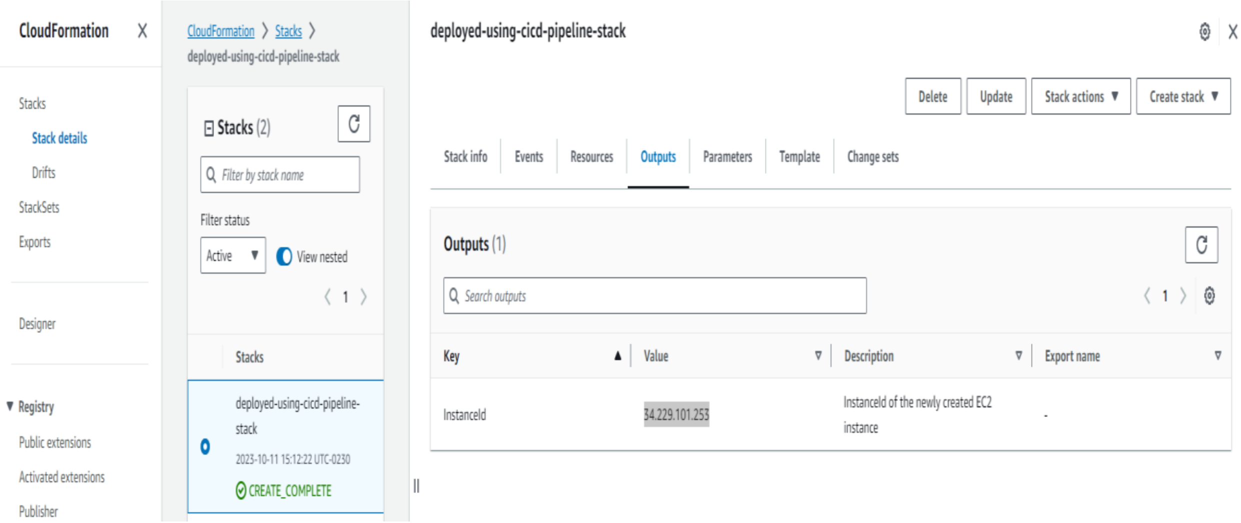
Deploy
AWS CloudFormation
Succeeded - Just now
Details

Approval
Manual approval
Waiting for approval -

DeployToProd
AWS CloudFormation
Didn't Run
No executions yet

77bda57f Source: adding buildspec to the codecommit repo

It was successfully deployed in the Test Environment. I copied the IP address from the stack created in Cloudformation and tested in a browser as shown below.



deployed-using-cicd-pipeline-stack

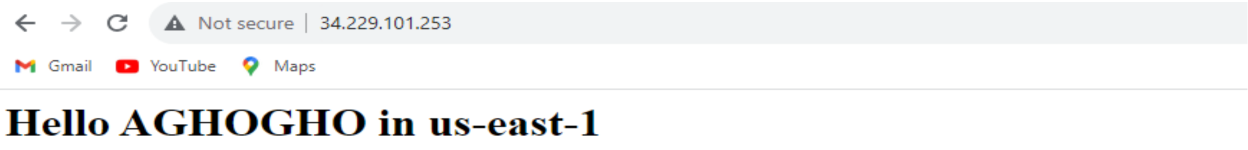
Stacks

deployed-using-cicd-pipeline-stack
2023-10-11 15:12:22 UTC-0230
CREATE_COMPLETE

Outputs (1)

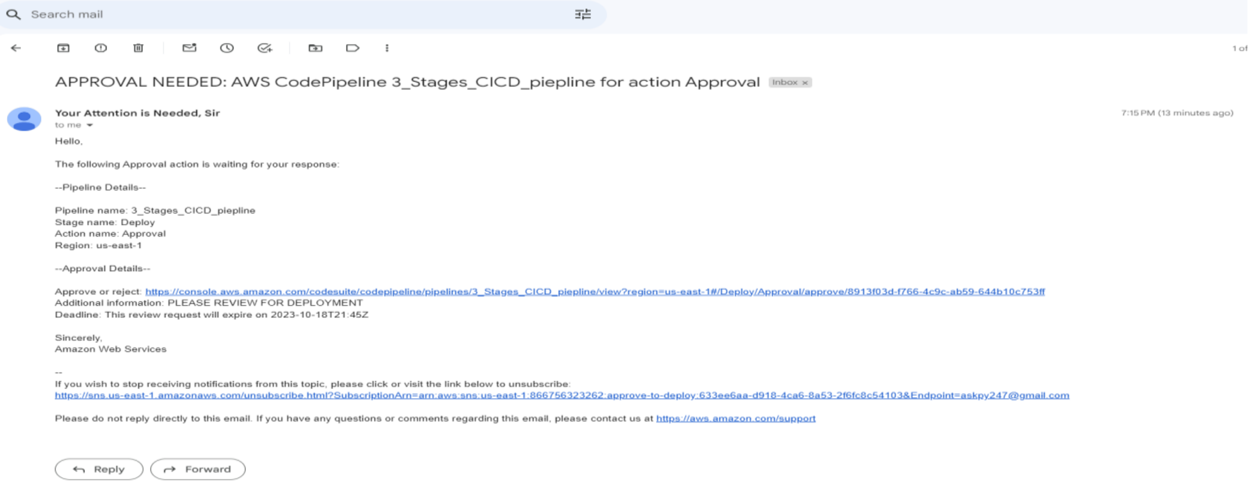
Key	Value	Description	Export name
InstanceId	34.229.101.253	InstanceId of the newly created EC2 instance	-

It works good in the Test Environment "us-east-1" region.



Hello AGHOGHO in us-east-1

I got a notification from SNS to manual review and approve the deployment to the Production Environment as shown below.



APPROVAL NEEDED: AWS CodePipeline 3_Stages_CICD_piepline for action Approval

Your Attention is Needed, Sir
to me
7:15 PM (13 minutes ago)

Hello,

The following Approval action is waiting for your response:

--Pipeline Details--

Pipeline name: 3_Stages_CICD_piepline
Stage name: Deploy
Action name: Approval
Region: us-east-1

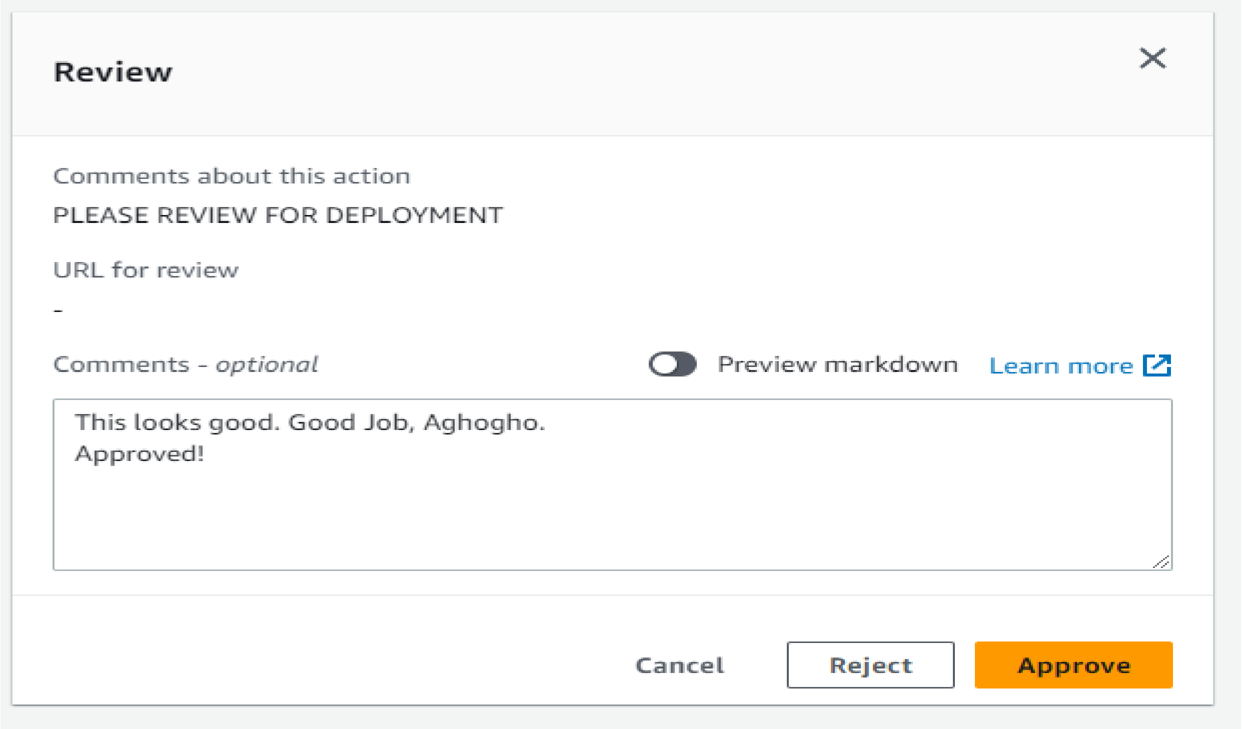
--Approval Details--

Approve or reject: https://console.aws.amazon.com/codesuite/codepipeline/pipelines/3_Stages_CICD_piepline/view?region=us-east-1#/Deploy/Approval/approve/8913f03d-f766-4c9c-ab59-644b10c753ff
Additional information: PLEASE REVIEW FOR DEPLOYMENT
Deadline: This review request will expire on 2023-10-18T21:45Z

Sincerely,
Amazon Web Services

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:866756323262:approve-to-deploy:633ee6aa-d918-4ca6-8a53-2f6fc8c54103&Endpoint=askpy247@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support



Review

Comments about this action

PLEASE REVIEW FOR DEPLOYMENT

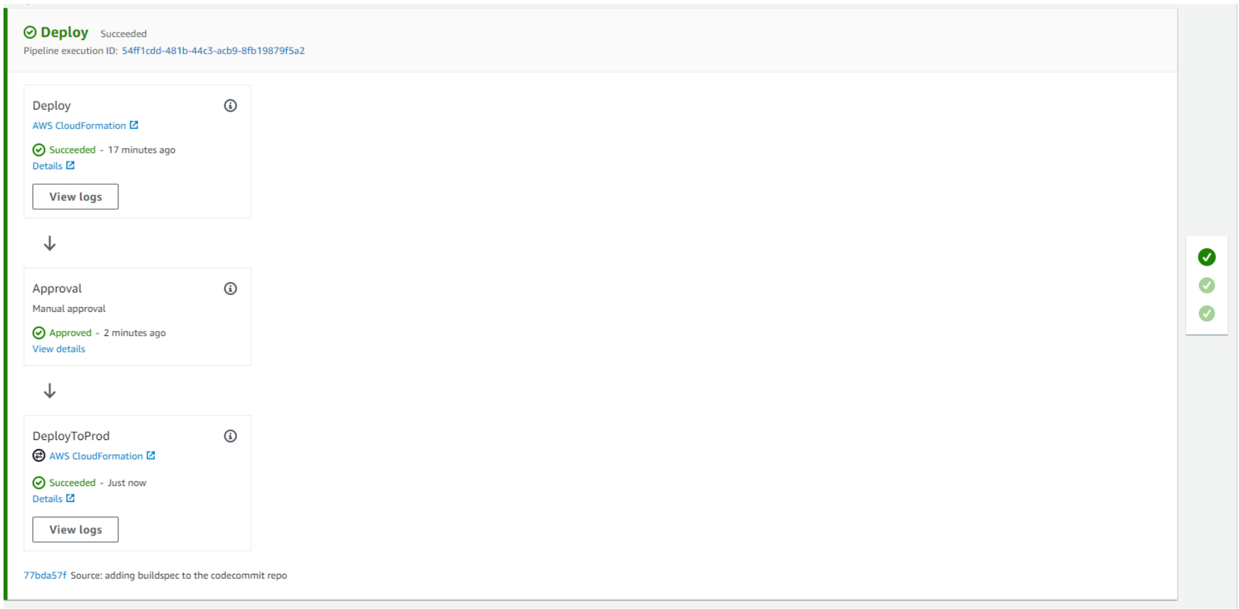
URL for review

-

Comments - *optional*

This looks good. Good Job, Aghogho.
Approved!

Immediately, it was approved, it started deployment to the Production Environment.



Deploy Succeeded

Pipeline execution ID: 54ff1cdd-481b-44c3-acb9-8fb19879f5a2

Deploy

AWS CloudFormation

Succeeded - 17 minutes ago

Approval

Manual approval

Approved - 2 minutes ago

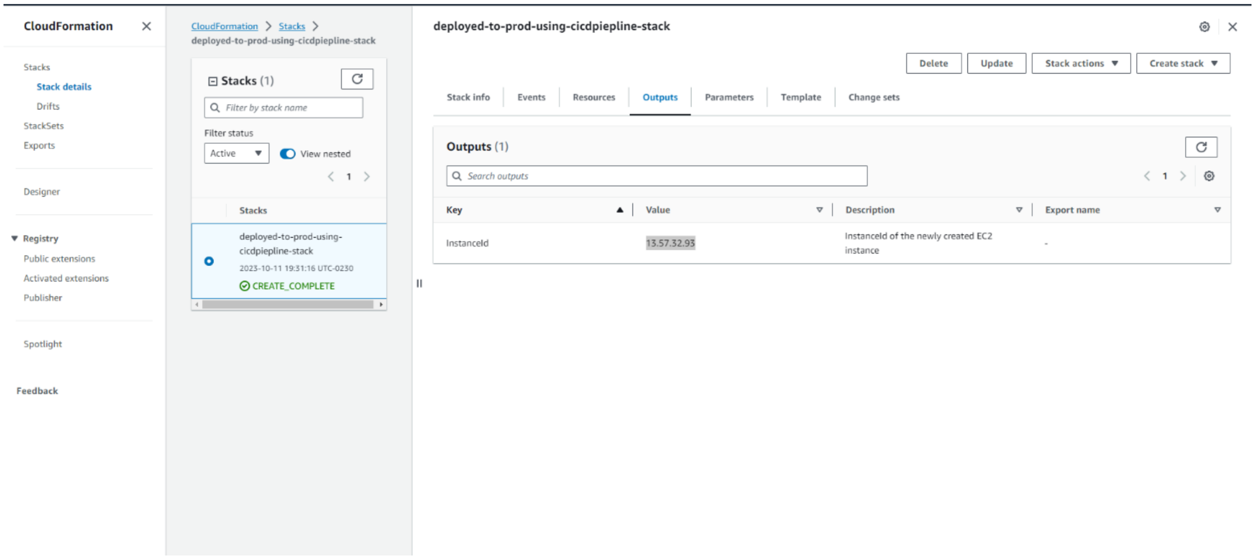
DeployToProd

AWS CloudFormation

Succeeded - Just now

77bda57f Source: adding buildspec to the codecommit repo

I confirmed its deployment to the Production Environment by going to the Production Environment Region “us-west-1” and copied the IP address of the output of the stack created in the Cloudformation and pasted it on a browser as shown below.



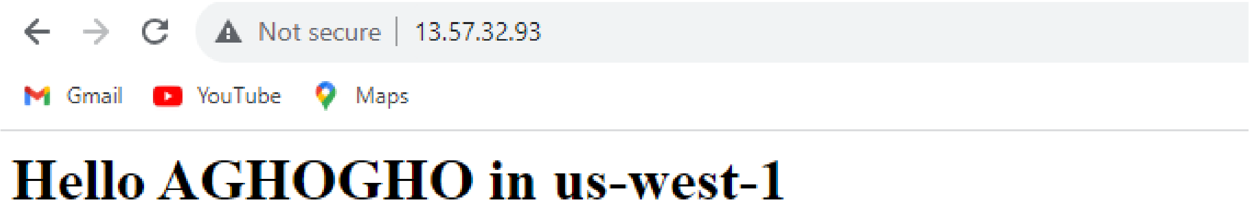
deployed-to-prod-using-cicdpiepline-stack

Stacks
deployed-to-prod-using-cicdpiepline-stack 2023-10-11 19:31:16 UTC-0230 CREATE_COMPLETE

Outputs (1)

Key	Value	Description	Export name
InstanceId	13.57.32.93	InstanceId of the newly created EC2 instance	-

It works fine in the Production Environment.



Hello AGHOGHO in us-west-1

Conclusion

I successfully built a three-stages pipeline to deploy an application using CloudFormation as a deploy provider.