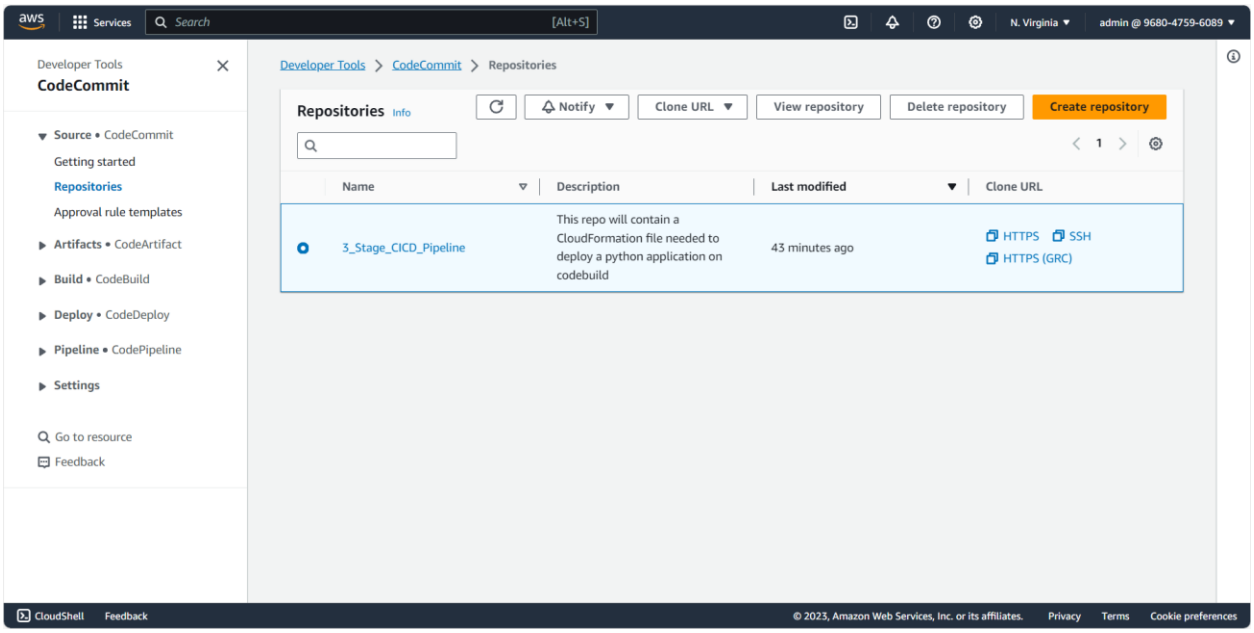
Nnadiekwe, Chiderah David

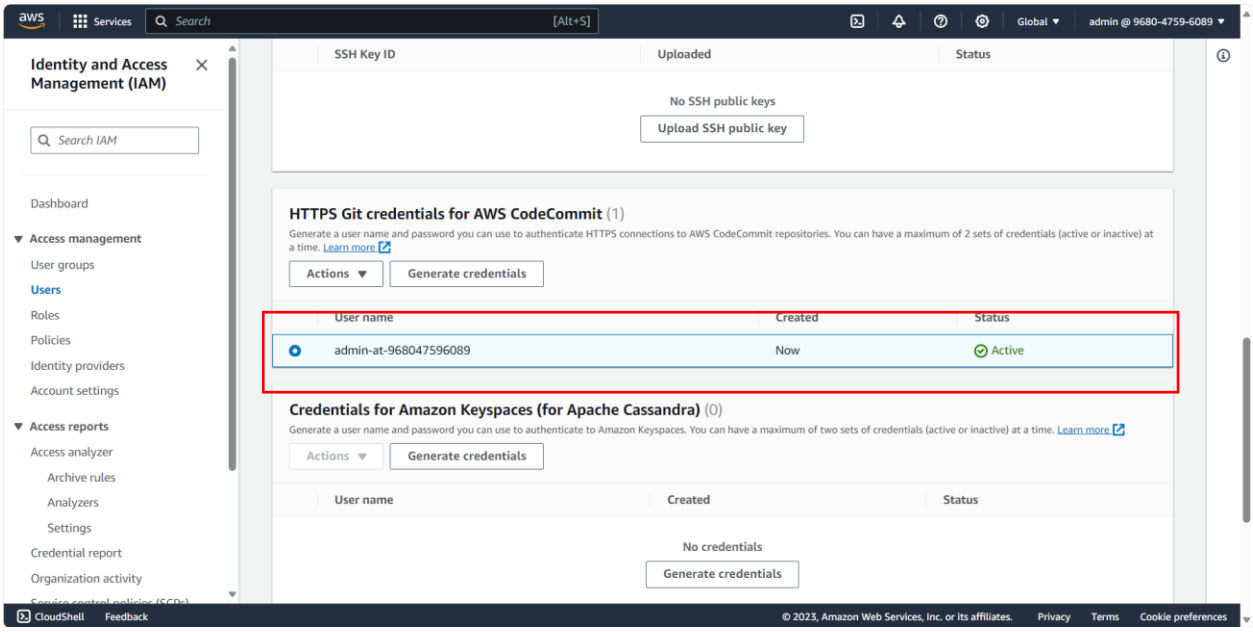
Project- *3 Stage CICD pipeline to deploy an application*

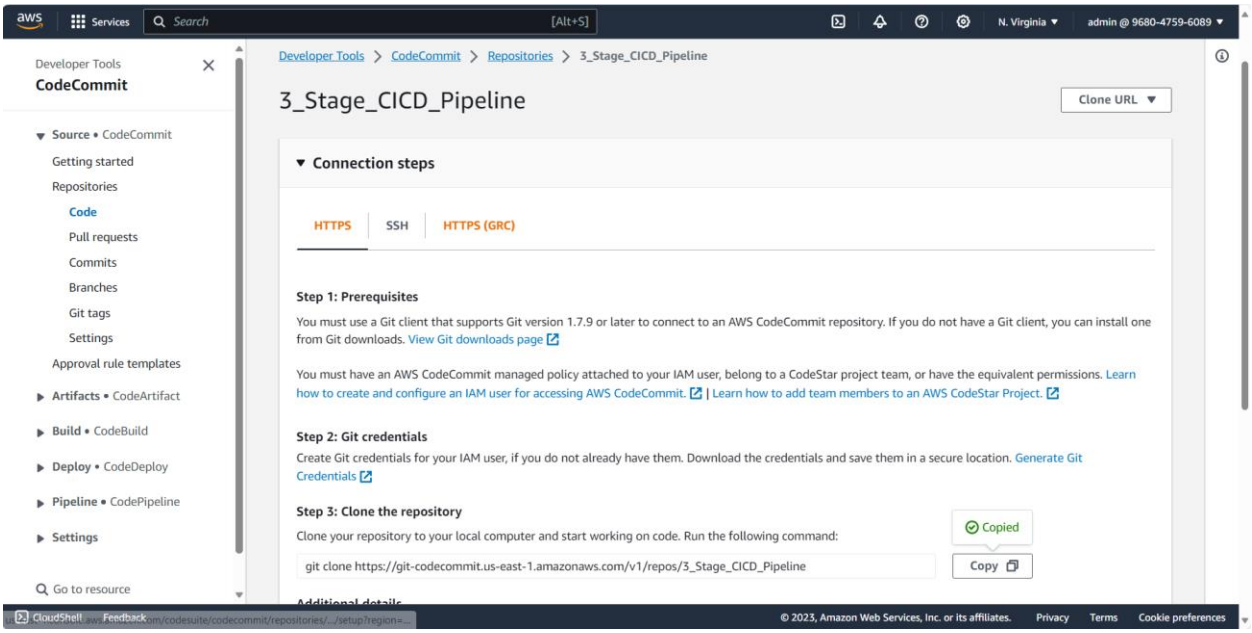
- **Build a three stages pipeline to deploy a given Application using CloudFormation as a deploy provider.**
- **Source your code from CodeCommit.**
- **Your application must be running on a custom VPC.**
- **Your application must be running on a custom VPC.**

- **First, I created a CodeCommit repository**

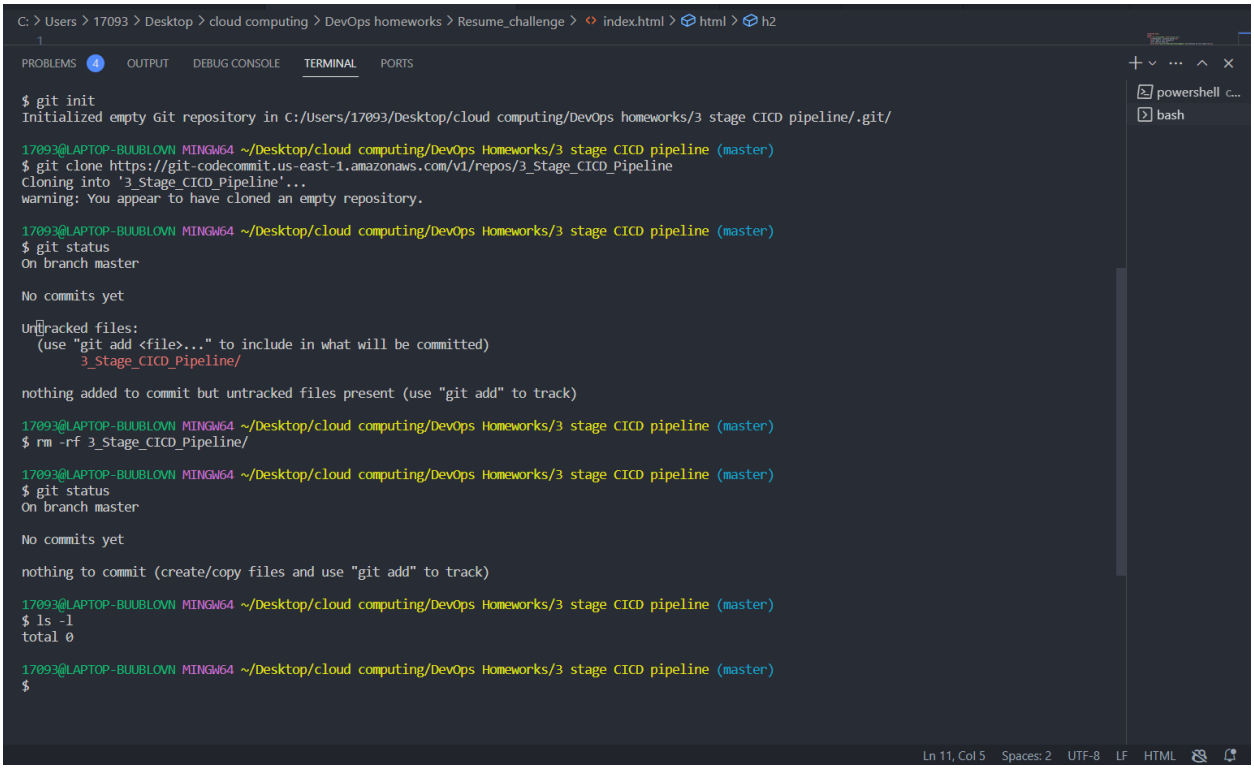


Then I created https git credentials for AWS code commit through IAM by selecting the user and then “security credentials”



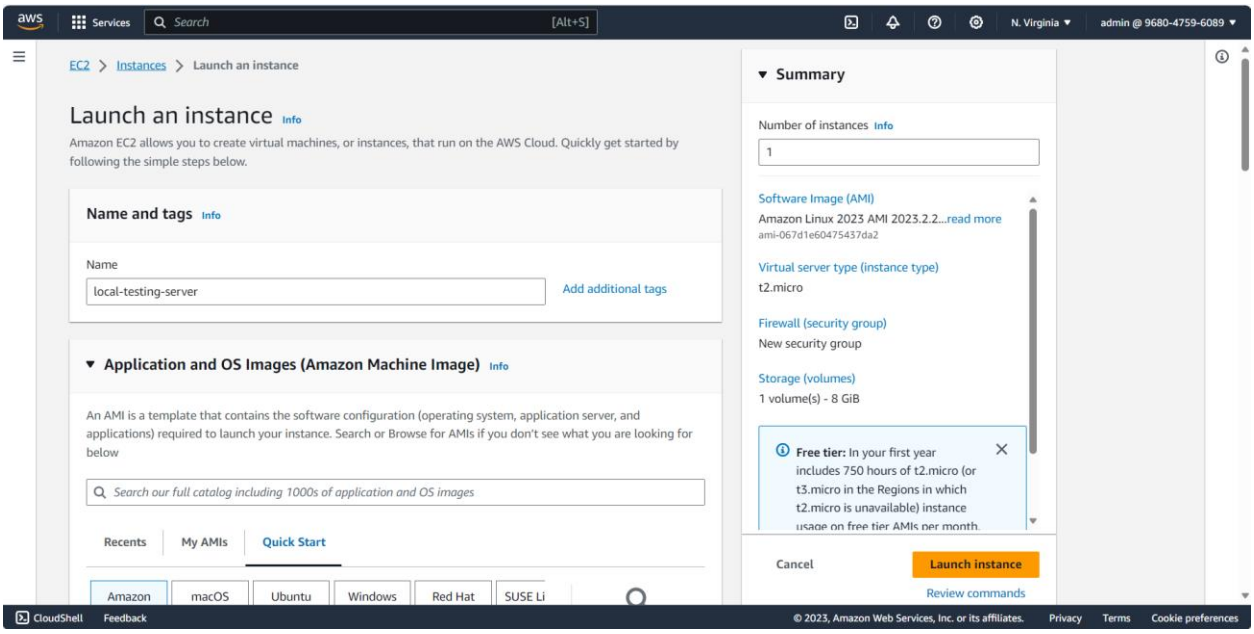


Next, I cloned the CodeComit repo to my local repo using the https git credentials for codecommit

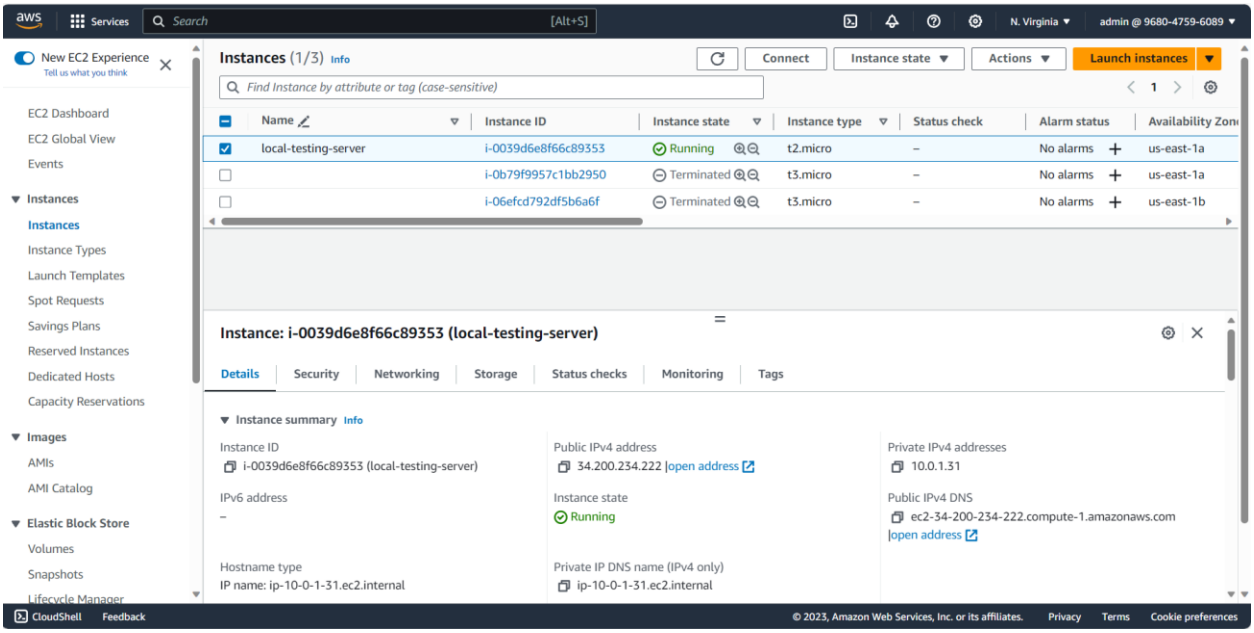


Successfully cloned!

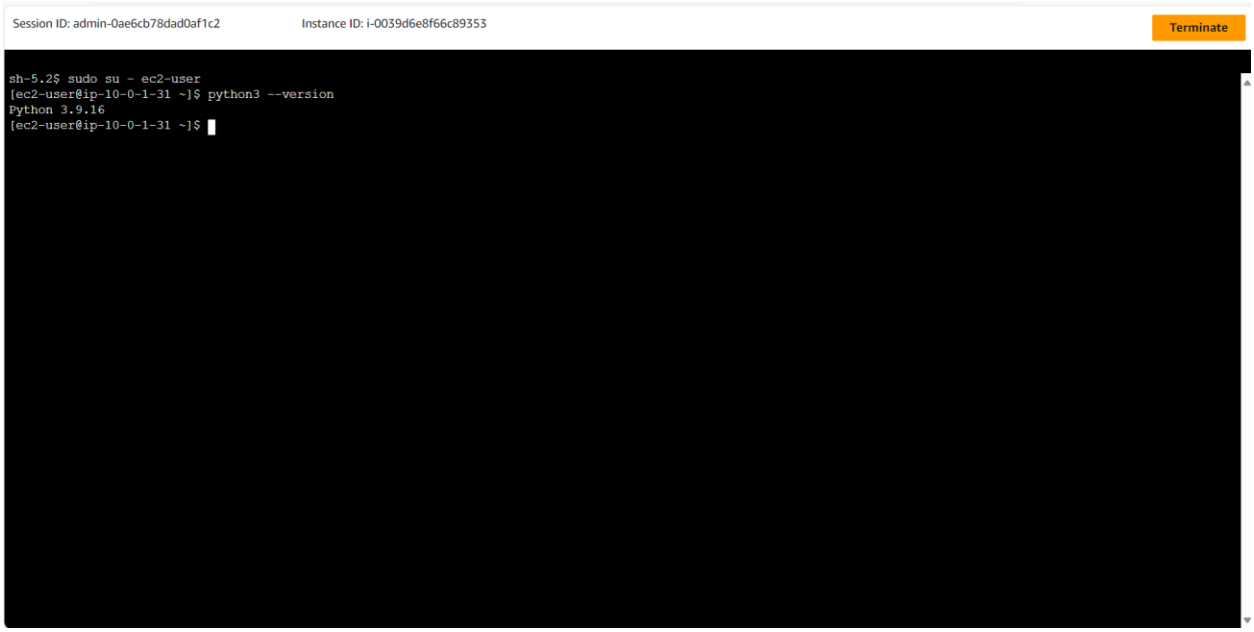
- Next, I created an instance in a custom VPC to manually test the cloudformation template using cfn-lint



The instance should have SSH enabled

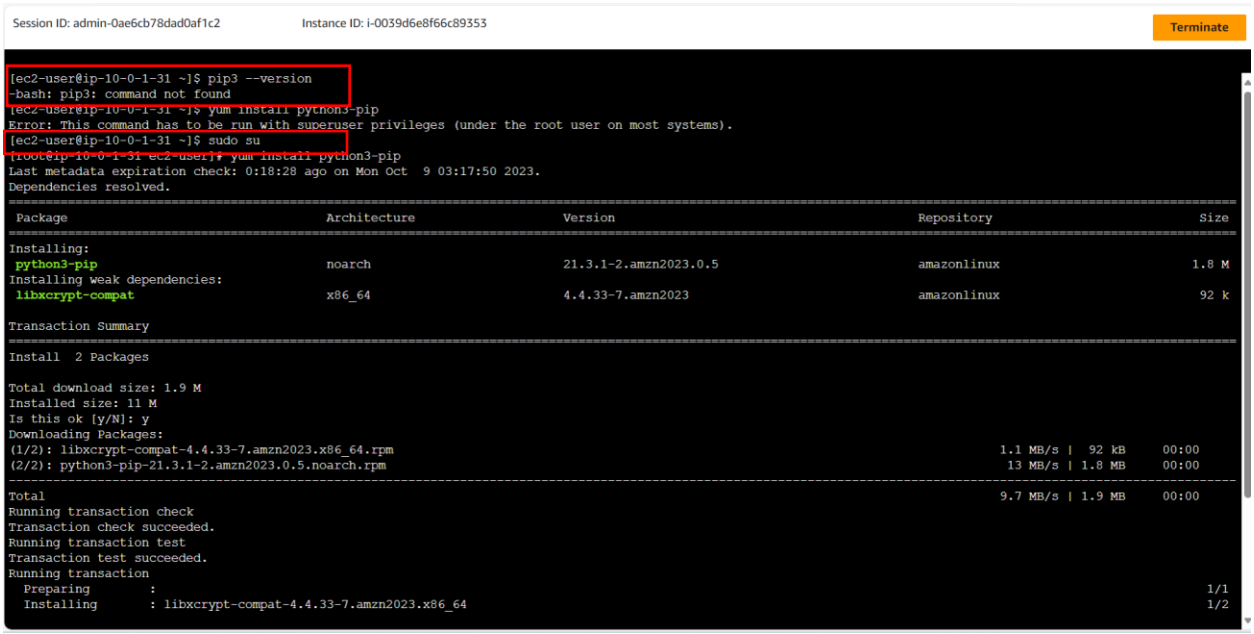


Next, I SSH into the instance using session manager to install cfn-lint. Cfn-lint requires python 3.7 and above to run



As seen, the python version in the Linux instance is 3.9.16 which meets the requirement to install cfn lint. So, we re good to go!

Next, we I checked if pip3 is installed. Pip3 is the package manager of python. I checked using the command “pip3 –version”. As seen, it says the “command is not found” hence, I had to install the pip3 using the command “yum install python3-pip” as a super user (sudo).



Now, I recheck the pip3 version using the “pip3 –version” command

```
 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                                                1.1 MB/s |  92 kB     00:00
(2/2): python3-pip-21.3.1-2.amzn2023.0.5.noarch.rpm                                                  13 MB/s | 1.8 MB     00:00
------------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                               9.7 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-10-0-1-31 ec2-user]# sudo su - ec2-user
Last login: Mon Oct  9 03:25:17 UTC 2023 on pts/0
[ec2-user@ip-10-0-1-31 ~]$ pip3 --version
pip 21.3.1 from /usr/lib/python3.9/site-packages/pip (python 3.9)
[ec2-user@ip-10-0-1-31 ~]$
```

Now, it shows that the version installed is pip 21.3.1

Next, I install the cfn-lint on the instance using the command “pip3 install cfn-lint”

Session ID: admin-0ae6cb78dad0af1c2 Instance ID: i-0039d6e8f66c89353 Terminate

```
[ec2-user@ip-10-0-1-31 ~]$ pip3 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 6.2 MB/s
Collecting networkx<4,>=2.4
  Downloading networkx-3.1-py3-none-any.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 18.5 MB/s
Collecting sympy>=1.0.0
  Downloading sympy-1.12-py3-none-any.whl (5.7 MB)
     |████████████████████████████████| 5.7 MB 50.6 MB/s
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 57.6 MB/s
Requirement already satisfied: jsonpatch in /usr/lib/python3.9/site-packages (from cfn-lint) (1.21)
Requirement already satisfied: pyyaml>5.4 in /usr/lib64/python3.9/site-packages (from cfn-lint) (5.4.1)
Collecting sarif-om~=1.0.4
  Downloading sarif_om-1.0.4-py3-none-any.whl (30 kB)
Collecting jschema-to-python~=1.2.3
  Downloading jschema_to_python-1.2.3-py3-none-any.whl (10 kB)
Collecting aws-sam-translator>=1.75.0
  Downloading aws_sam_translator-1.77.0-py3-none-any.whl (377 kB)
     |████████████████████████████████| 377 kB 59.2 MB/s
Collecting junit-xml~=1.9
  Downloading junit_xml-1.9-py2.py3-none-any.whl (7.1 kB)
Requirement already satisfied: jsonschema<5,>=3.0 in /usr/lib/python3.9/site-packages (from cfn-lint) (3.2.0)
Collecting typing-extensions<5,>=4.4
  Downloading typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Collecting pydantic<3,>=1.8
  Downloading pydantic-2.4.2-py3-none-any.whl (395 kB)
     |████████████████████████████████| 395 kB 55.5 MB/s
Collecting boto3==1.*,>=1.19.5
  Downloading boto3-1.28.62-py3-none-any.whl (135 kB)
     |████████████████████████████████| 135 kB 57.9 MB/s
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)
```

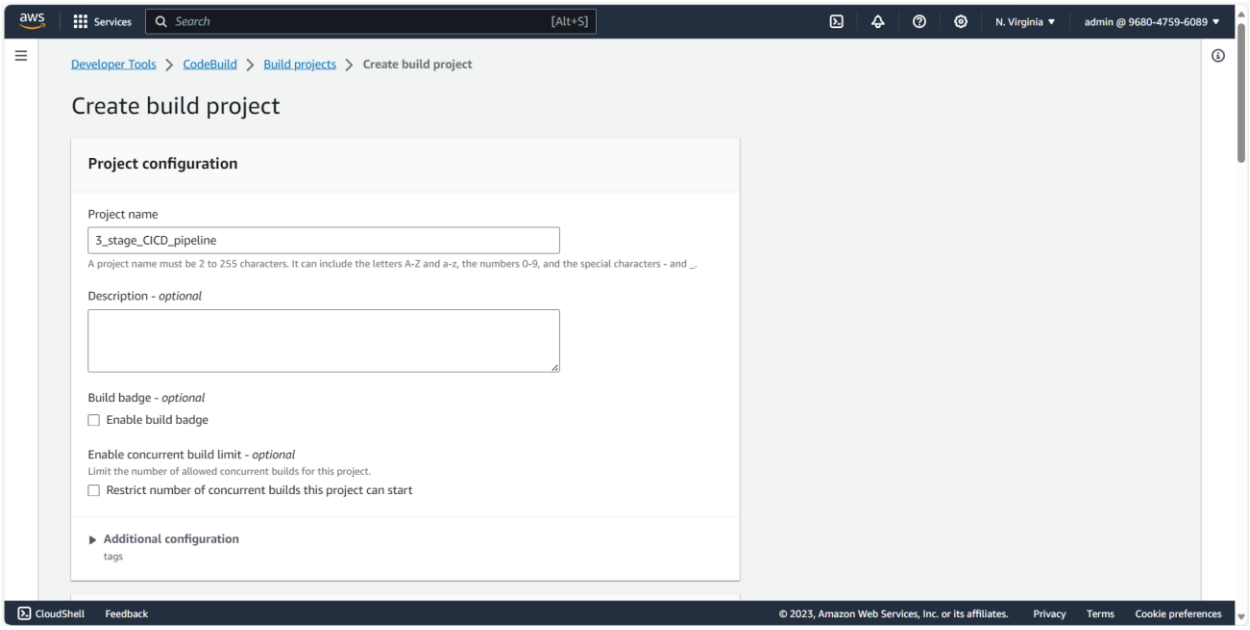
Verifying the cfn-lint installation

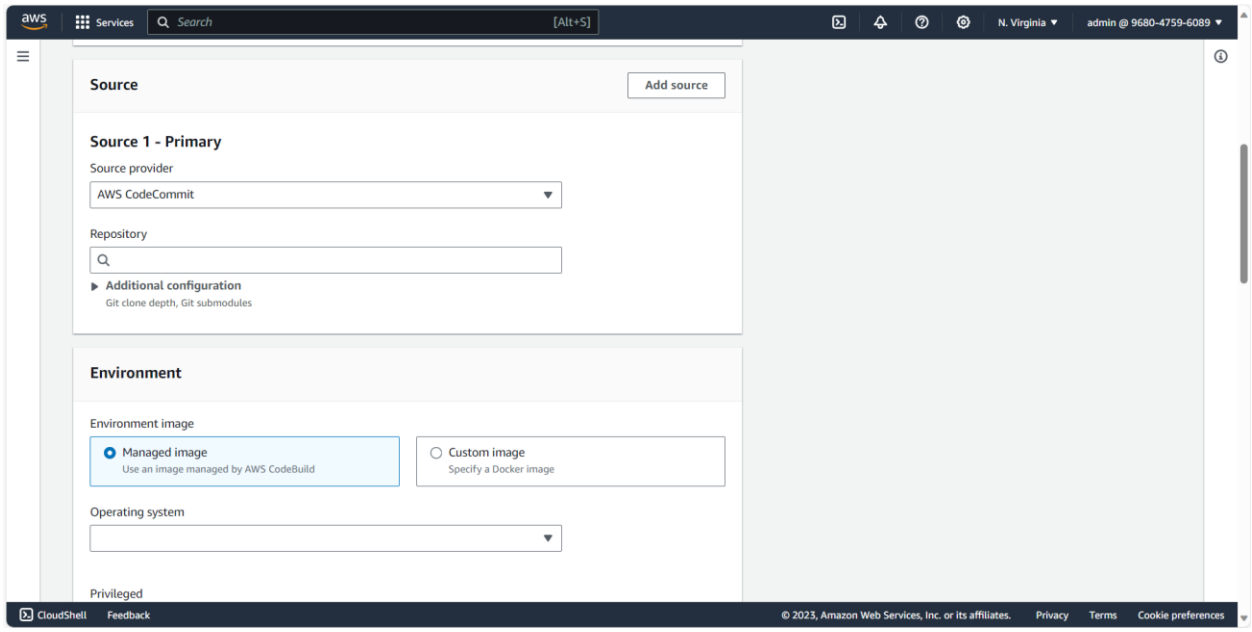
```
[ec2-user@ip-10-0-1-31 ~]$ cfn-lint --version
cfn-lint 0.81.0
[ec2-user@ip-10-0-1-31 ~]$
```

Testing my cloudformation template using cfn-lint in the ec2. My cloudformation template is “launch_instance_in_custom_vpc.yml”

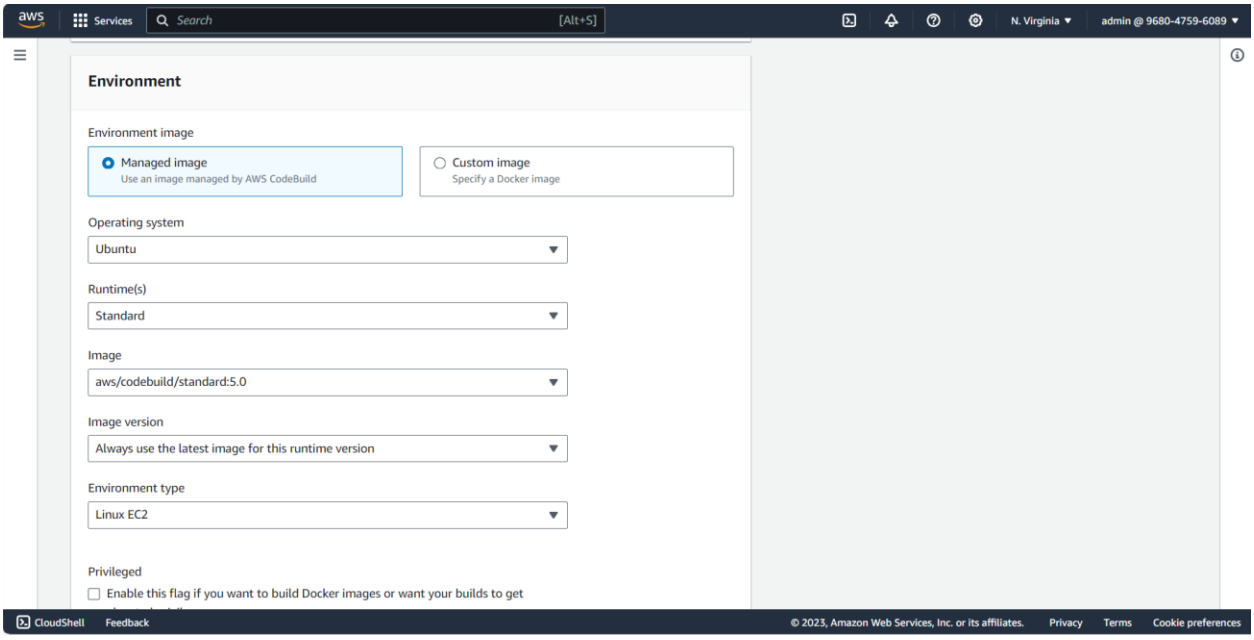
```
Outputs:
  VPCId:
    Description: VPC ID
    Value: !Ref VPC
    Export:
      Name: !Sub '${AWS::StackName}-VPCID'
  PublicSubnet:
    Description: The subnet ID to use for public web servers
    Value: !Ref PublicSubnet
    Export:
      Name: !Sub '${AWS::StackName}-SubnetID' #replacing a string
  WebServerSecurityGroup:
    Description: The security group ID to use for public web servers
    Value: !GetAtt
      - WebServerSecurityGroup
      - GroupId
    Export:
      Name: !Sub '${AWS::StackName}-SecurityGroupID'



[root@ip-10-0-1-31 bin]# vim launch_instance_in_custom_vpc.yml
[root@ip-10-0-1-31 bin]# cfn-lint launch_instance_in_custom_vpc.yml
E2002 Parameter AvailabilityZone has invalid type AWS::EC2::AvailabilityZone::us-east-1a
launch_instance_in_custom_vpc.yml:19:5

E3008 Property "AvailabilityZone" can Ref to parameter of types [String, AWS::SSM::Parameter::Value<String>, AWS::EC2::AvailabilityZone::Name, AWS::SSM::Parameter::Va
lue<AWS::EC2::AvailabilityZone::Name>] at Resources/PublicSubnet/Properties/AvailabilityZone/Ref
launch_instance_in_custom_vpc.yml:38:7
[root@ip-10-0-1-31 bin]# vim launch_instance_in_custom_vpc.yml
[root@ip-10-0-1-31 bin]# cfn-lint launch_instance_in_custom_vpc.yml
[root@ip-10-0-1-31 bin]#
```

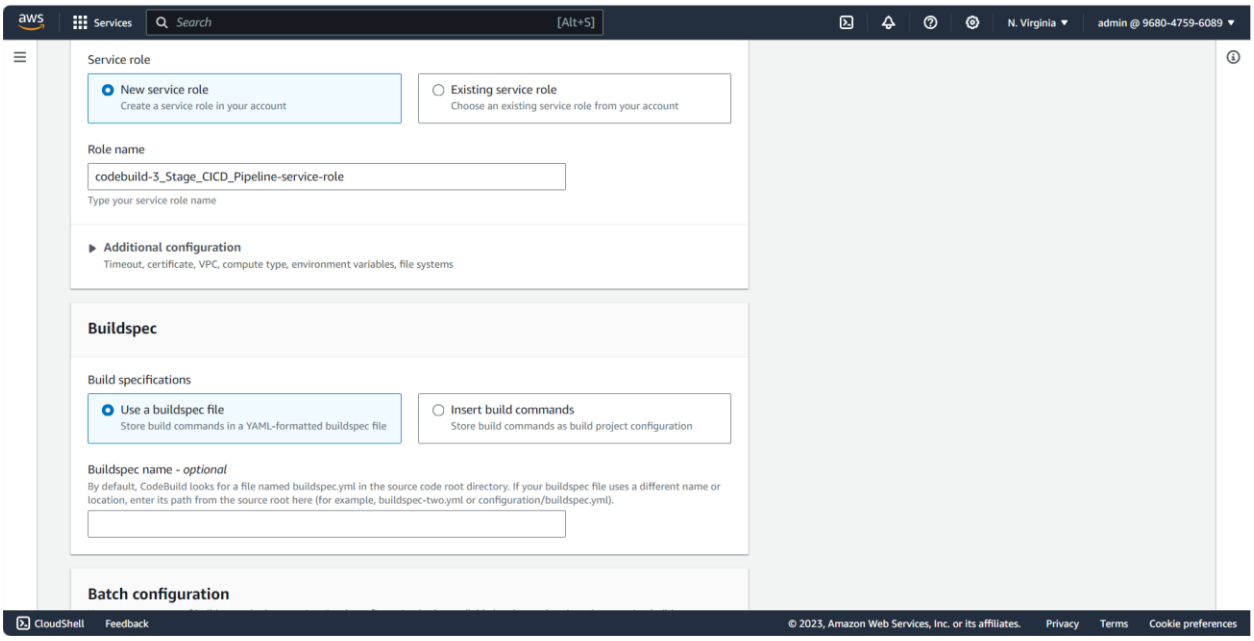
- **Next, I created the CodeBuild project**

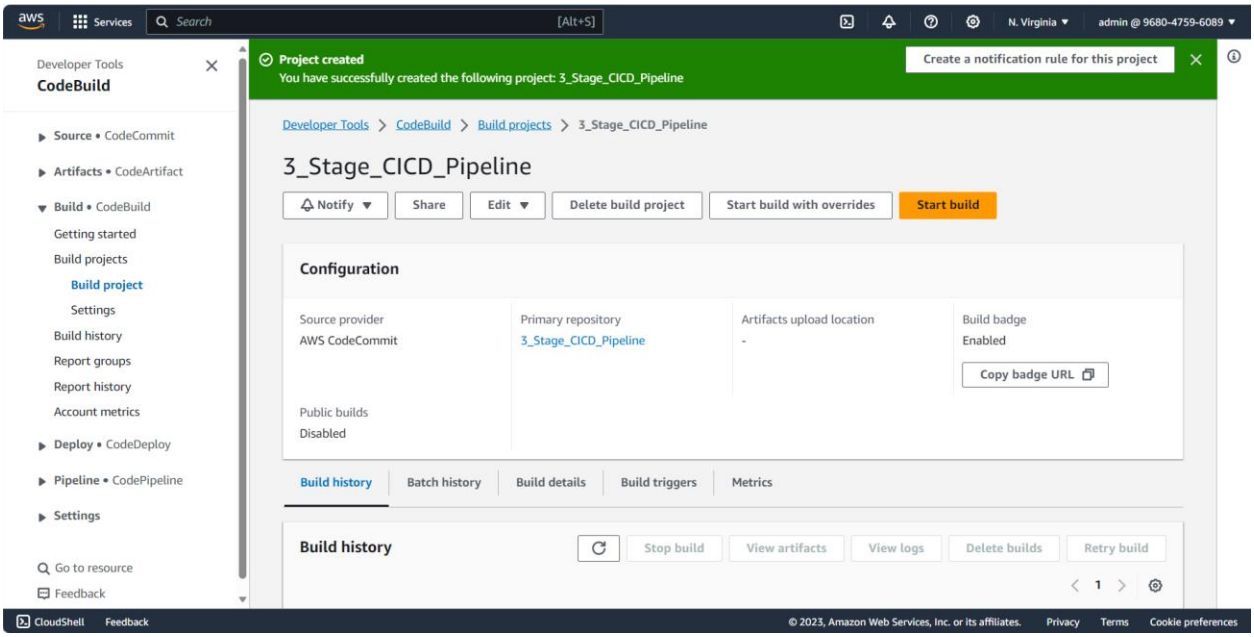




For this project, I utilized Ubuntu Environment







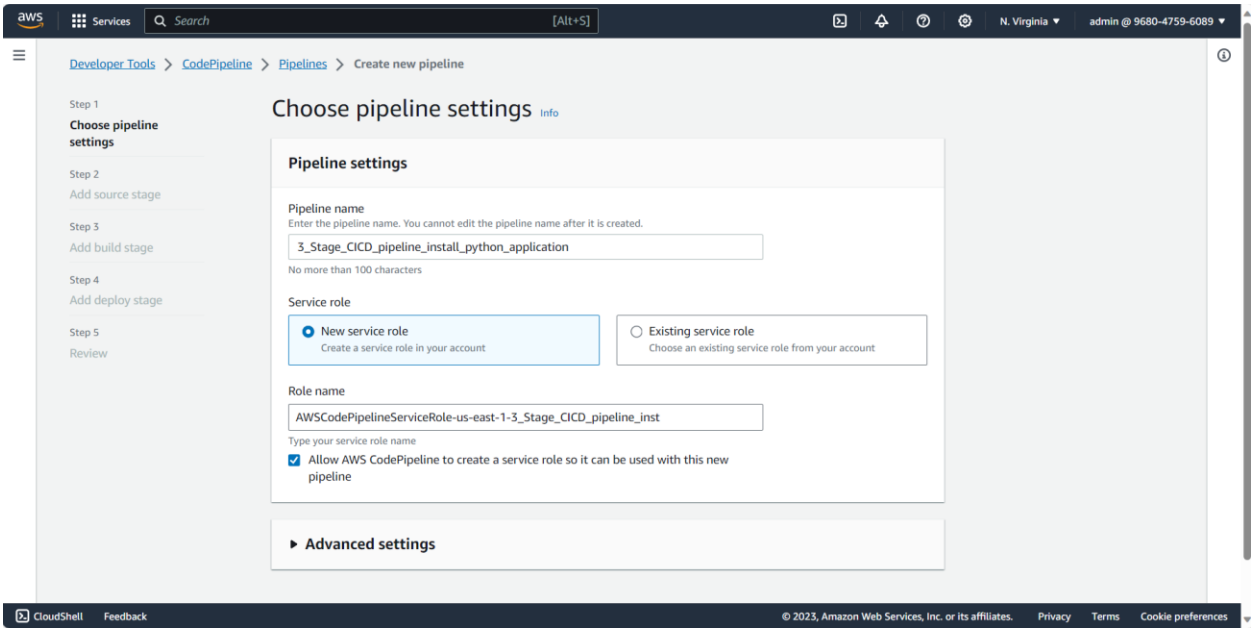
The CodeBuild project has been built successfully!

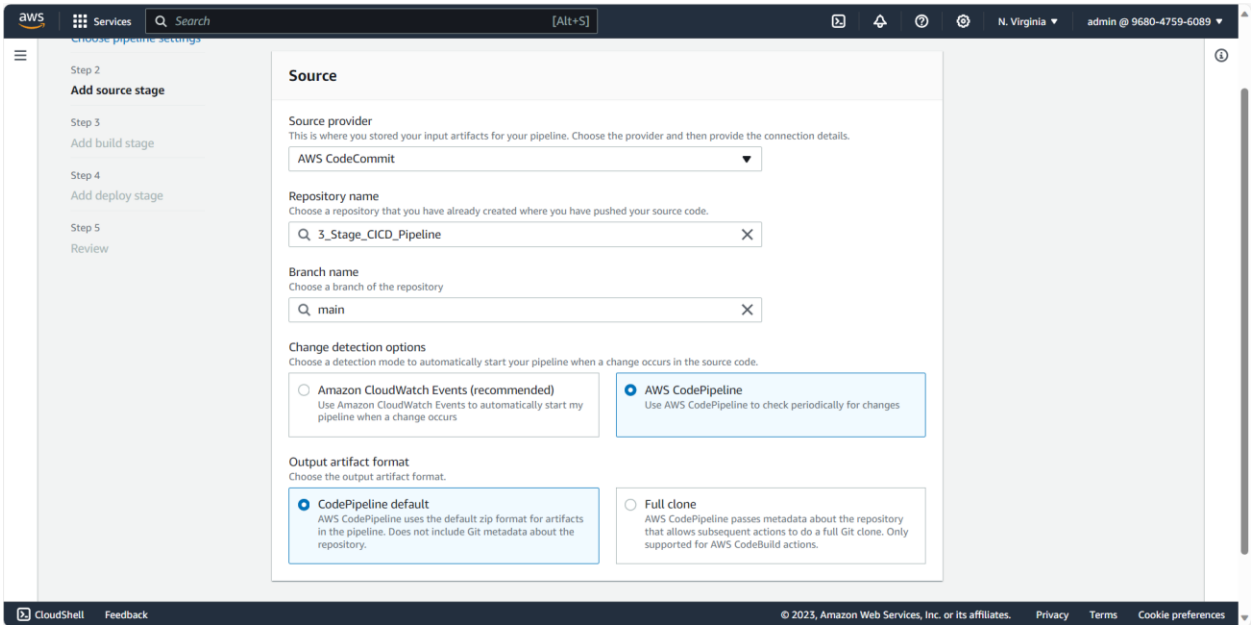
- **Next, I created the BuildSpec**

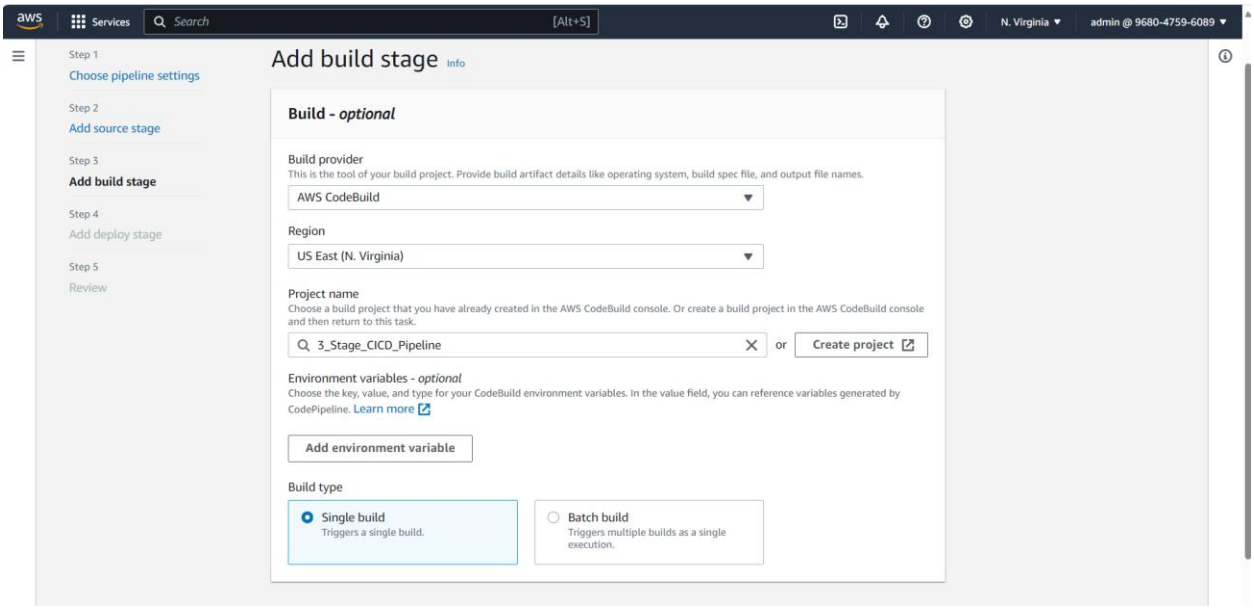
I created a buildspec for the build stage of the pipeline “CodeBuild”. I created a BuildSpec YAML file which contains the command to test install cfn-lint and test a cloudformation template. In order to successfully achieve creating the BuildSpec devoid of errors, I referenced the AWS documentation https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html

```yaml
version: 0.2
phases:
  install:
    on-failure: ABORT
    runtime-versions:
      python: 3.8 #installing python version 3.8
  pre_build:
    on-failure: CONTINUE
    commands:
      - echo ****Checking if python installed successfully****
      - python3 --version #will return the version of python that has been
installed
      - echo ***Checking if pip3 is installed***
      - pip3 --version
      - echo ***installing cfn-lint***
      - pip3 install cfn-lint
      - echo ***checking if cfn-lint is installed***
      - cfn-lint --version
  build:
    on-failure: ABORT
    commands:
      - echo ***Build started at `date` ***
      - pwd
      - cfn-lint launch_instance_in_custom_vpc.yml
  post_build:
    on-failure: ABORT
    commands:
      - echo ***Build complete***
      - ls -l
artifacts:
  files:
    - '**/*'
```

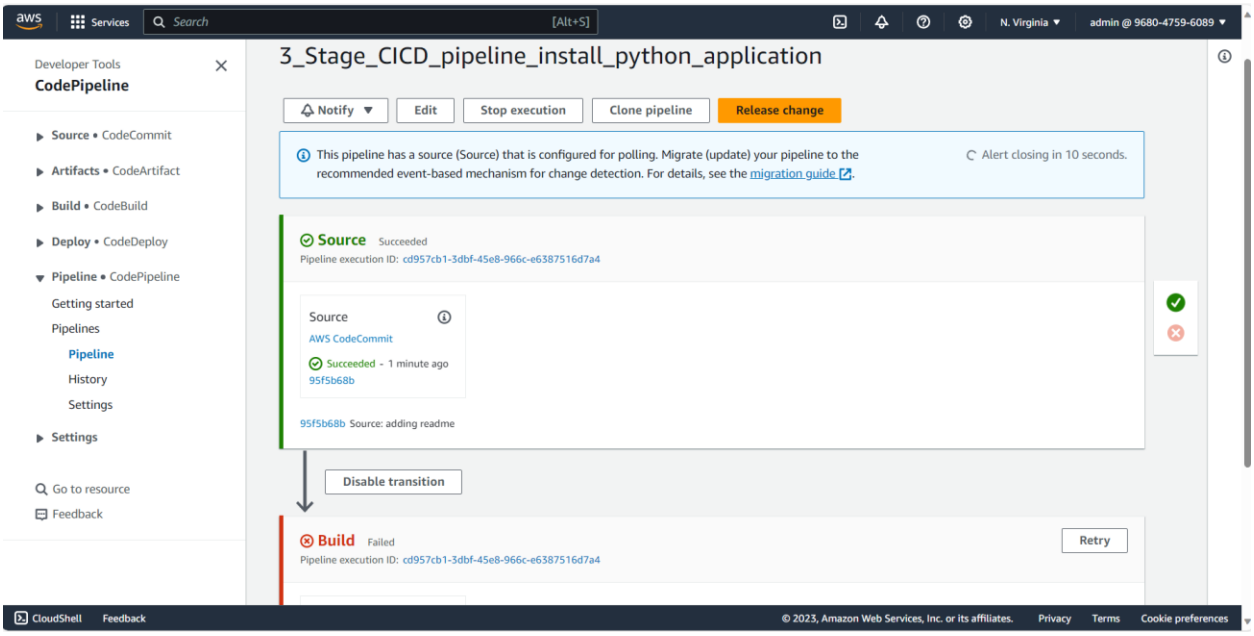
- **Next, I created the CodePipeline**





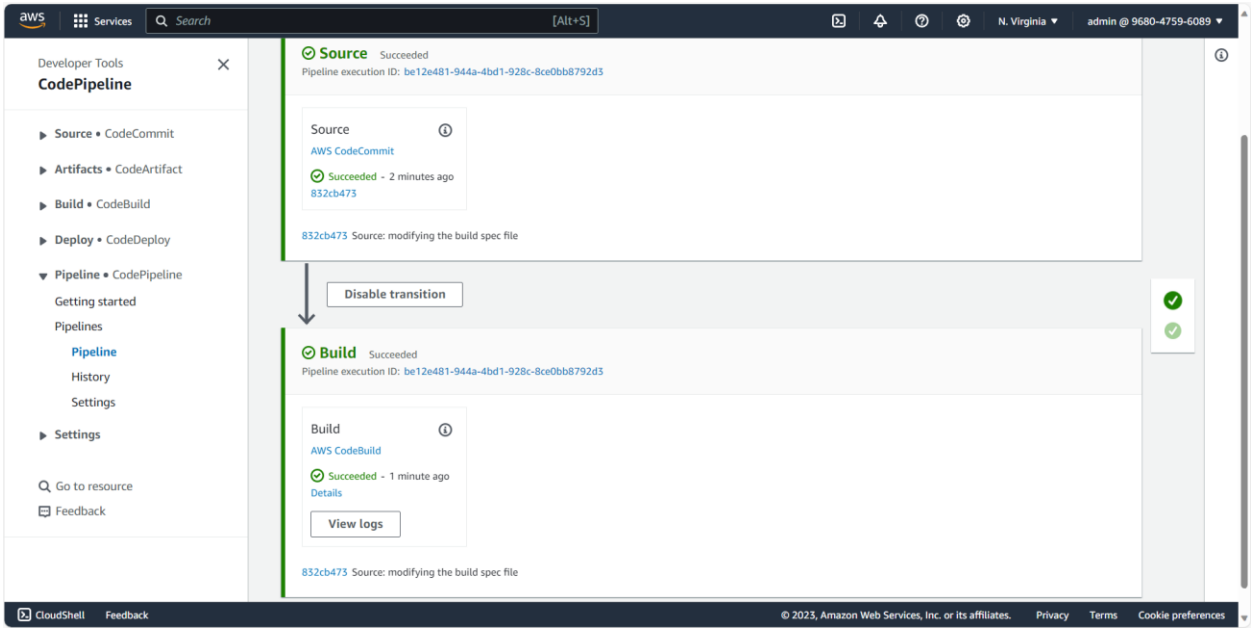


The build stage failed because I have NOT yet pushed the buildspec to the codecommit repository!



Next, I did a git push from the local repo the code commit repository! I pushed the buildspec file “buildspec.yml” and the cloudformation template “launch_instance_in_custom_vpc.yml” to the codecommit repo.

The Build Stage is now successful! 😊

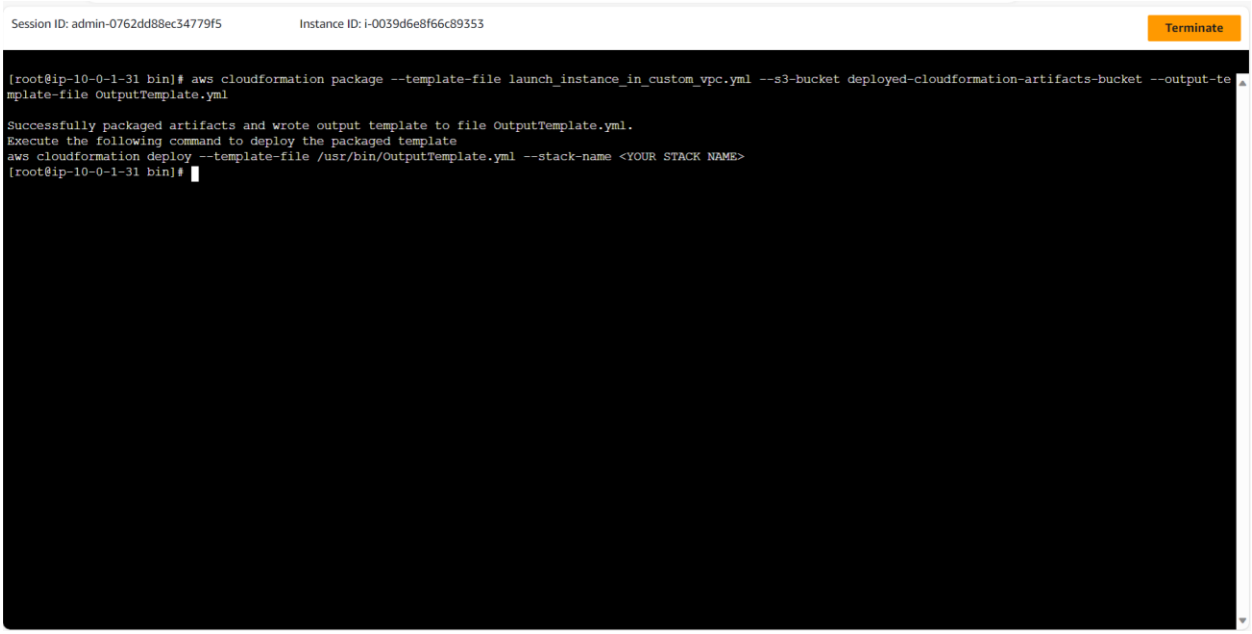


- **The Cloudformation packaging for deployment**

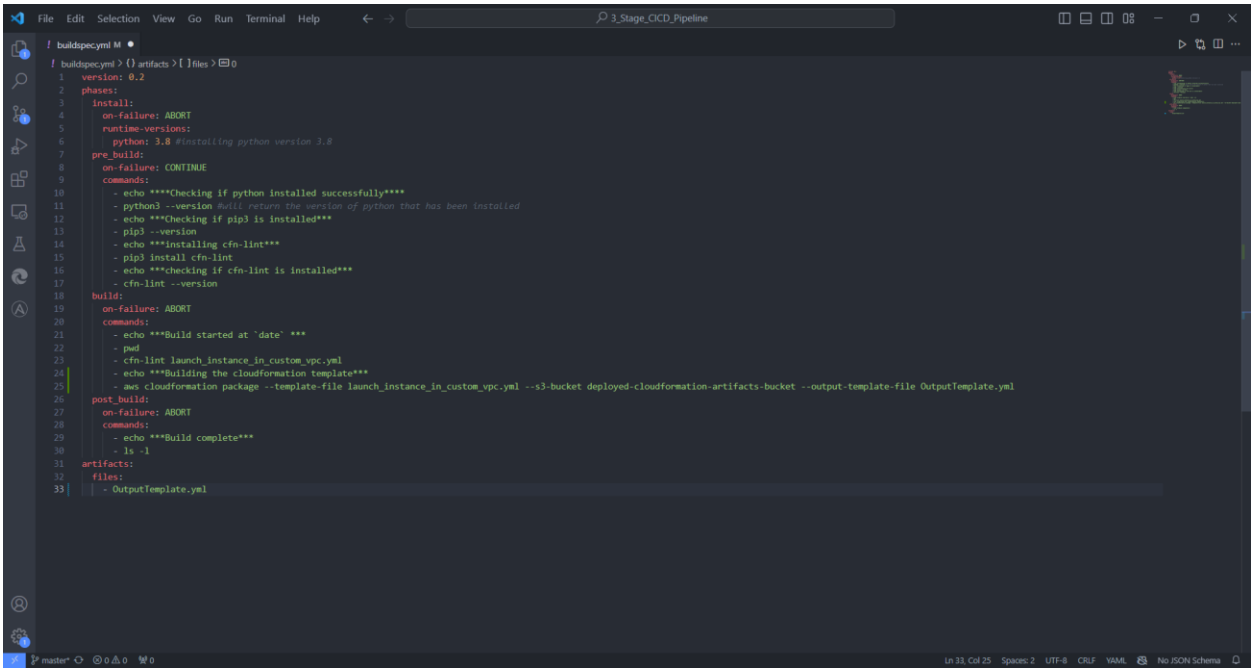
  In order to deploy the Cloudformation template, we need to install a cloudformation package during the build stage. Hence, we need to modify the buildspec file.
  I referenced this aws documentation to achieve this
  https://docs.aws.amazon.com/cli/latest/reference/cloudformation/package.html

  In this stage, I first manually tested the command below in the linux instance to be sure it works before passing the command into the buildspec
  "aws cloudformation package --template-file launch_instance_in_custom_vpc.yml --s3-bucket deployed-cloudformation-artifacts-bucket --output-template-file OutputTemplate.yml"

  This command will launch an s3 bucket where it will store the artifact ready to be deployed.

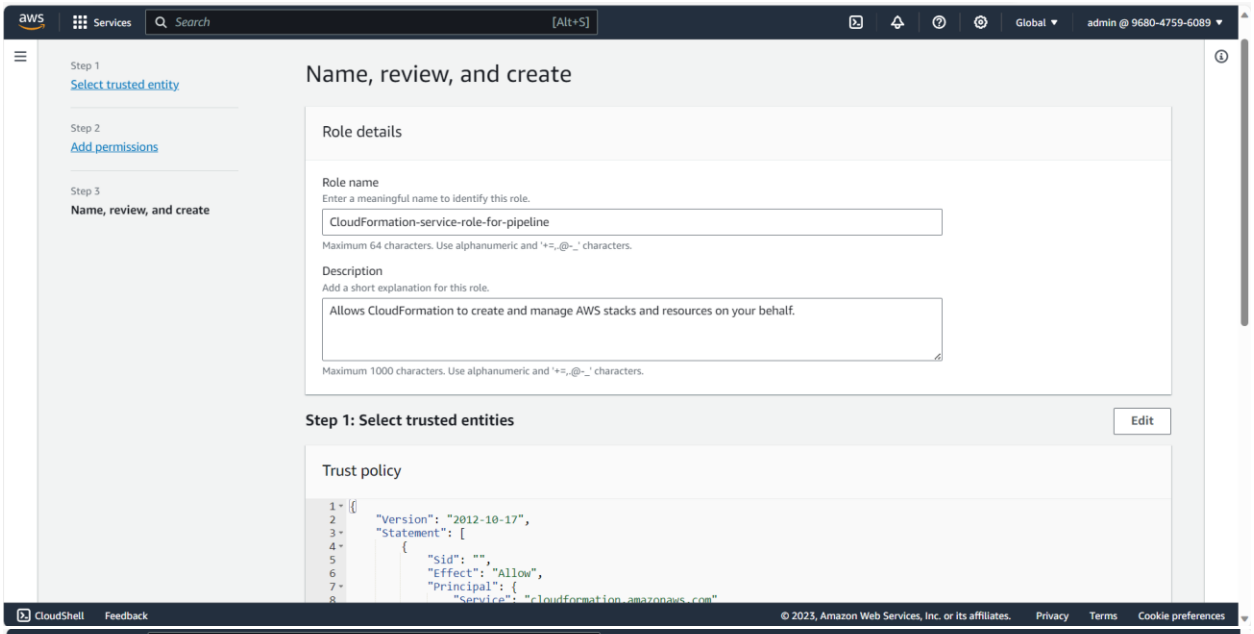


Since the command is successful, I then passed the command to the BuildSpec

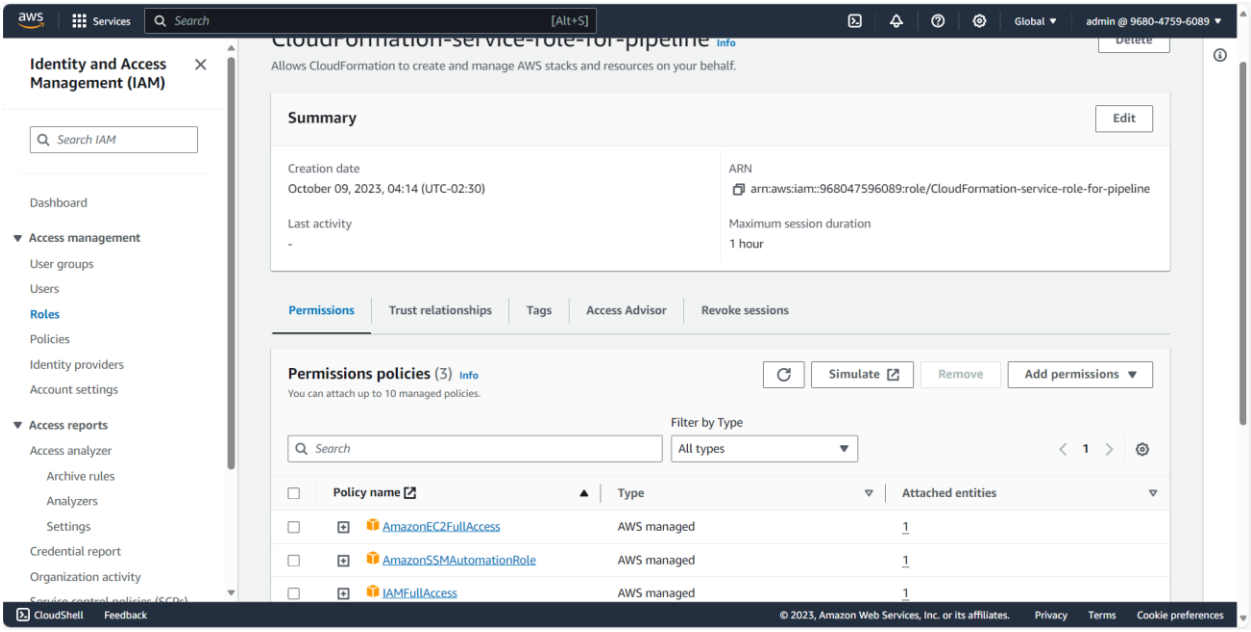


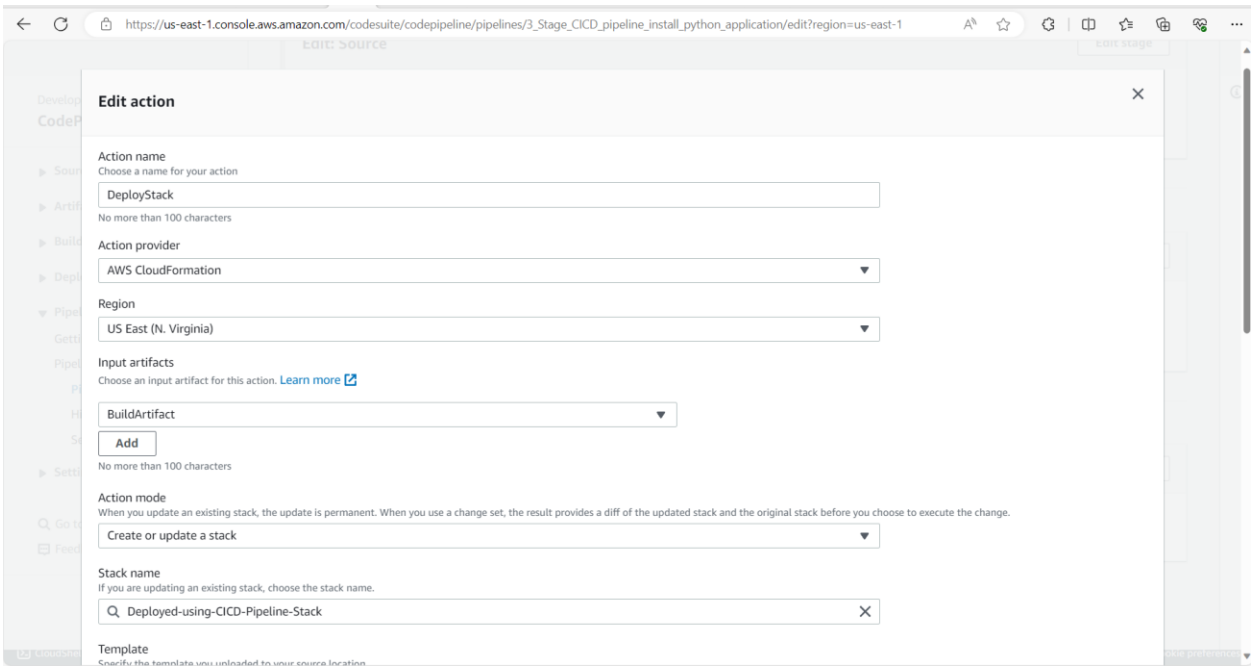
- **Adding the third stage of the pipeline- The Deploy stage**

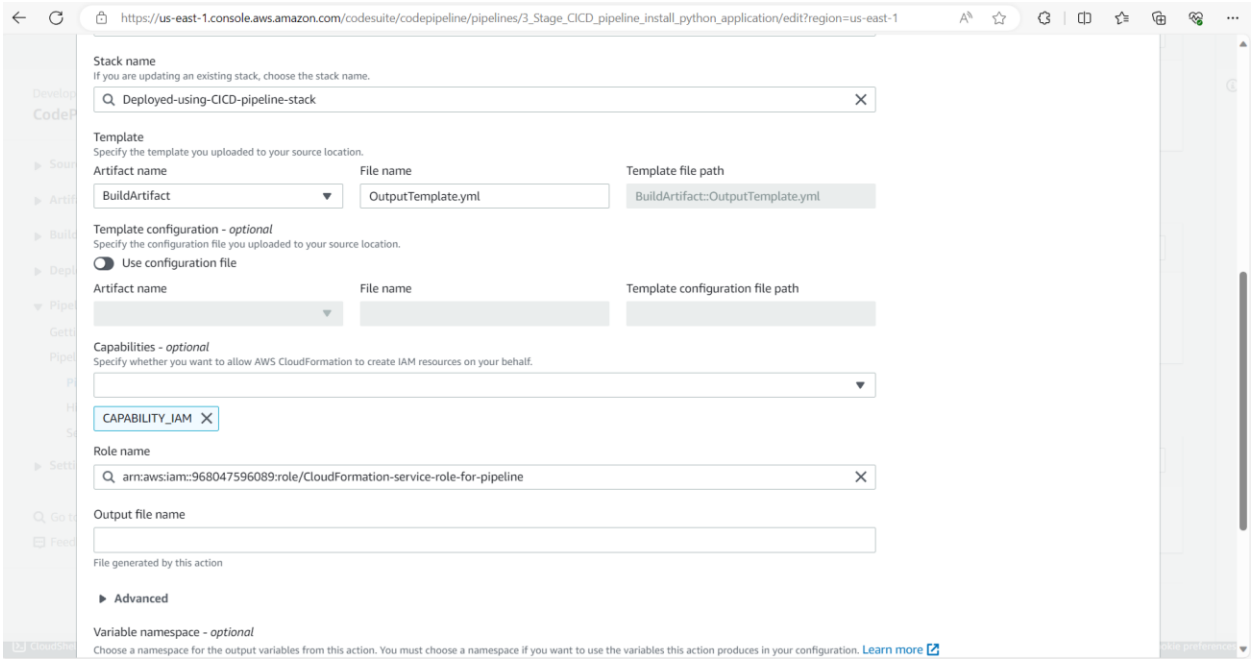
I created an IAM role using Cloudformation as the principal to enable cloudformation launch resources on my behalf.



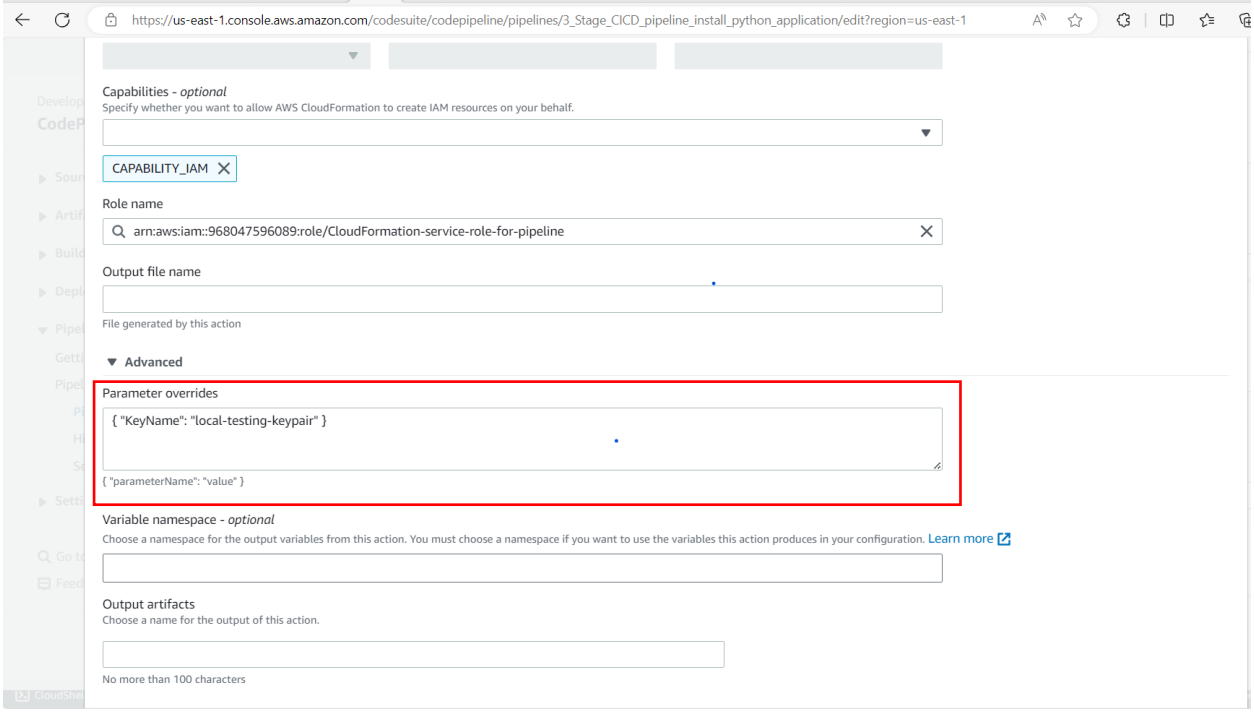
Next, I edited the Pipeline and added the deploy stage



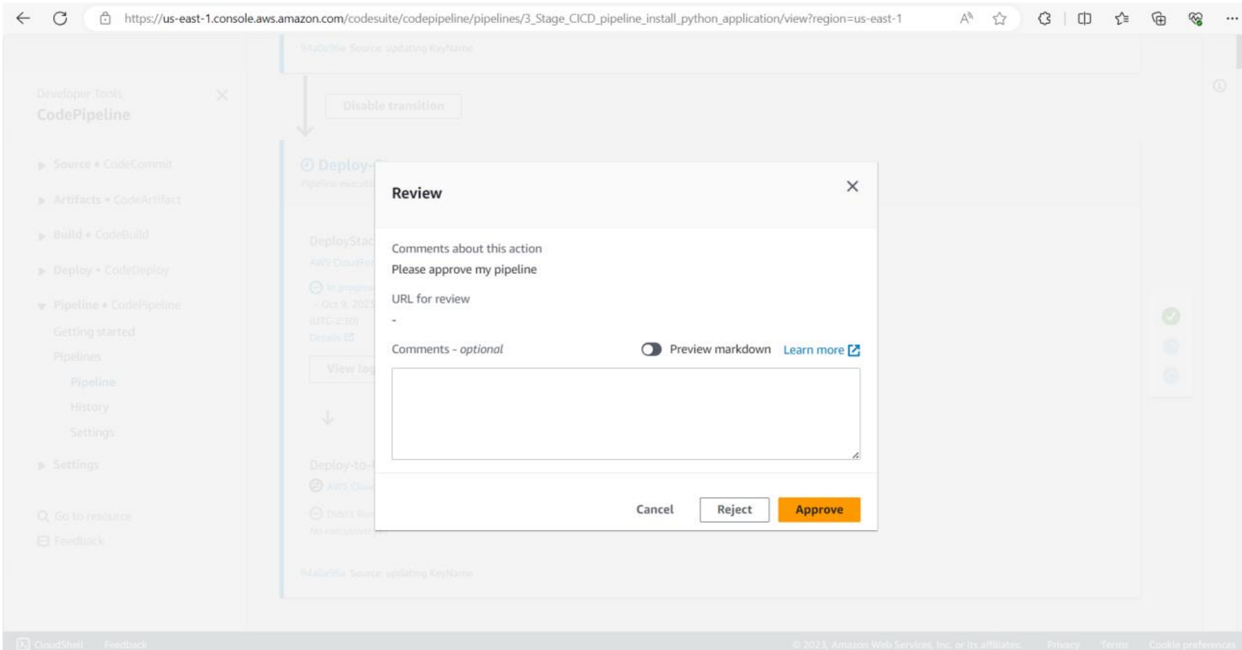


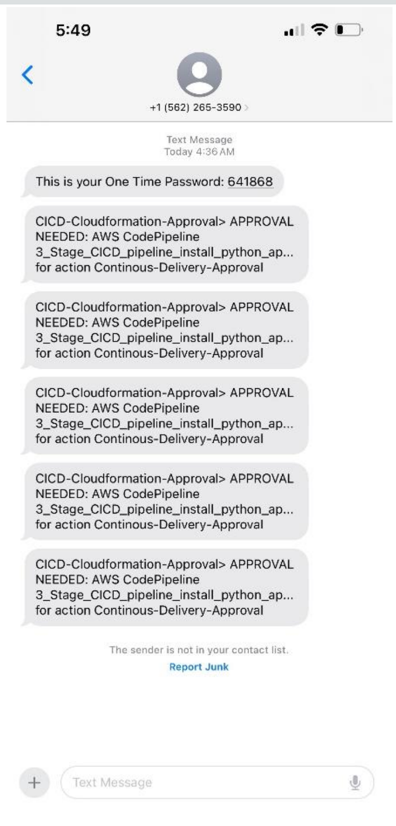


Next, I passed the keypair name from the pipeline into the cloudformation template



For this pipeline, I will be performing a “continuous Delivery” which requires approval before the resources are deployed. Hence, I setup another Action in the deploy stage to approve deployments by leveraging SNS.

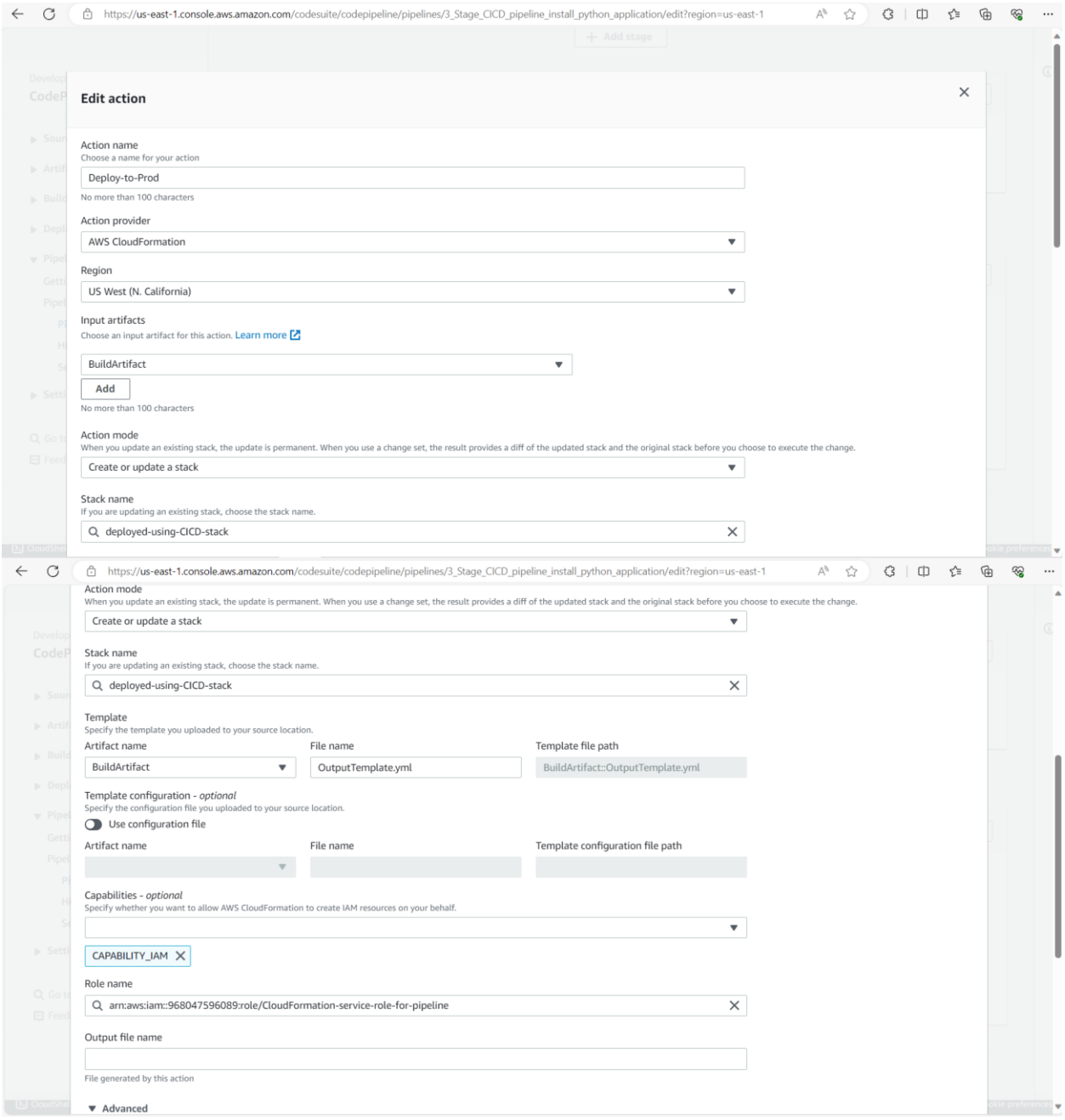

https://us-east-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/3_Stage_CICD_pipeline_install_python_application/view?region=us-east-1
Review
Comments about this action
Please approve my pipeline
URL for review
-
Comments - optional
Preview markdown
Learn more
Cancel
Reject
Approve

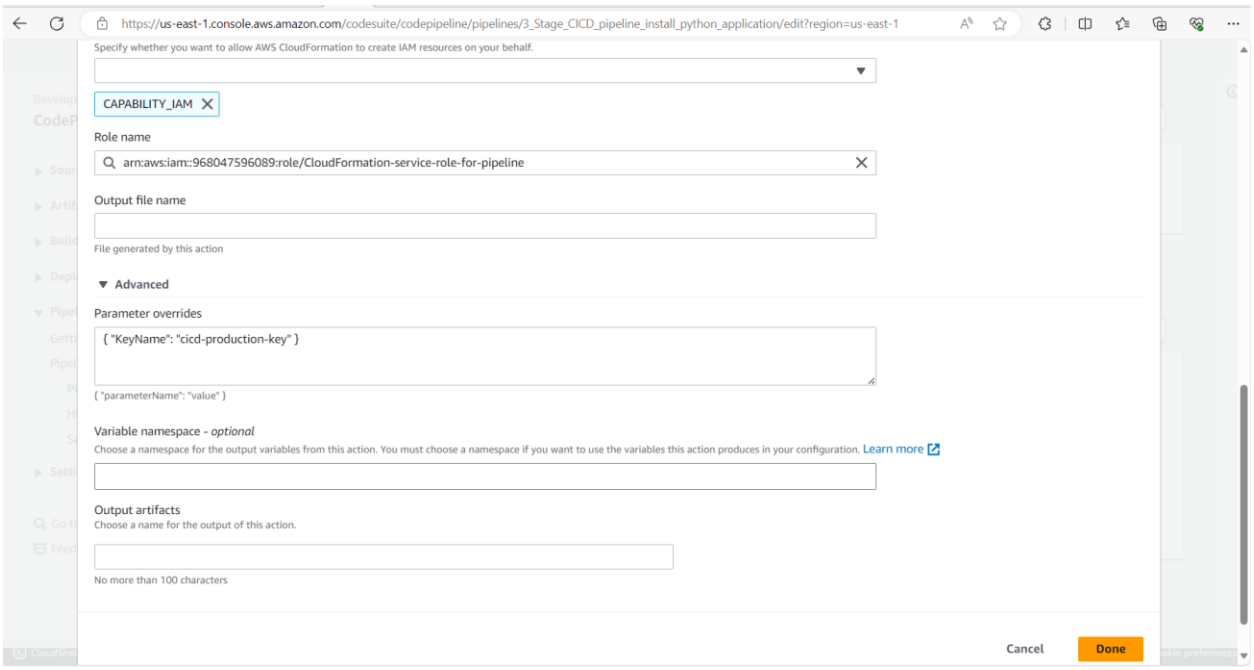

5:49
+1 (562) 265-3590
Text Message
Today 4:36 AM
This is your One Time Password: 641868
CICD-Cloudformation-Approval> APPROVAL NEEDED: AWS CodePipeline 3_Stage_CICD_pipeline_install_python_ap... for action Continous-Delivery-Approval
CICD-Cloudformation-Approval> APPROVAL NEEDED: AWS CodePipeline 3_Stage_CICD_pipeline_install_python_ap... for action Continous-Delivery-Approval
CICD-Cloudformation-Approval> APPROVAL NEEDED: AWS CodePipeline 3_Stage_CICD_pipeline_install_python_ap... for action Continous-Delivery-Approval
CICD-Cloudformation-Approval> APPROVAL NEEDED: AWS CodePipeline 3_Stage_CICD_pipeline_install_python_ap... for action Continous-Delivery-Approval
CICD-Cloudformation-Approval> APPROVAL NEEDED: AWS CodePipeline 3_Stage_CICD_pipeline_install_python_ap... for action Continous-Delivery-Approval
The sender is not in your contact list.
Report Junk
Text Message

Next, I set up the deploy to production stage

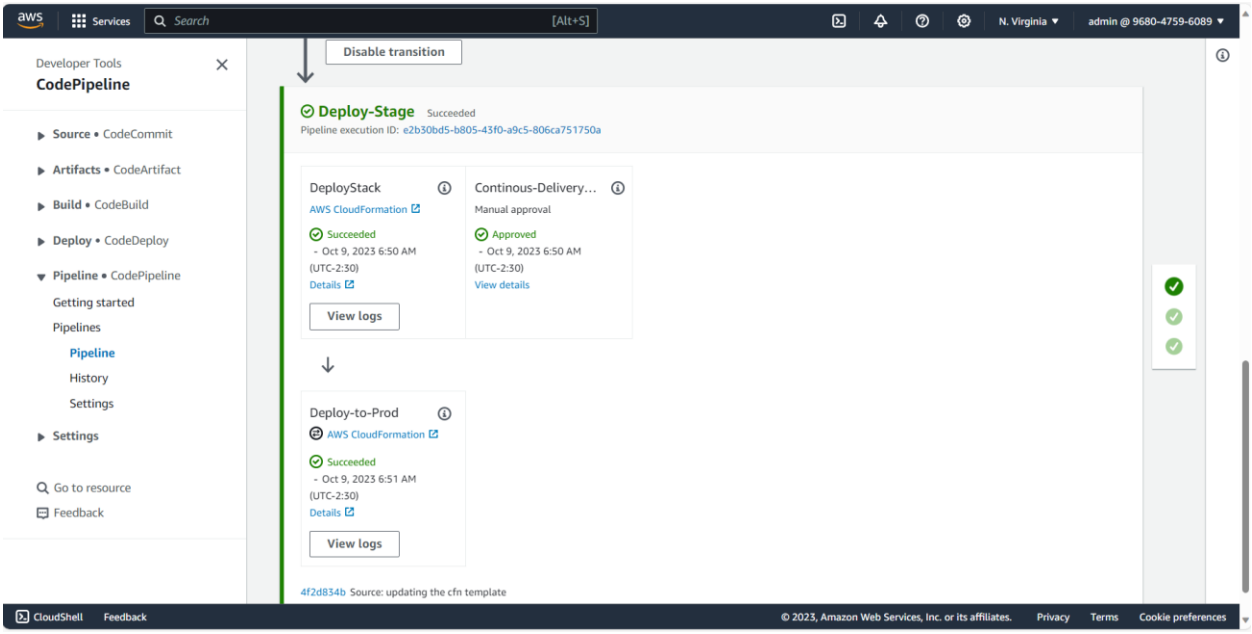
NB: -This is only available in **US-WEST-1 region**

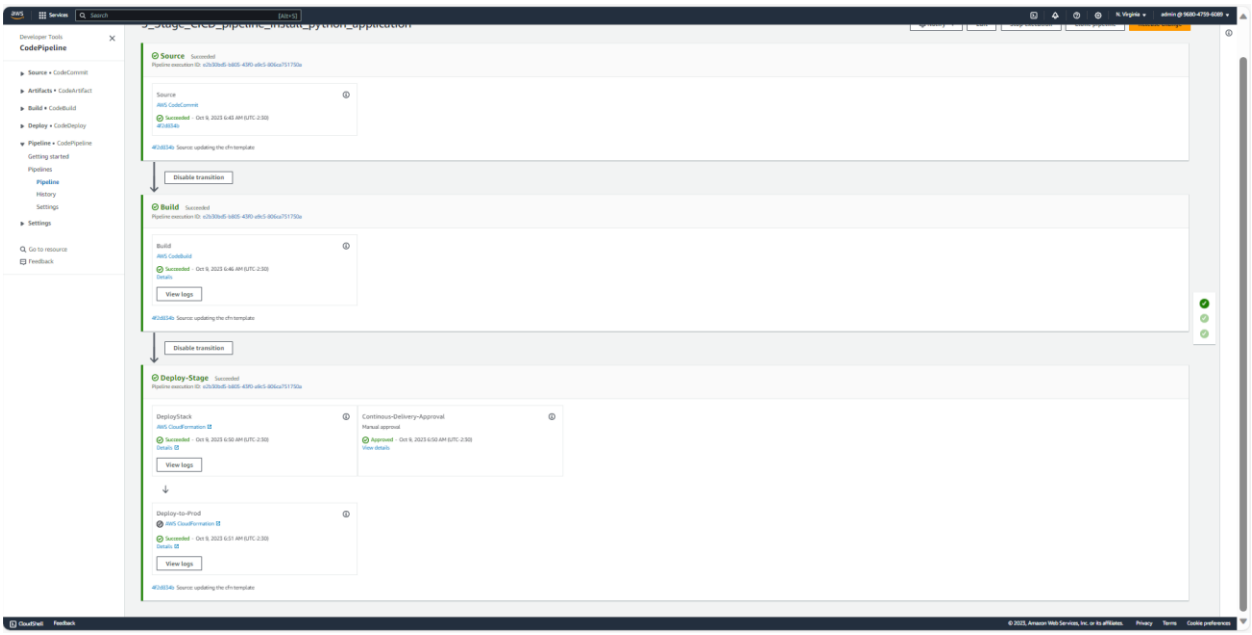
- Since Keyname is regional, you need to select a key in the US-WEST-1 region
- You can use the same Role as you used in the deploy-to-test stage





After everything is set and I approved the delivery, the pipeline moved to the Production stage as seen below;

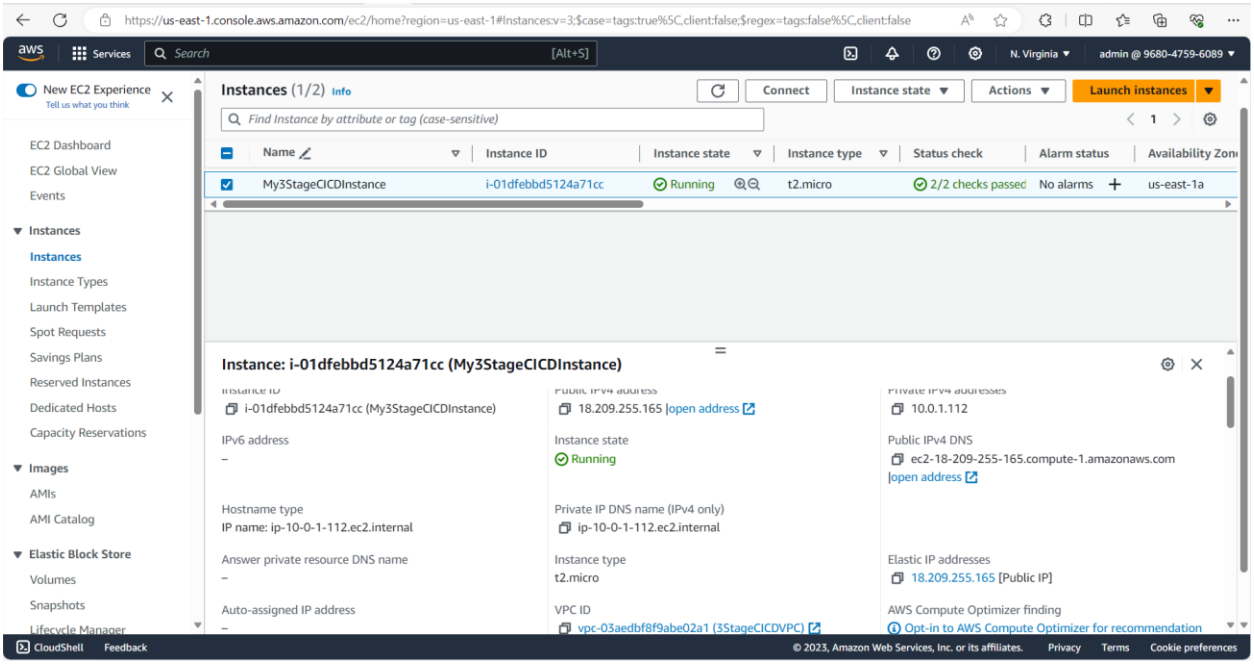




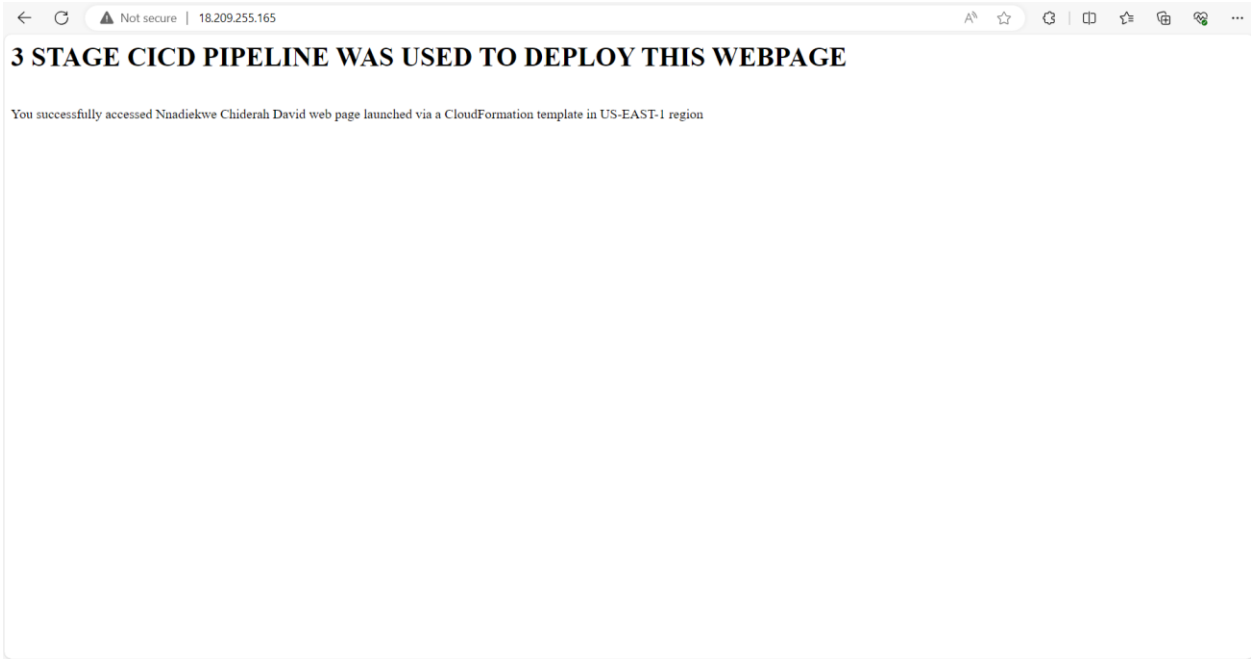
Pipeline was deployed successfully!

Results:

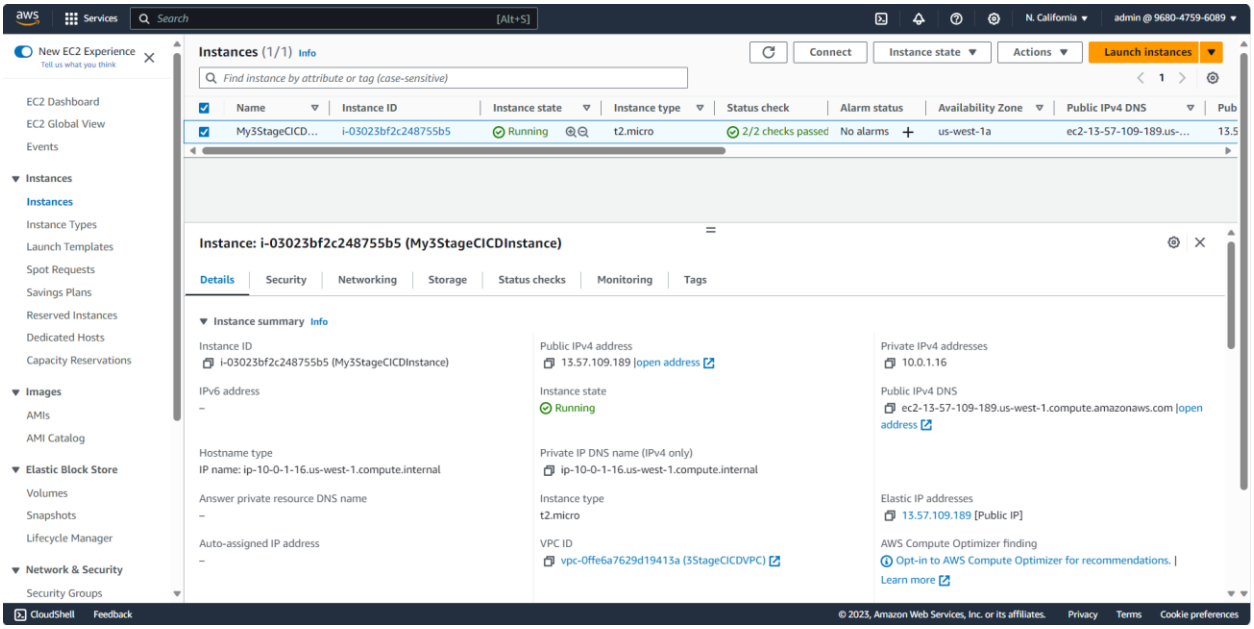
The instance launched in us-east-1 (Testing)



The static website in us-east-1



The instance launched in US-WEST-1 (production)



The static webpage in US-WEST-1

3 STAGE CICD PIPELINE WAS USED TO DEPLOY THIS WEBPAGE

You successfully accessed Nnadiekwe Chiderah David web page launched via a CloudFormation template in US-WEST-1 region

The deployed stack