

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can easily integrate AWS CodePipeline with third-party services such as GitHub or with your own custom plugin. With AWS CodePipeline, you only pay for what you use. There are no upfront fees or long-term commitments

This activity guide cover steps for:

- 1.Create a CodeCommit repository
- 2.Add sample code to your CodeCommit repository
- 3.Create an EC2 Linux instance and install the CodeDeploy agent
- 4.To launch an instance
- 5.Create an application in CodeDeploy
- 6.Create your first pipeline in CodePipeline
- 7.To verify that your pipeline ran successfully
- 8.Modify code in your CodeCommit repository
- 9.To verify your pipeline ran successfully
- 10.Clean up resources

a.Unzip the files from

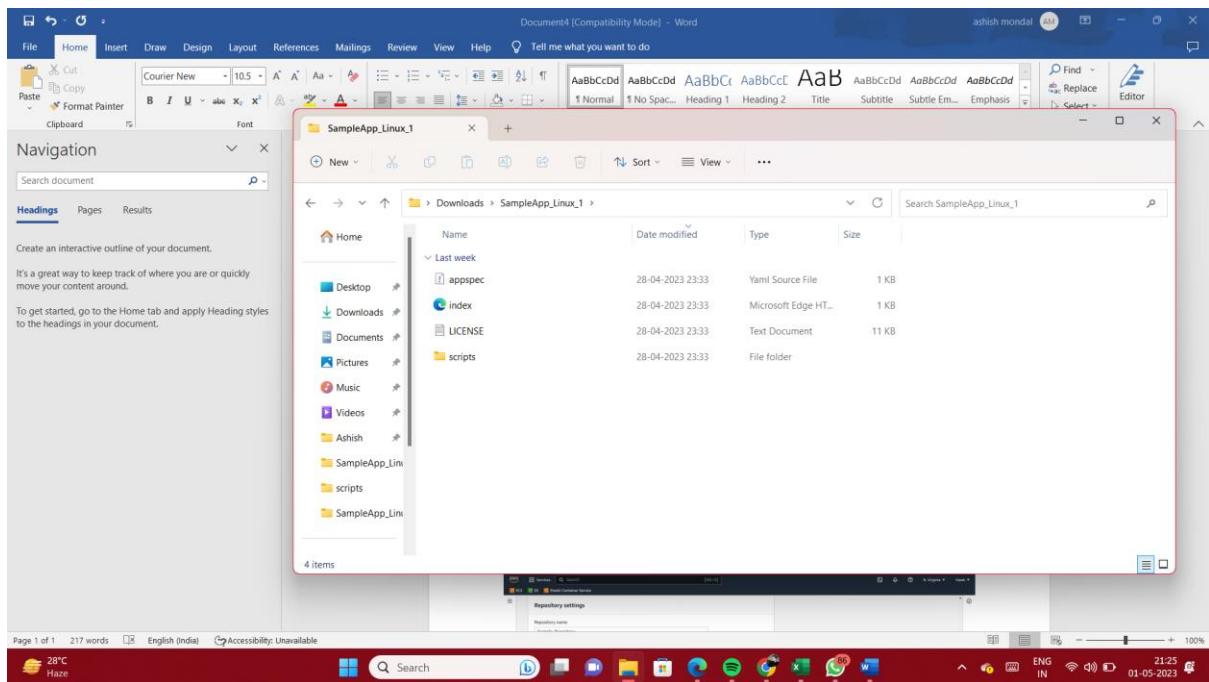
https://docs.aws.amazon.com/codepipeline/latest/userguide/samples/SampleApp_Linux.zip

into the local directory (for example, /tmp/MyDemoRepo or c:\temp\MyDemoRepo).

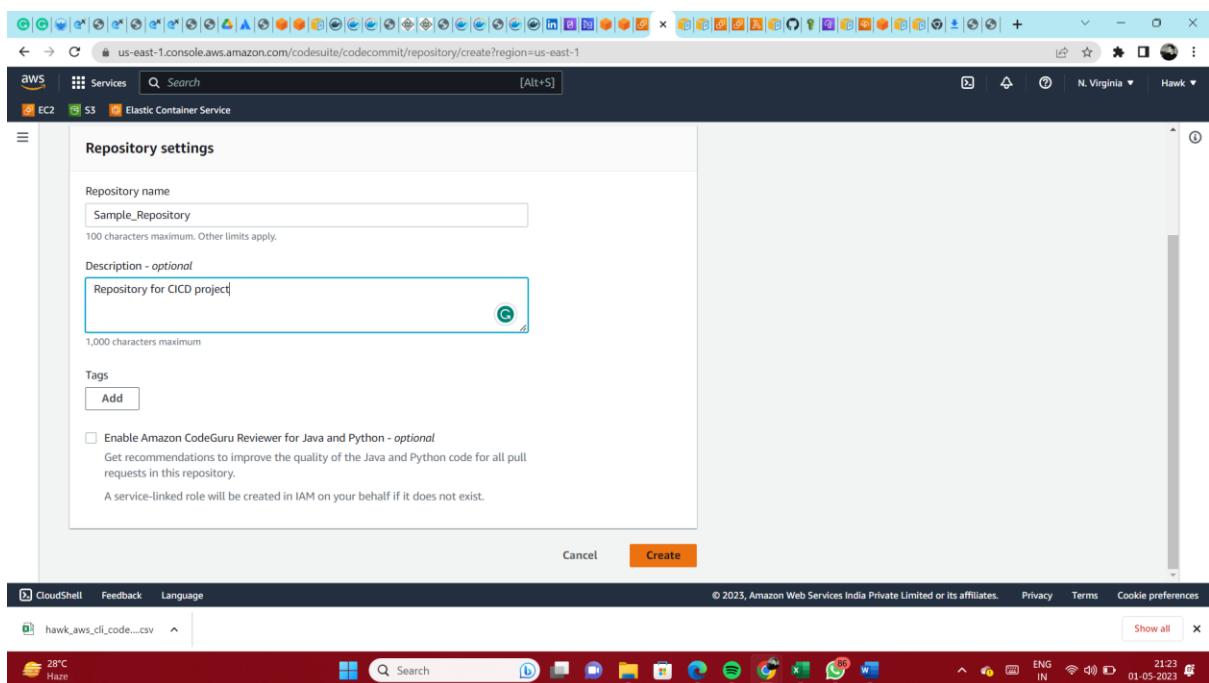
Be sure to place the files directly into your local repository. Do not include a

SampleApp_Linux folder. On your local machine for example, your directory and file hierarchy should look like this:

Note – download the zip file unzip it



Note – create a codecommit repository and then git bash over the unzipped folder and perform the git commands to clone this codecommit repository then create a remote user then push the updated repository back to the codecommit repo.



Note – to clone this repository you need to copy https url and then put that in gitbash as “git clone https://url” and most import while cloning it will ask for git credential which you will get from IAM user got to IAM user create a corresponding IAM user and generate the credentials for https

Success
Repository successfully created

Developer Tools > CodeCommit > Repositories > Sample_Repository

Clone URL

Connection steps

HTTPS (selected), SSH, HTTPS (GRC)

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](#)

Date modified	Type	Size
28-04-2023 23:33	YAML Source File	1 KB
28-04-2023 23:33	Microsoft Edge HT...	1 KB
28-04-2023 23:33	Text Document	11 KB
28-04-2023 23:33	File folder	

b. Use git commands to upload the code in the local directory to public repository in Codecommit

```

|-- appspec.yml
|-- index.html
|-- LICENSE.txt
└-- scripts
    |-- install_dependencies
    |-- start_server
    └-- stop_server

```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1
```

```
$ ls
```

```
LICENSE.txt appspec.yml index.html scripts/
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1
```

```
$ git init
```

```
Initialized empty Git repository in C:/Users/Ashish/Downloads/SampleApp_Linux_1/.git/
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ git add .
```

```
warning: LF will be replaced by CRLF in LICENSE.txt.
```

```
The file will have its original line endings in your working directory
```

```
warning: LF will be replaced by CRLF in appspec.yml.
```

```
The file will have its original line endings in your working directory
```

```
warning: LF will be replaced by CRLF in index.html.
```

```
The file will have its original line endings in your working directory
```

```
warning: LF will be replaced by CRLF in scripts/install_dependencies.
```

```
The file will have its original line endings in your working directory
```

```
warning: LF will be replaced by CRLF in scripts/start_server.
```

```
The file will have its original line endings in your working directory
```

```
warning: LF will be replaced by CRLF in scripts/stop_server.
```

```
The file will have its original line endings in your working directory
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ git commit -m "commit 1"
```

```
[master (root-commit) 9474278] commit 1
```

```
6 files changed, 266 insertions(+)
```

```
create mode 100644 LICENSE.txt
```

```
create mode 100644 appspec.yml
create mode 100644 index.html
create mode 100644 scripts/install_dependencies
create mode 100644 scripts/start_server
create mode 100644 scripts/stop_server
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ cd ../
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads
$ mkdir awscode
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads
$ cd SampleApp_Linux_1
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ ls
LICENSE.txt appspec.yml index.html scripts/
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/awscode
Cloning into 'awscode'...
warning: You appear to have cloned an empty repository.
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ ls
LICENSE.txt appspec.yml awscode/ index.html scripts/
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/awscode
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ git remote -V
```

```
error: unknown switch 'V'
```

```
usage: git remote [-v | --verbose]
```

```
or: git remote add [-t <branch>] [-m <master>] [-f] [--tags | --no-tags] [--mirror=<fetch|push>] <name> <url>
```

```
or: git remote rename <old> <new>
```

```
or: git remote remove <name>
```

```
or: git remote set-head <name> (-a | --auto | -d | --delete | <branch>)
```

```
or: git remote [-v | --verbose] show [-n] <name>
```

```
or: git remote prune [-n | --dry-run] <name>
```

```
or: git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]
```

```
or: git remote set-branches [--add] <name> <branch>...
```

```
or: git remote get-url [--push] [--all] <name>
```

```
or: git remote set-url [--push] <name> <newurl> [<oldurl>]
```

```
or: git remote set-url --add <name> <newurl>
```

```
or: git remote set-url --delete <name> <url>
```

-v, --verbose be verbose; must be placed before a subcommand

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ git remote -v
```

```
origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/awscode (fetch)
```

```
origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/awscode (push)
```

```
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
```

```
$ git push -u origin master
```

Enumerating objects: 9, done.

Counting objects: 100% (9/9), done.

Delta compression using up to 12 threads

Compressing objects: 100% (7/7), done.

Writing objects: 100% (9/9), 5.01 KiB | 2.50 MiB/s, done.

Total 9 (delta 0), reused 0 (delta 0), pack-reused 0

remote: Validating objects: 100%

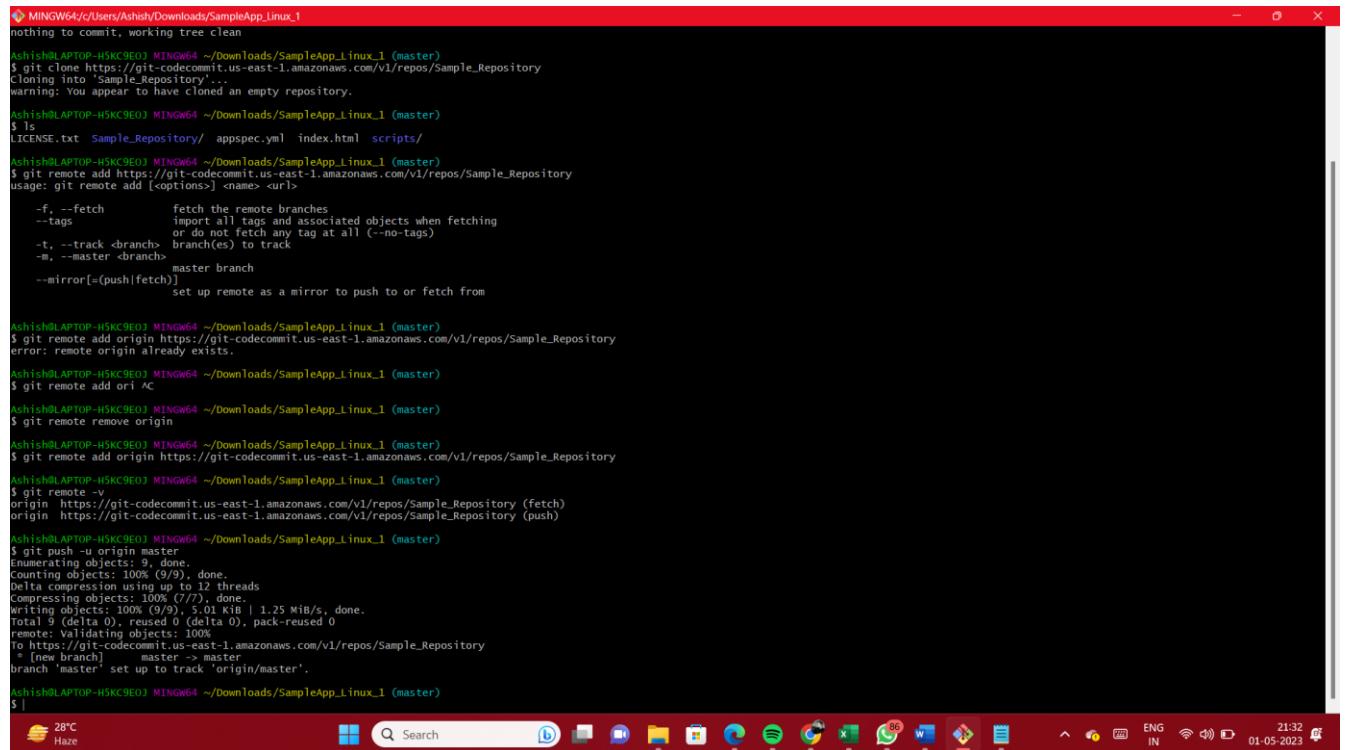
To <https://git-codecommit.us-east-1.amazonaws.com/v1/repos/awscode>

```
* [new branch] master -> master
```

branch 'master' set up to track 'origin/master'.

Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)

\$

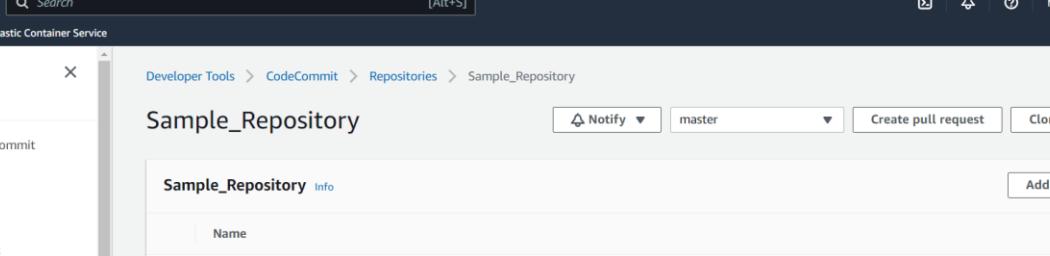


```
MINGW64/c/Users/Ashish/Downloads/SampleApp_Linux_1
nothing to commit, working tree clean
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository
Cloning into 'Sample_Repository'...
warning: You appear to have cloned an empty repository.
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ ls
LICENSE.txt Sample_Repository appspec.yml index.html scripts/
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git remote add https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository
usage: git remote add [<options>] <name> <url>
      -f, --fetch      fetch the remote branches
      -t, --tags       import all tags and associated objects when fetching
      or do not fetch any tag at all (--no-tags)
      -t, --track <branch> branch(es) to track
      -m, --master <branch> master branch
      --mirror=[(push|fetch)] set up remote as a mirror to push to or fetch from

Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository
error: remote origin already exists.
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git remote add ori 'AC'
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git remote remove origin
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository
Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git remote -v
origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository (fetch)
origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository (push)

Ashish@LAPTOP-H5KC9EOJ MINGW64 ~/Downloads/SampleApp_Linux_1 (master)
$ git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 5.01 KiB | 1.25 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Sample_Repository
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
```

Note – you will face error if you don't create a IAM user and give it a AdministratorAccess as a permission policy and generate a credentials for https



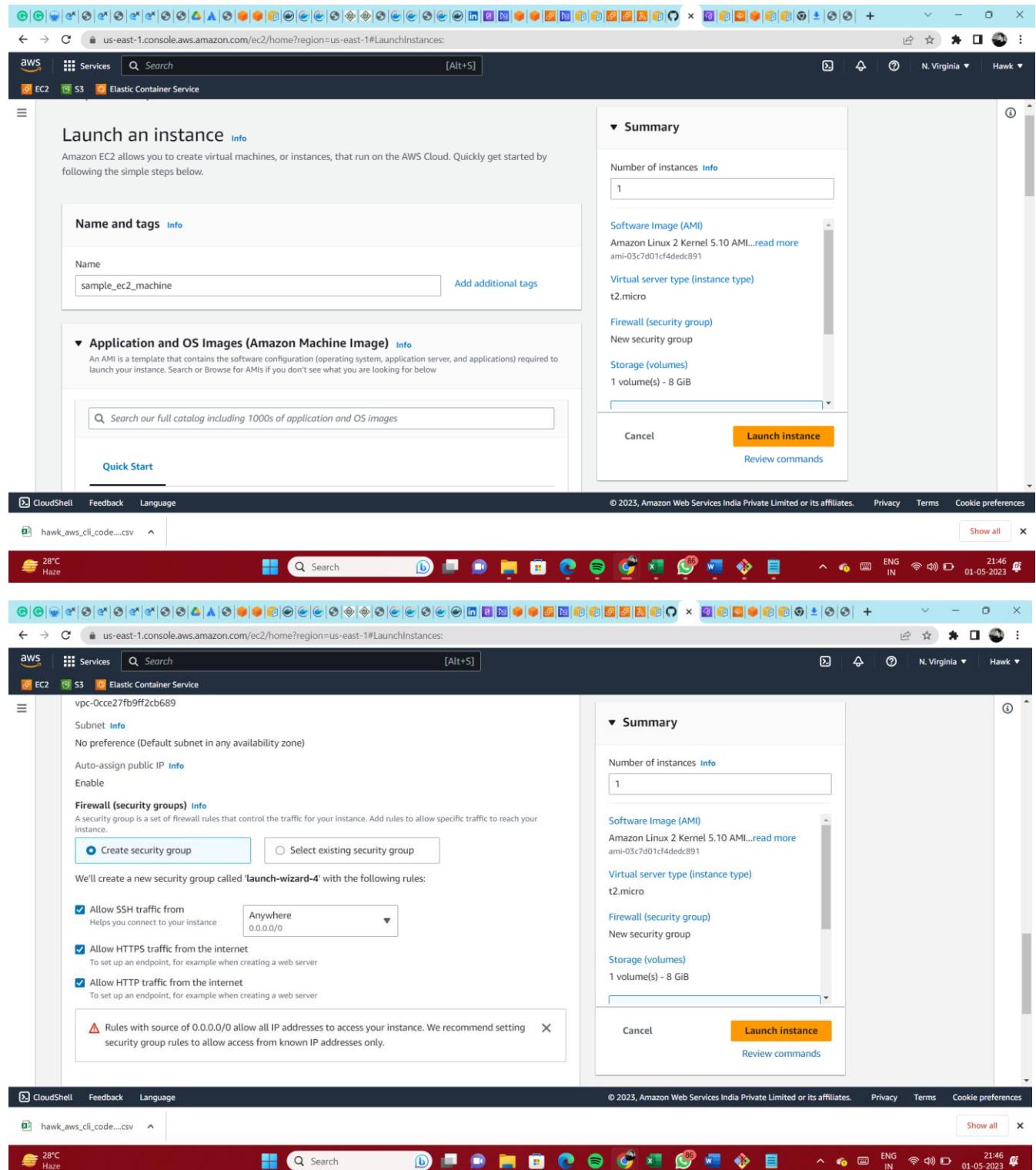
The screenshot shows the AWS CodeCommit interface for the 'Sample_Repository'. The repository contains the following files:

- scripts
- appspec.yml
- index.html
- LICENSE.txt

Note – create a ec2 instance and attach a IAM role AmazonEC2RoleforAwsCodeDeploy

The screenshot shows the AWS IAM Permissions page for the 'ec2-codeddeploy' role. The 'Permissions' tab is selected. The 'Permissions policies' section lists two managed policies: 'AWSCodeCommitFullAccess' and 'AmazonEC2RoleforAWSCodeDeploy'. The 'AWSCodeCommitFullAccess' policy is described as providing full access to AWS CodeCommit. The 'AmazonEC2RoleforAWSCodeDeploy' policy is described as providing EC2 access to S3 buckets. The 'Trust relationships' and 'Tags' tabs are also visible.

Policy name	Type	Description
AWSCodeCommitFullAccess	AWS managed	Provides full access to AWS CodeCommit
AmazonEC2RoleforAWSCodeDeploy	AWS managed	Provides EC2 access to S3 buckets



The screenshot shows the AWS CloudWatch Metrics console. A single metric, 'AWS Lambda Function Invocations', is displayed. The value is 1 for all data points. The chart is a line graph with a 5-minute interval for the last hour.

Metrics Overview

Metrics

- AWS Lambda Function Invocations** (1)

Time Range (Last 1 hour)

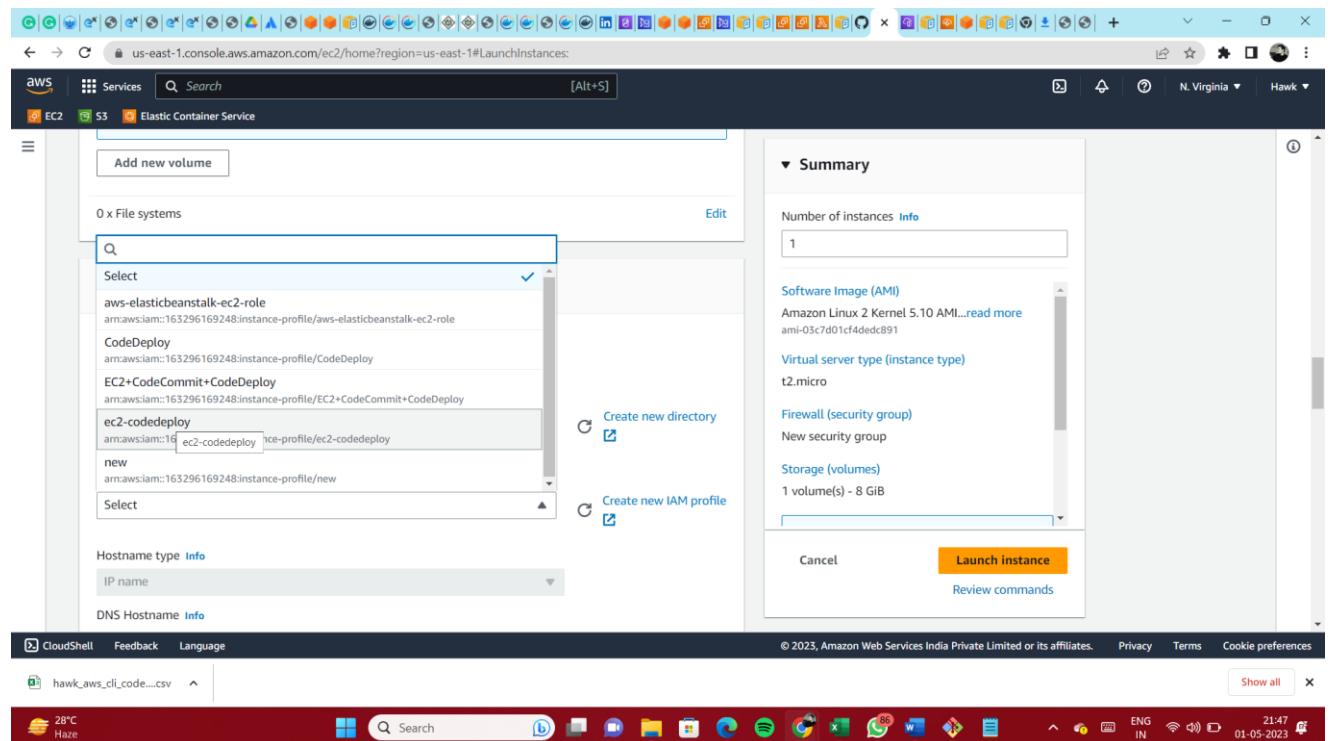
Metrics

Time

2023-05-01 00:00:00 - 2023-05-01 01:00:00

Approximate Data

Time	Value
2023-05-01 00:00:00	1
2023-05-01 00:05:00	1
2023-05-01 00:10:00	1
2023-05-01 00:15:00	1
2023-05-01 00:20:00	1
2023-05-01 00:25:00	1
2023-05-01 00:30:00	1
2023-05-01 00:35:00	1
2023-05-01 00:40:00	1
2023-05-01 00:45:00	1
2023-05-01 00:50:00	1
2023-05-01 00:55:00	1
2023-05-01 01:00:00	1



```
#!/bin/bash
sudo yum -y update
sudo yum -y install ruby
sudo yum -y install wget
cd /home/ec2-user
wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
sudo chmod +x ./install
sudo ./install auto
sudo yum install -y python-pip
sudo pip install awscli
```

Allow tags in metadata [Info](#)

User data - optional [Info](#)
 Enter user data in the field.

```
#!/bin/bash
sudo yum -y update
sudo yum -y install ruby
sudo yum -y install wget
cd /home/ec2-user
wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/latest/install
sudo chmod +x ./install
sudo ./install auto
sudo yum install -y python-pip
sudo pip install awscli
```

User data has already been base64 encoded

Summary

Number of instances [Info](#)

Software Image (AMI)
 Amazon Linux 2 Kernel 5.10 AMI... [read more](#)
 ami-03c7d01cf4dedc891

Virtual server type (instance type)
 t2.micro

Firewall (security group)
 New security group

Storage (volumes)
 1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Review commands](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawk_aws_cli_code...csv

Instances (1/1) [Info](#)

Instance state = running [Clear filters](#)

<input checked="" type="checkbox"/> Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
sample_ec2_machine...	i-0f06c28704a28fff7	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-3-89-10...

Instance: i-0f06c28704a28fff7 (sample_ec2_machine)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

Instance summary [Info](#)

Instance ID i-0f06c28704a28fff7 (sample_ec2_machine)	Public IPv4 address 3.89.101.182 open address	Private IPv4 addresses 172.31.20.142
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-89-101-182.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-20-142.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-20-142.ec2.internal	

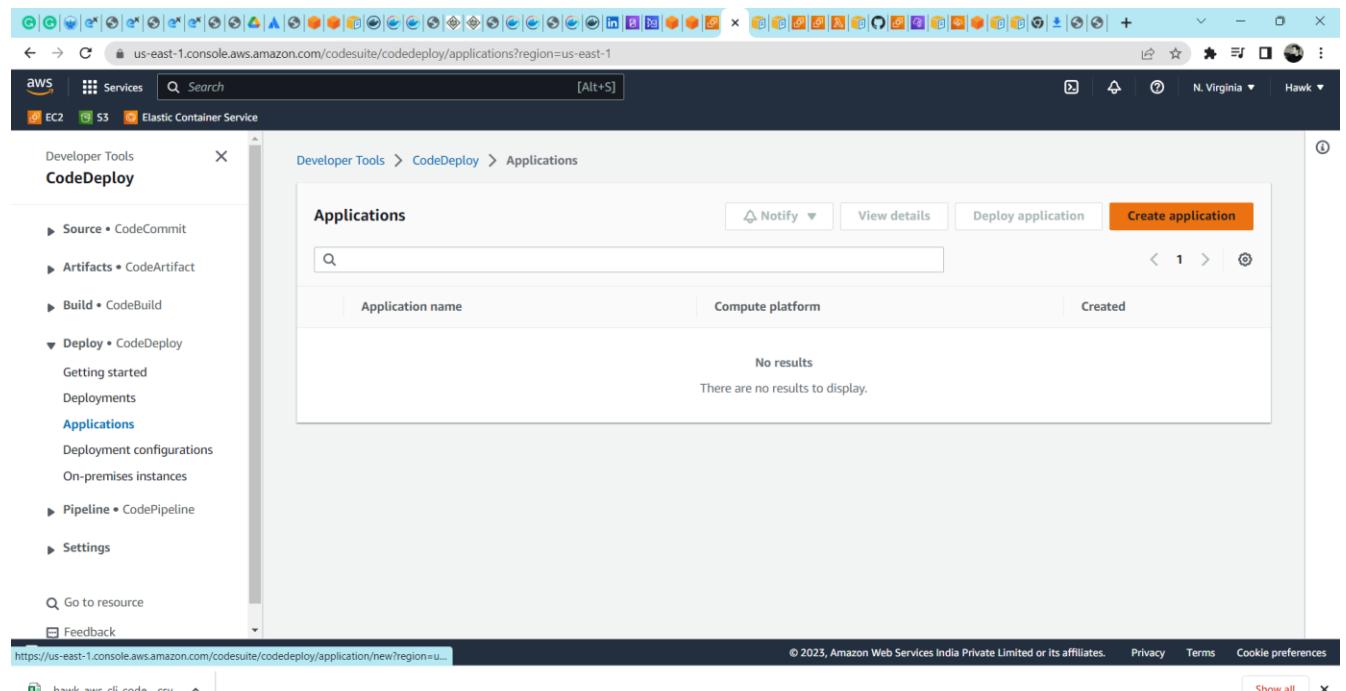
Processing request... © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawk_aws_cli_code...csv

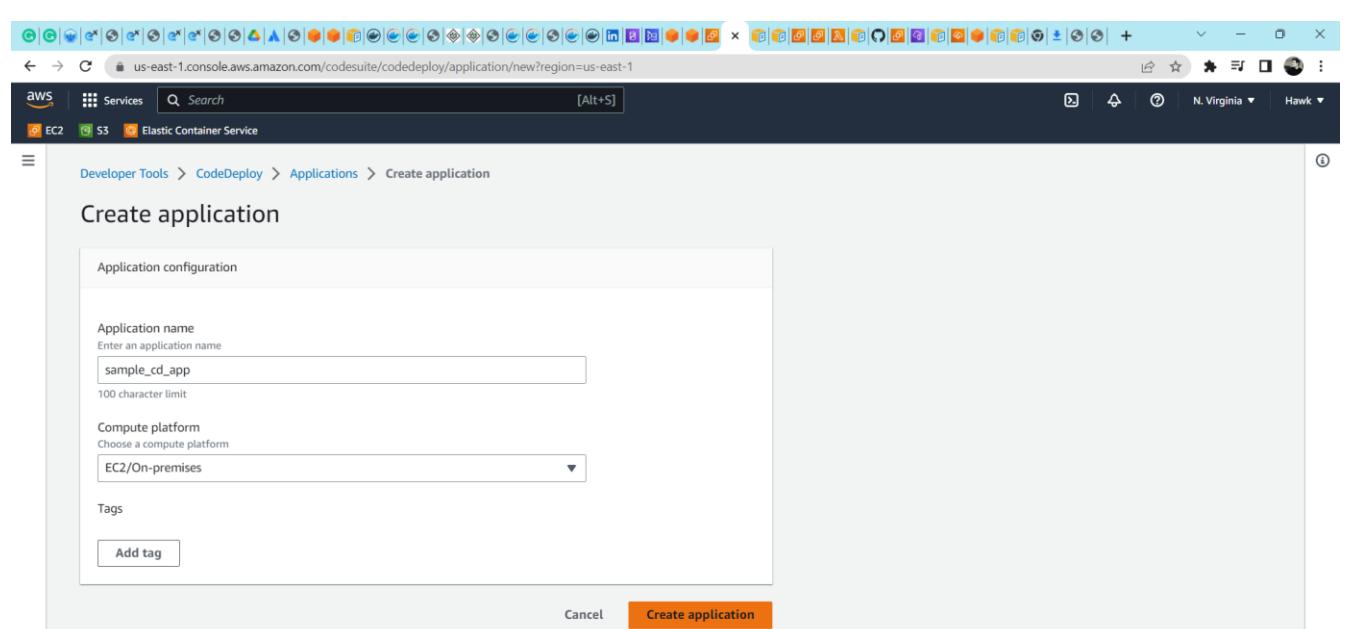
CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawk_aws_cli_code...csv

c.Create and configure Code Deploy (Amazon EC2 instance)



The screenshot shows the AWS CodeDeploy Applications page. The left sidebar is titled 'CodeDeploy' and includes sections for Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), and Settings. Under Deploy, 'Applications' is expanded, showing 'Deployment configurations' and 'On-premises instances'. The main content area is titled 'Applications' and shows a search bar, a 'Create application' button, and a table with columns for 'Application name', 'Compute platform', and 'Created'. A message 'No results' and 'There are no results to display.' is displayed.



The screenshot shows the 'Create application' wizard. The first step, 'Application configuration', is displayed. It includes fields for 'Application name' (with 'sample_cd_app' entered), 'Compute platform' (set to 'EC2/On-premises'), and 'Tags' (with a 'Add tag' button). At the bottom are 'Cancel' and 'Create application' buttons.

Application created

In order to create a new deployment, you must first create a deployment group.

Name: sample_cd_app Compute platform: EC2/On-premises

Deployment groups

No deployment groups

Before you can deploy your application using CodeDeploy, you must create a deployment group.

Create deployment group

Deployment group name: sample_cd_dg

Service role: arn:aws:iam::163296169248:role/codedeploy-new

to this deployment

Amazon EC2 Auto Scaling groups

Amazon EC2 instances
1 unique matched instance. [Click here for details](#)

You can add up to three groups of tags for EC2 instances to this deployment group.

One tag group: Any instance identified by the tag group will be deployed to.

Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key Value - optional

On-premises instances

Matching instances
1 unique matched instance. [Click here for details](#)

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawh_aws_cli_code...csv

28°C Haze

Search

14 Days

Deployment settings

Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

CodeDeployDefault.AllAtOnce

Load balancer

Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

Enable load balancing

Advanced - optional

Cancel

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawh_aws_cli_code...csv

28°C Haze

Search

d. Create a pipeline using AWS CodePipeline , AWS Code Commit and Code Deploy to deploy index.html to Amazon Linux ec2 instance

Screenshot of the AWS CodePipeline console showing the Pipelines list. The sidebar on the left is titled "CodePipeline" and includes sections for Source, Artifacts, Build, Deploy, Pipeline, Pipelines, and Settings. The main content area shows a table with columns: Name, Most recent execution, Latest source revisions, and Last executed. A message at the bottom states "No results" and "There are no results to display."

Below this screenshot is a Windows taskbar with various application icons. The taskbar shows the AWS CloudShell icon is active, and the date and time are 01-05-2023 21:58.

The next section shows the "Pipeline settings" step of creating a new pipeline. The pipeline name is set to "sample_pipeline". The "Service role" section shows "New service role" selected, with a note to "Create a service role in your account". The "Role name" field contains "AWSCodePipelineServiceRole-us-east-1-sample_pipeline". A checkbox for "Allow AWS CodePipeline to create a service role so it can be used with this new pipeline" is checked. A "Next" button is visible at the bottom right.

Below this is another screenshot of the AWS CloudShell interface, showing the command "hawkeye aws cli code...csv" entered in the terminal. The taskbar at the bottom shows the date and time as 01-05-2023 21:59.

us-east-1.console.aws.amazon.com/codesuite/codepipeline/pipeline/new?region=us-east-1

Source

Add source stage

Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Source provider: AWS CodeCommit

Repository name: Sample_Repository

Branch name: master

Change detection options:

- 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:

- 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.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawh_aws_cli_code...csv

28°C Haze

us-east-1.console.aws.amazon.com/codesuite/codepipeline/pipeline/new?region=us-east-1

Add build stage

Step 1: Choose pipeline settings

Step 2: Add source stage

Step 3: Add build stage

Step 4: Add deploy stage

Step 5: Review

Build - optional

Build provider: Skip build stage

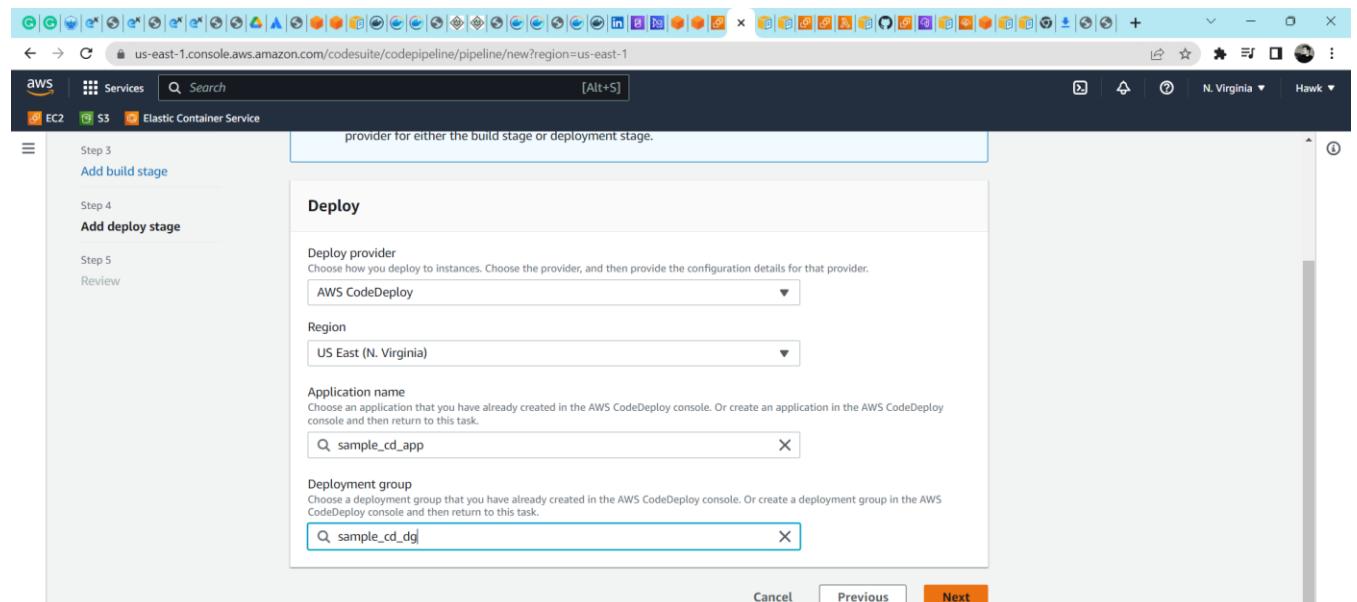
Your pipeline will not include a build stage. Are you sure you want to skip this stage?

Cancel Skip Next

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

hawh_aws_cli_code...csv

28°C Haze



Deploy

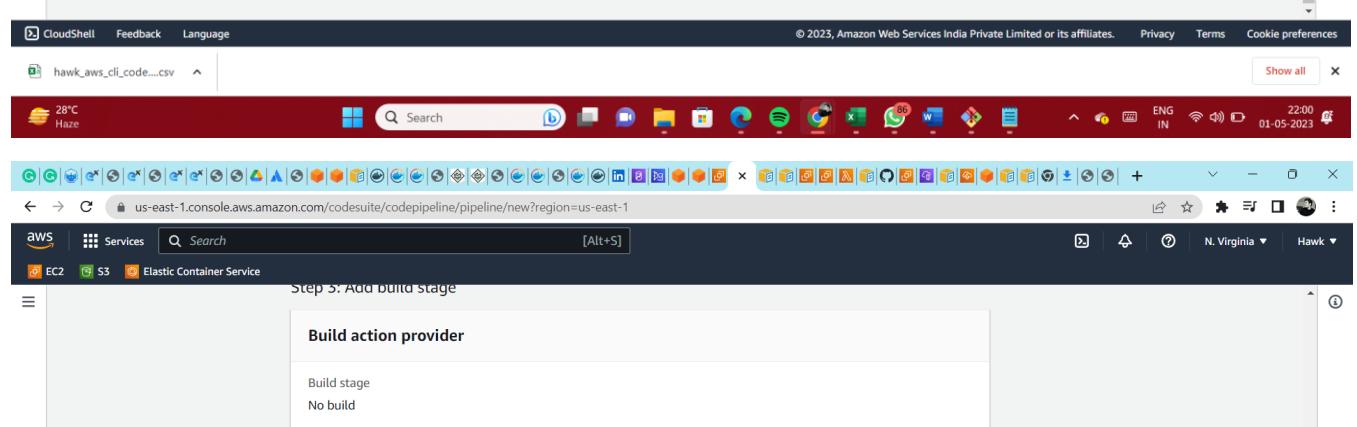
Deploy provider: AWS CodeDeploy

Region: US East (N. Virginia)

Application name: sample_cd_app

Deployment group: sample_cd_dg

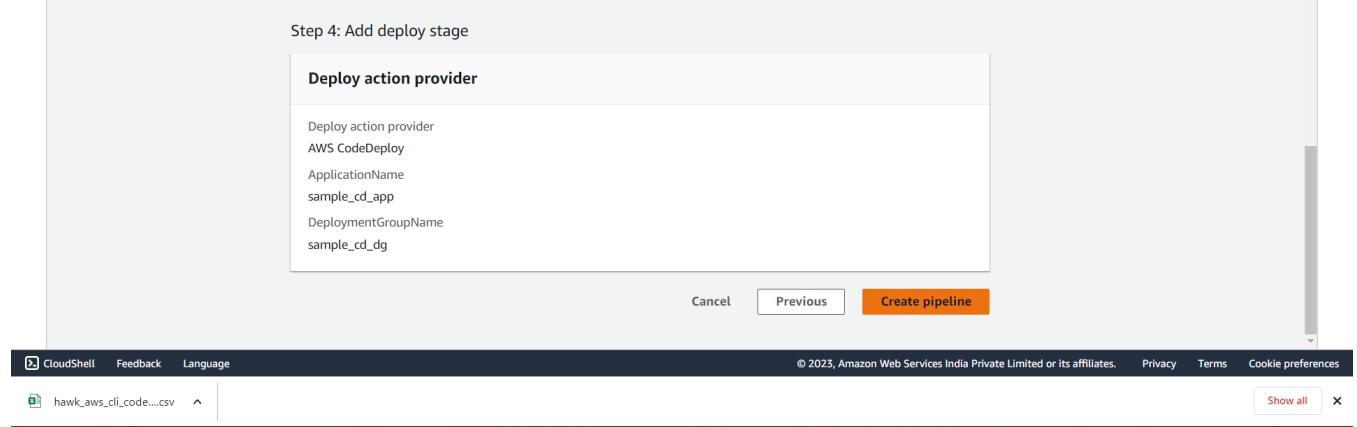
Cancel Previous Next



Step 3: Add build stage

Build action provider: No build

Cancel Previous Create pipeline



Step 4: Add deploy stage

Deploy action provider: AWS CodeDeploy

ApplicationName: sample_cd_app

DeploymentGroupName: sample_cd_dg

Cancel Previous Create pipeline

Note – once you deployment is done successfully you can access the boiler page directly by pasting the public IP of the EC2 instance which you have tagged in the deployment group

The screenshot shows the AWS CodePipeline console with a successful pipeline run. The pipeline consists of two steps: Source (AWS CodeCommit) and Deploy (AWS CodeDeploy). Both steps are marked as 'Succeeded'. A green banner at the top says 'Success' and 'Congratulations! The pipeline sample_pipeline has been created.'

e. Check the o/p , by accessing the public ip of the ec2 instance . Contents of the index.html must be displayed

The screenshot shows a web browser displaying the deployed application. The page has a large 'Congratulations' message and a sub-message stating 'This application was deployed using AWS CodeDeploy.' Below the message is a link 'For next steps, read the [AWS CodeDeploy Documentation](#)'.