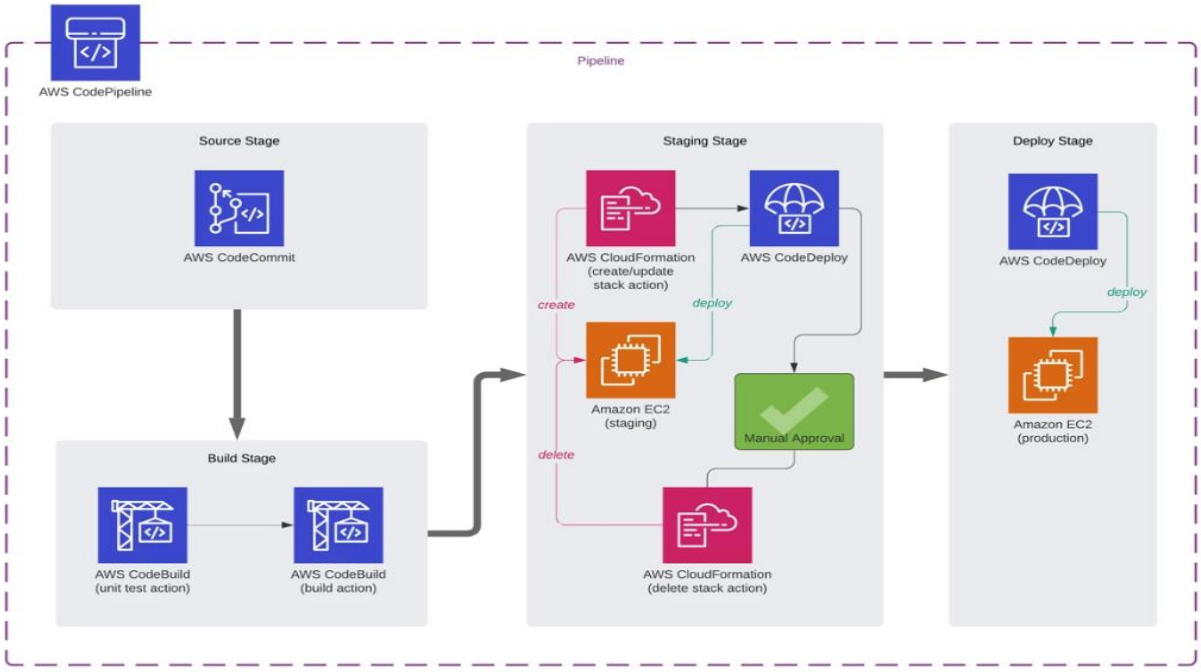
Code Deploy/Code Pipeline (CD/CP) on AWS



NAME : T MANOJ

EMPID : LYAKE2KHS

Introduction

Continuous Delivery and Continuous Deployment (CD/CP) are crucial components of modern DevOps practices. They ensure that applications are delivered to users efficiently, reliably, and with minimal manual intervention. Leveraging AWS services such as EC2, S3, CodeDeploy, CodePipeline, CloudWatch, and SNS enables seamless automation, robust monitoring, and streamlined deployment processes.

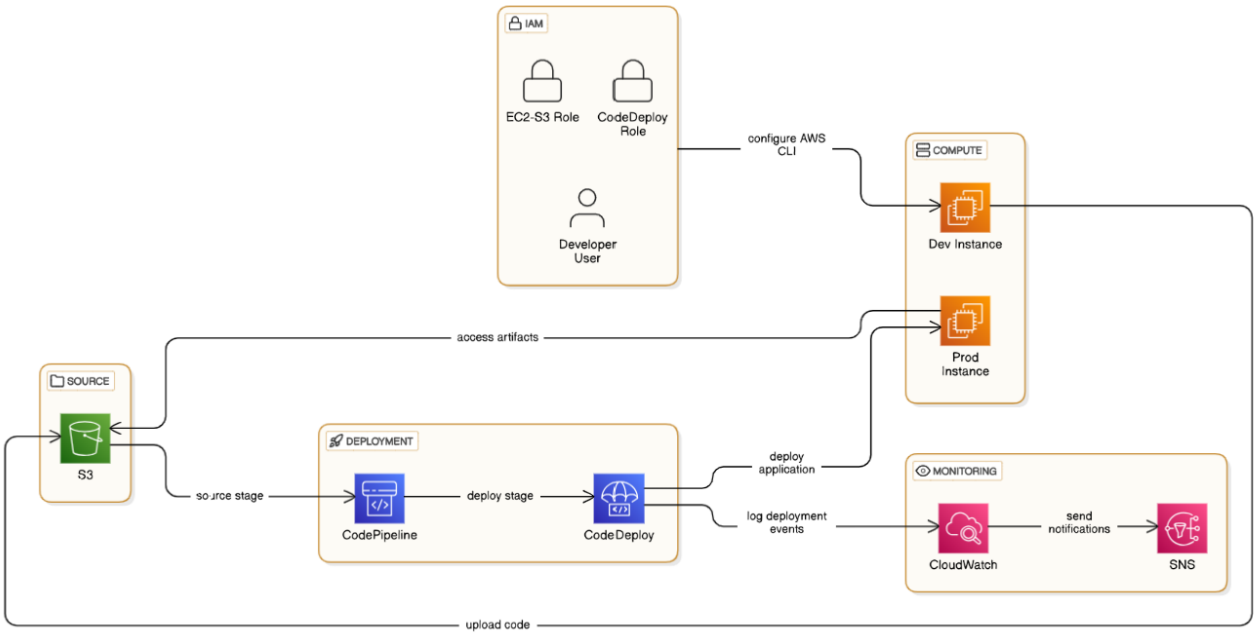
Overview

This document outlines a step-by-step approach to implementing a CD/CP pipeline on AWS. The process involves creating development and production environments, automating deployments using CodePipeline and CodeDeploy, monitoring deployments with CloudWatch, and sending real-time notifications via SNS. The pipeline ensures rapid, consistent, and error-free deployments, reducing downtime and accelerating delivery cycles.

Prerequisites

- **AWS Account:** Active AWS account with appropriate access.
- **IAM Roles and Users:** IAM roles for EC2 (Ec2-s3) and CodeDeploy (codedeploy), along with an IAM user for developers with CLI access.
- **EC2 Instances:** Developer and production instances configured appropriately.
- **AWS CLI:** Configured on the developer instance.
- **S3 Bucket:** Public bucket for storing application artifacts with versioning enabled.

Goal: To automate the deployment of an application to an EC2 instance using AWS services like CodeDeploy and CodePipeline.



AWS Services Used:

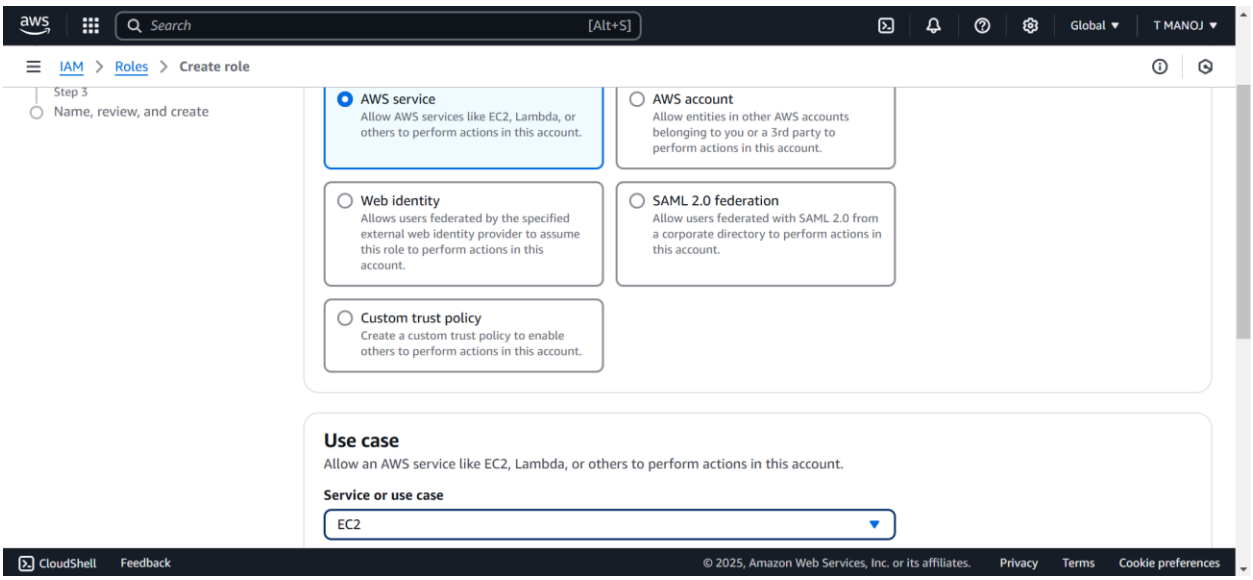
- EC2 (Elastic Compute Cloud): Provides virtual servers for development and production.
- S3 (Simple Storage Service): Stores application code and deployment artifacts.
- CodeDeploy: Automates application deployments to EC2 instances.
- CodePipeline: Orchestrates the entire CD/CP pipeline.
- SNS (Simple Notification Service): Sends notifications for deployment events.
- CloudWatch: Monitors AWS resources and provides logging.
- IAM (Identity and Access Management): Manages permissions.

Detailed WorkFlow:

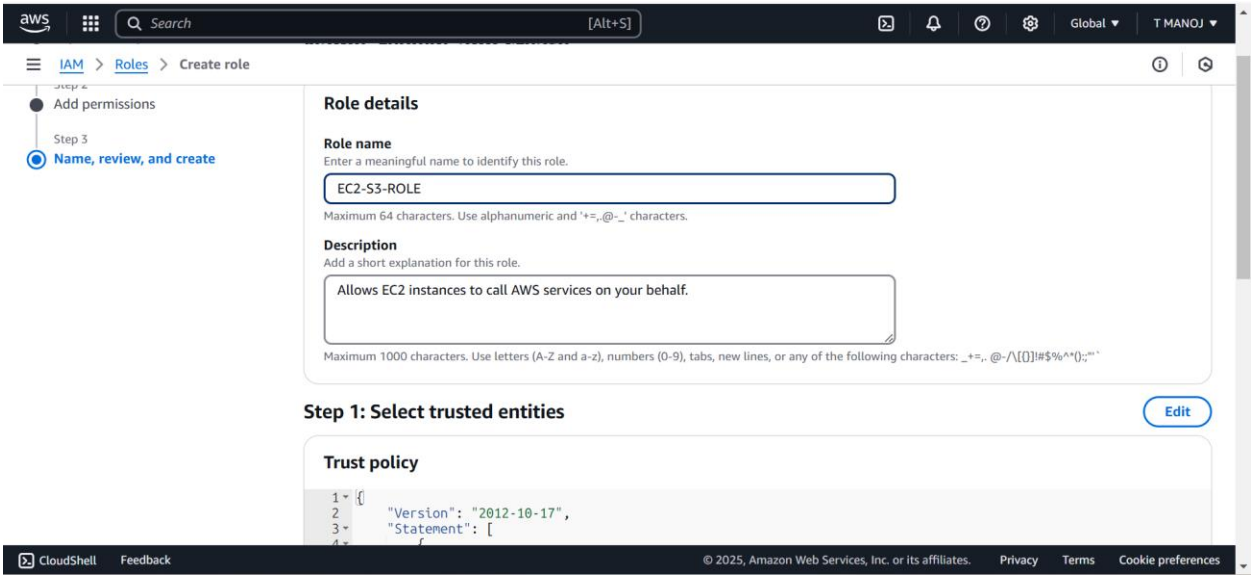
1. Create IAM Roles:

- Role: Ec2-s3
    - Purpose: Grants EC2 instances permission to access S3.
    - Action:
        - Navigate to the IAM console.

- Create a new role.



  - Select "EC2" as the trusted entity.
  - Attach the "AmazonS3FullAccess" policy.



  - Name the role "Ec2-s3."

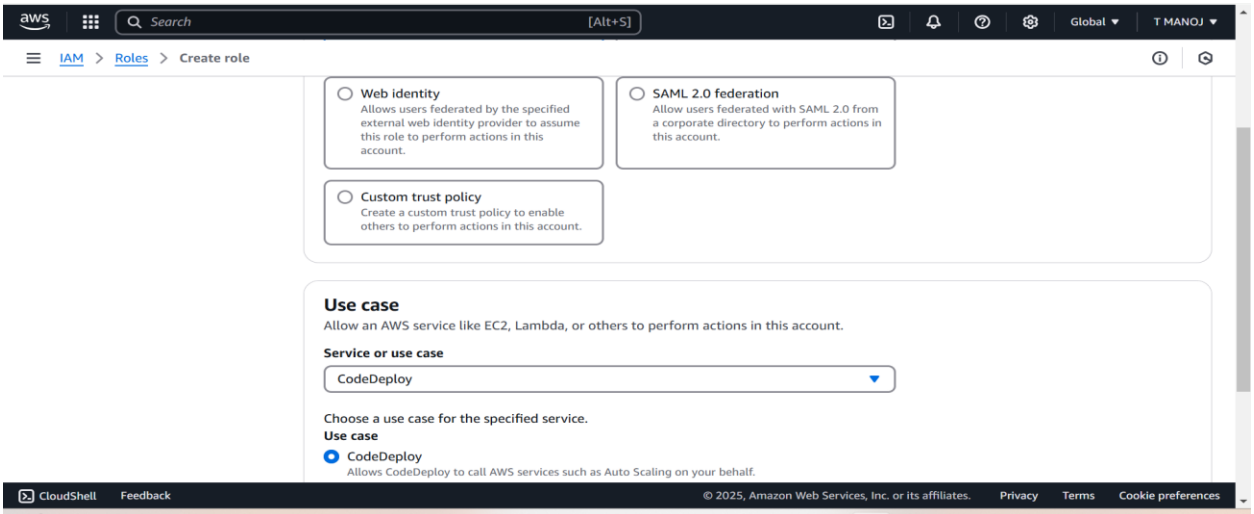
Step 2: Add permissions Edit

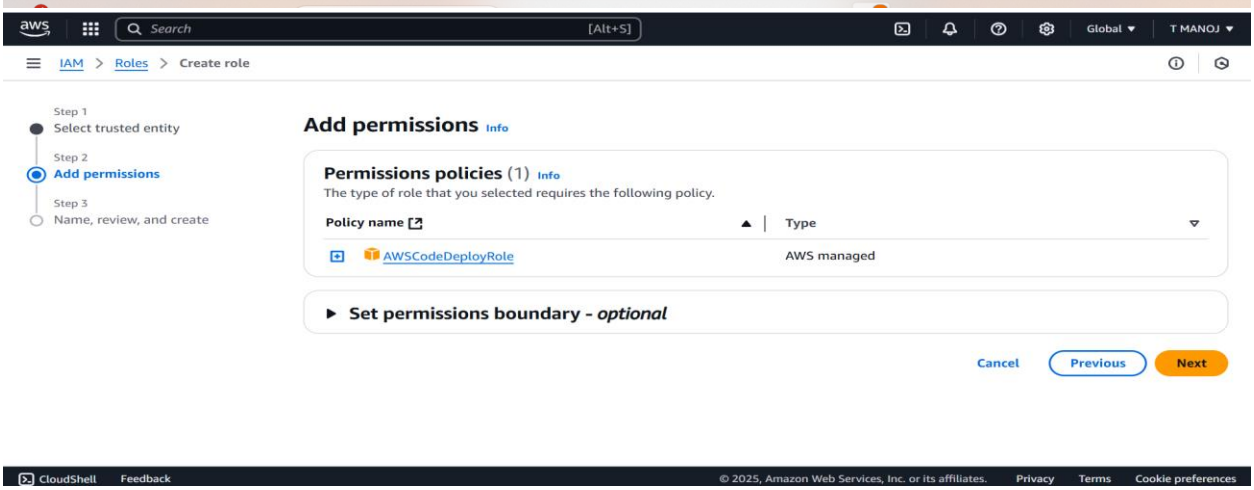
Permissions policy summary

Policy name	Type	Attached as
AmazonS3FullAccess	AWS managed	Permissions policy

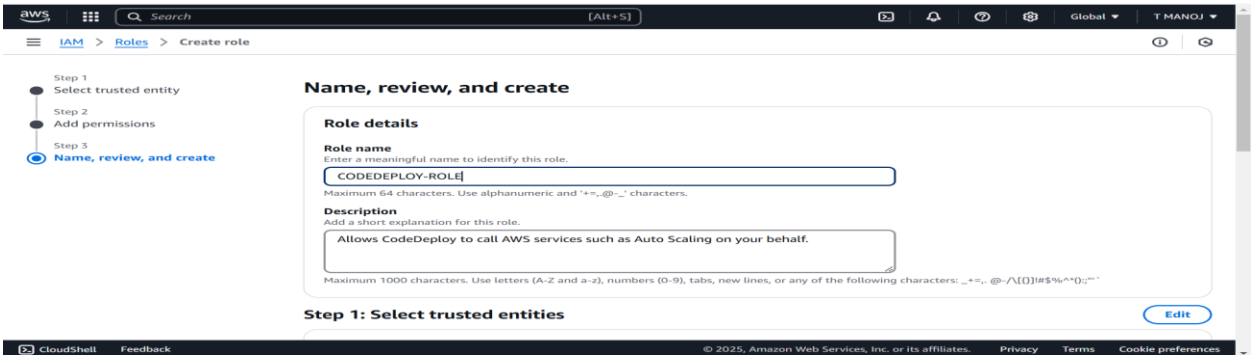
- Role: codedeploy
  - Purpose: Grants CodeDeploy permission to deploy applications.
  - Action:
    - Create a new role.

    - Attach the "AWSCodeDeployFullAccess" policy.





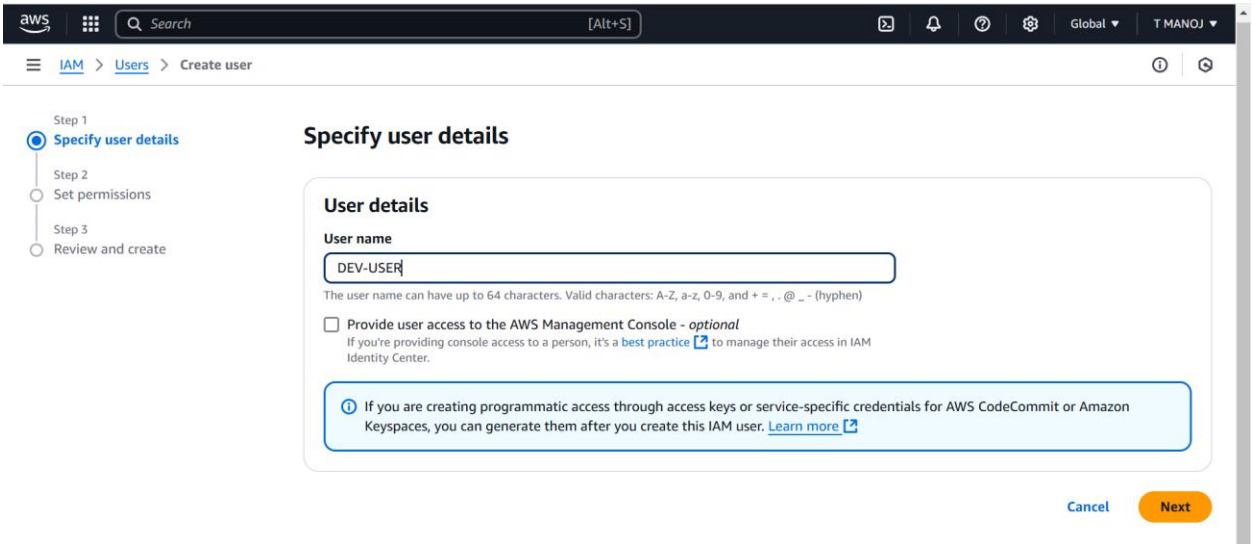
    - Name the role "codedeploy."



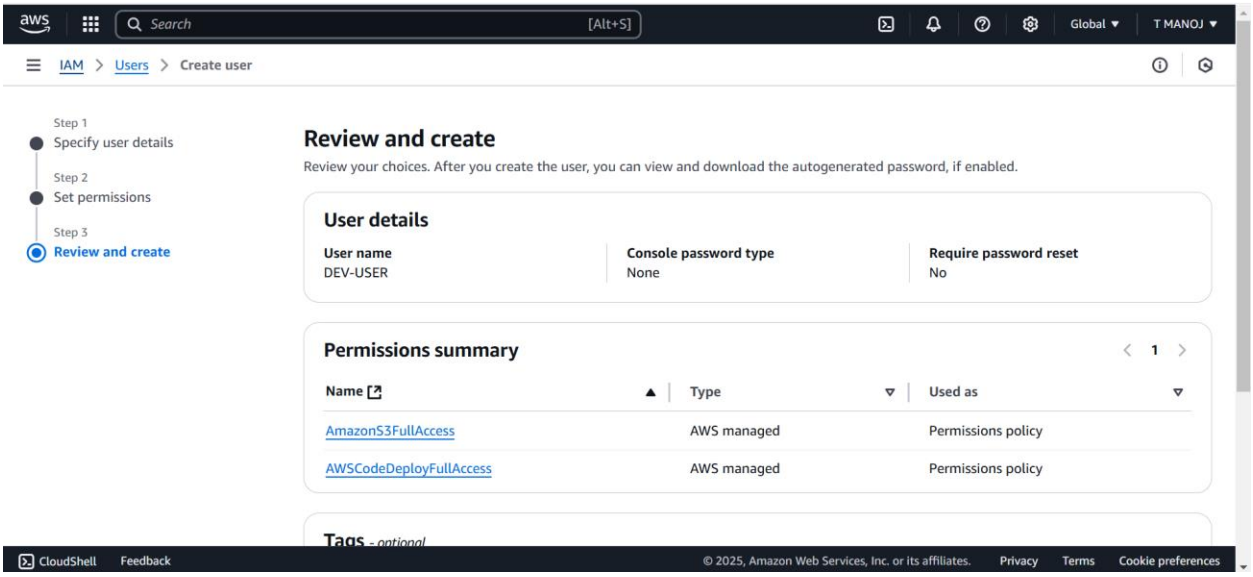
2. Create IAM User (Developer):

- Purpose: Creates an IAM user for developers to interact with AWS via the CLI.
- Action:
    - Navigate to the IAM console.

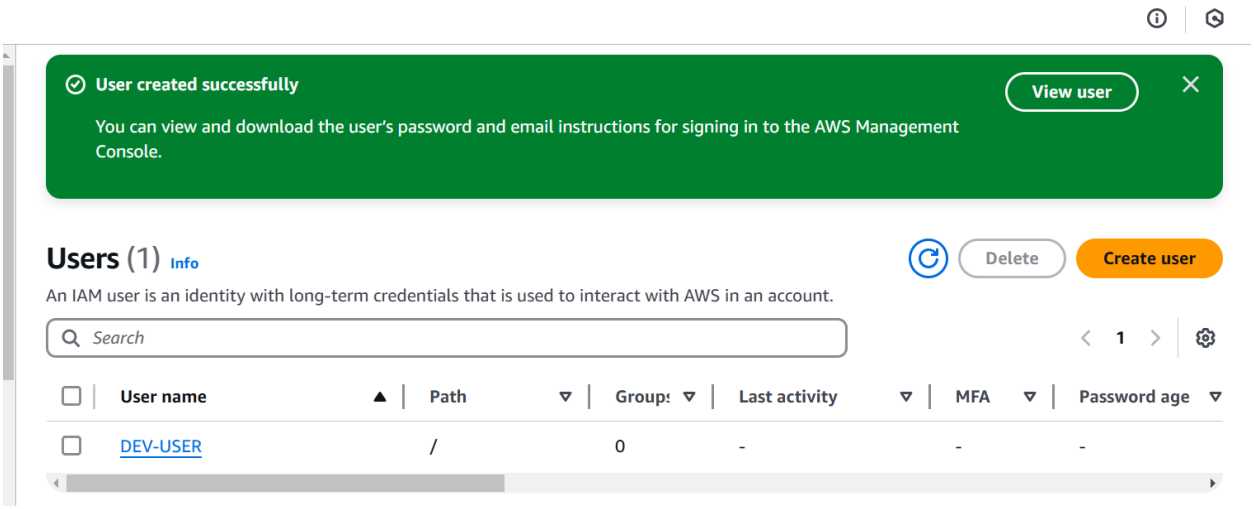
    - Create a new user.
    - Provide a username.



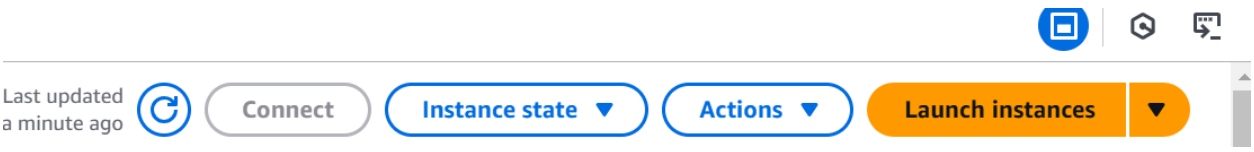
    - Uncheck "AWS Management Console access."
    - Attach "AmazonS3FullAccess" and "AWSCodeDeployFullAccess" policies.



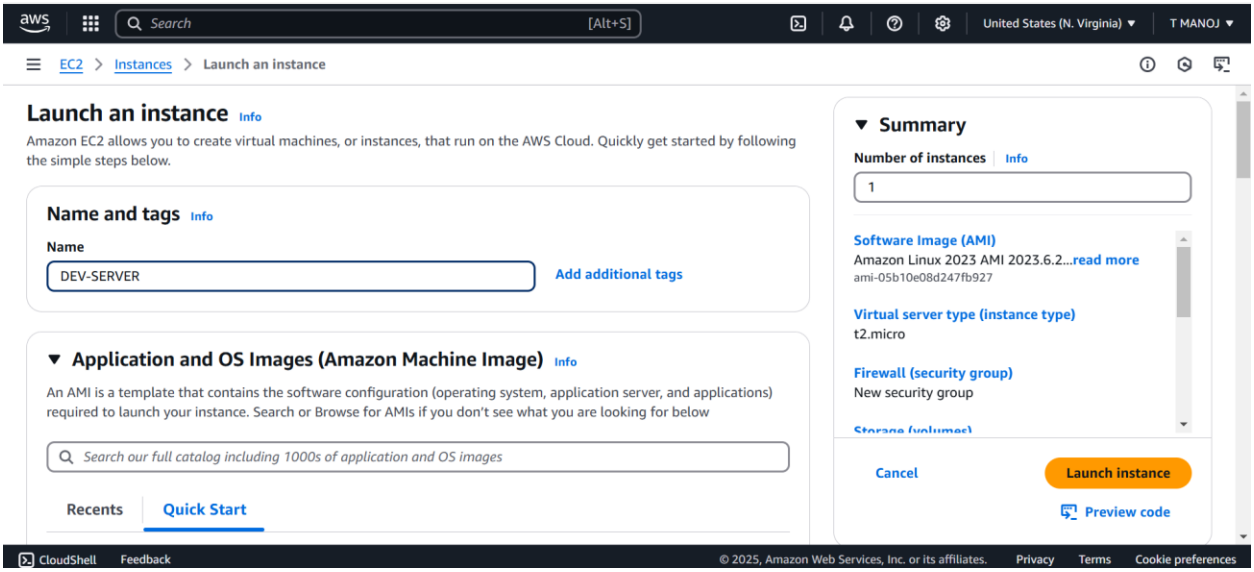
- Create an access key and download the credentials.
- Choose "Amazon CLI" as the key type.



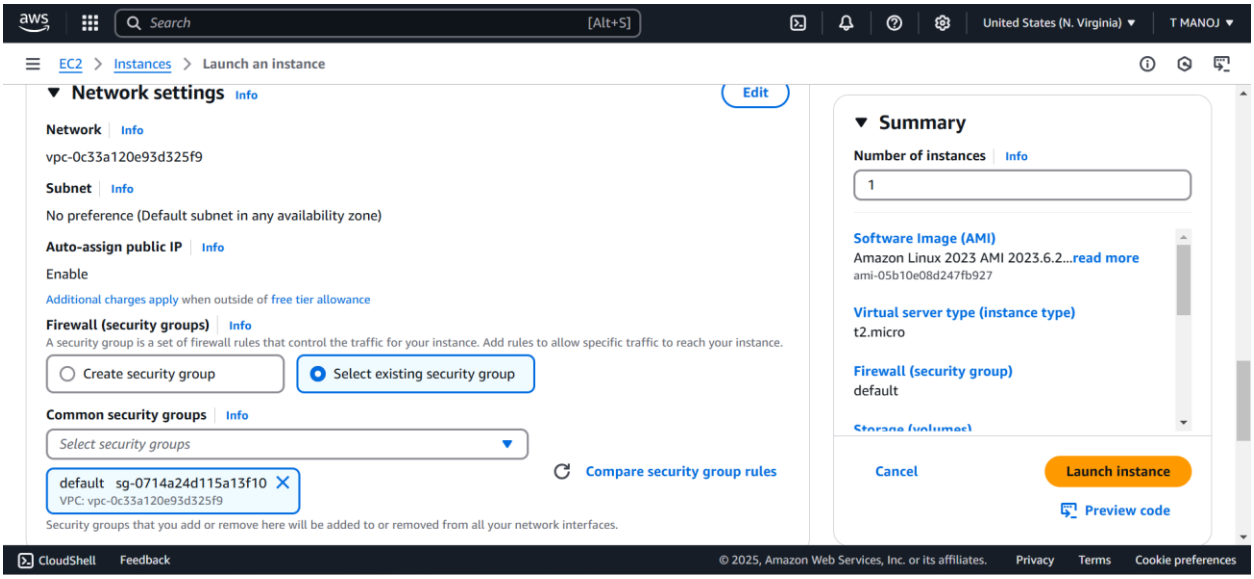
3. Create EC2 Instances:



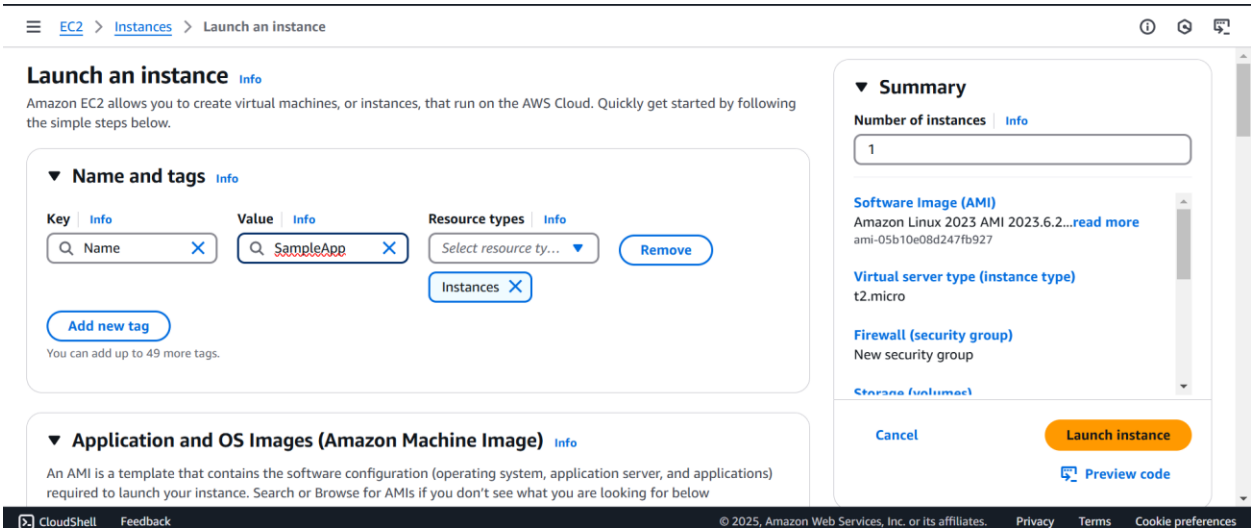
- Developer Instance:
  - Purpose: Development environment.
  - Action:
    - Launch an EC2 instance.



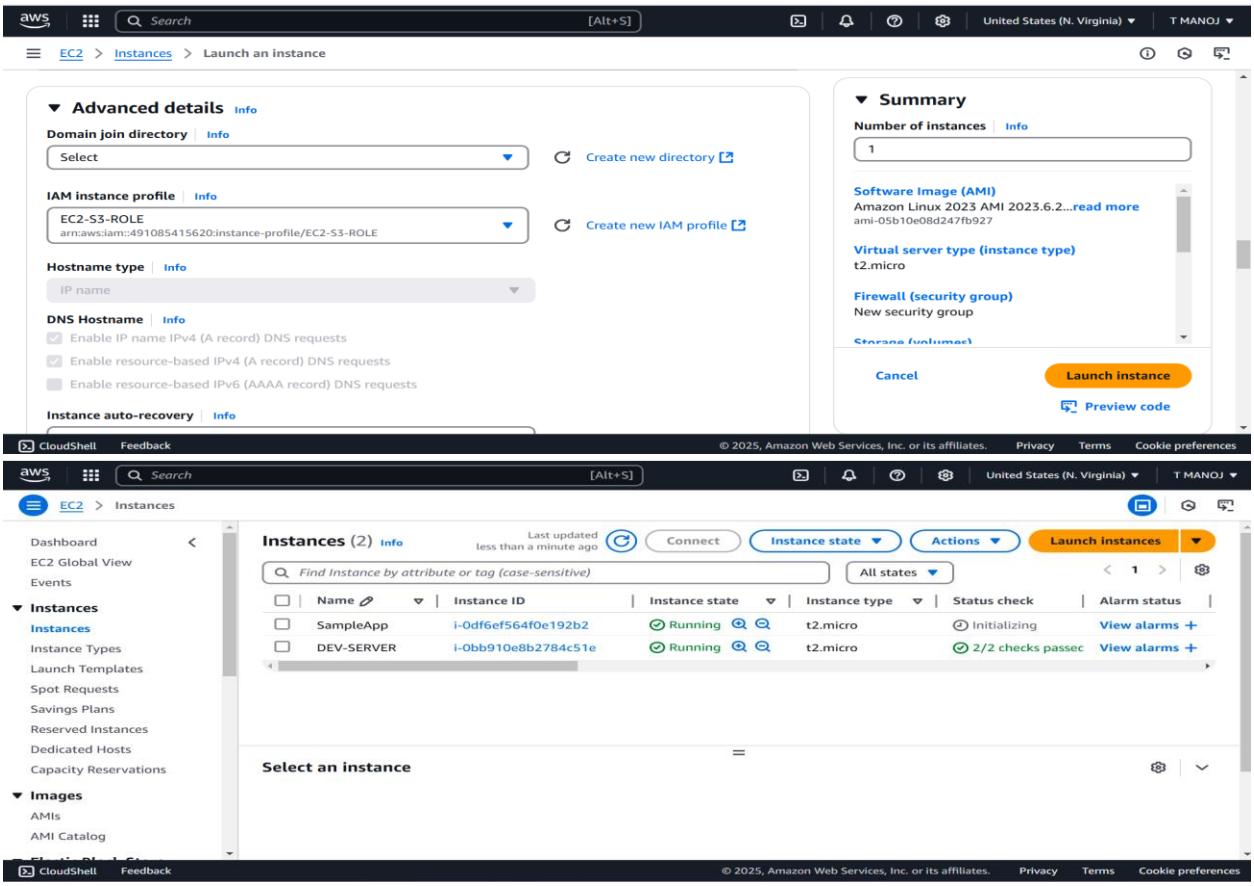
- Configure security group and storage (30GB).



- Production Instance:
  - Purpose: Hosts the deployed application.
  - Action:
    - Launch an EC2 instance.



- Attach the "Ec2-s3" IAM role in "Advanced details."

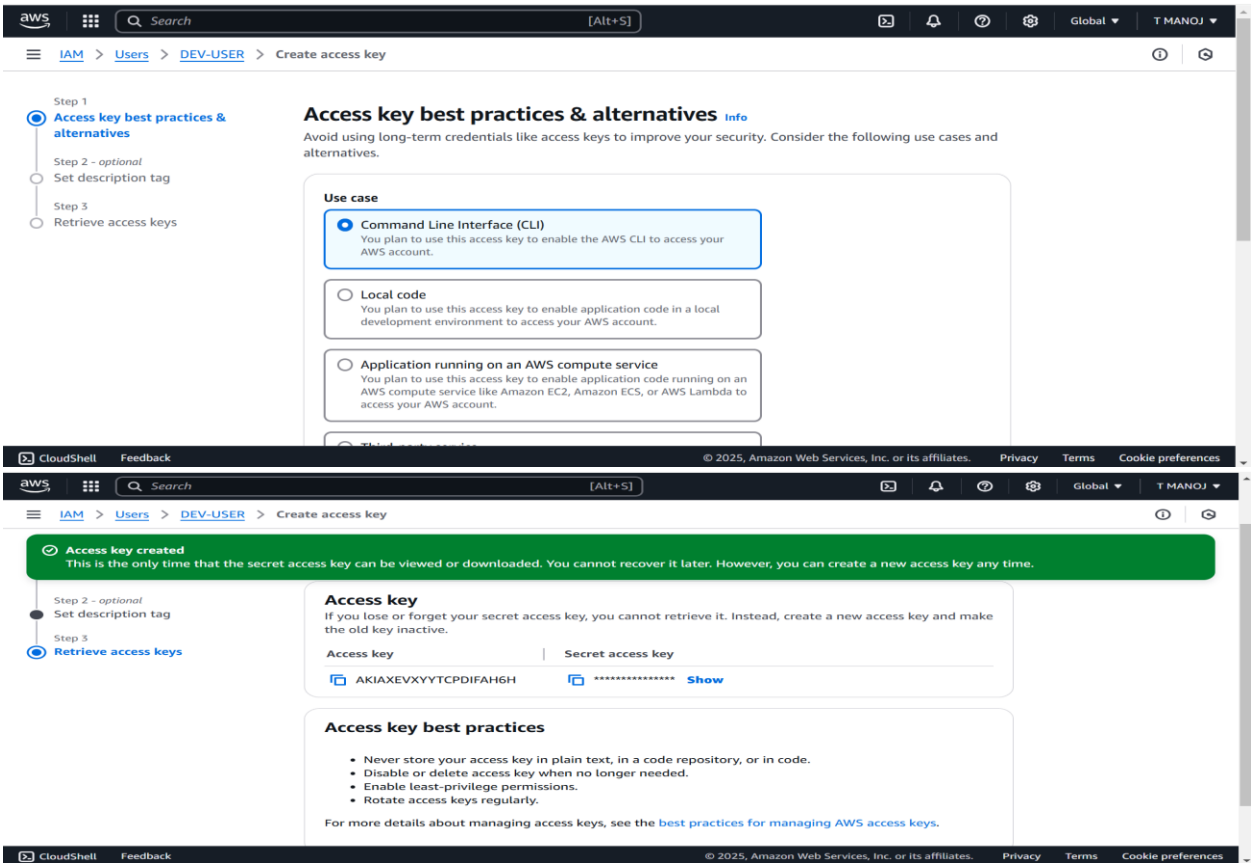


4. Configure Developer Machine:

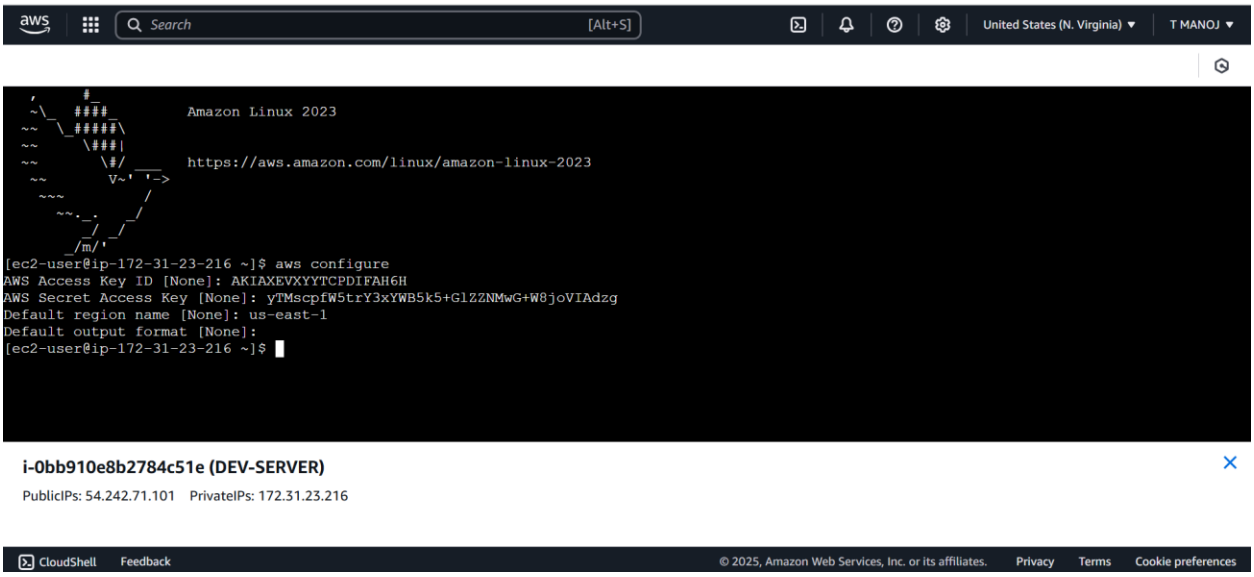
- Purpose: Configure AWS CLI on the developer instance.
- Action:
  - Connect to the developer instance via SSH (PuTTY).

Access key 1

Create access key



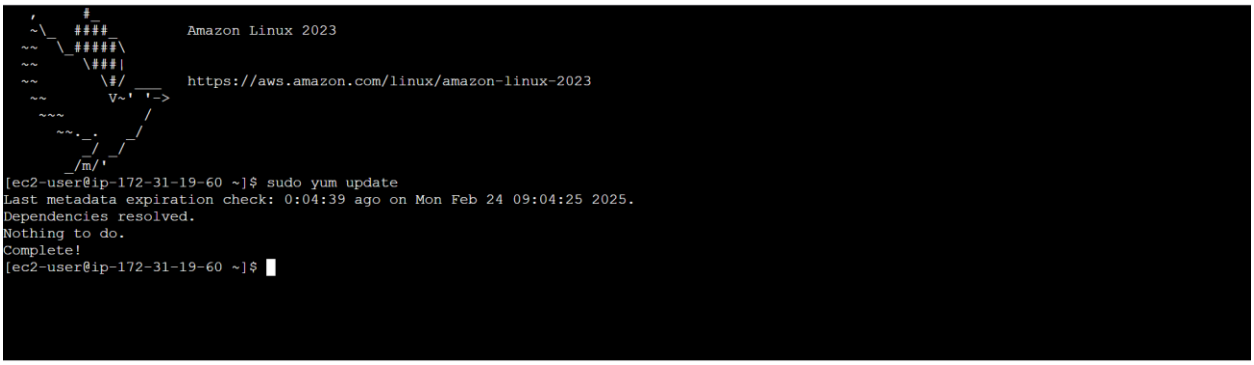
- Switch to root user (sudo su -).



- Run aws configure and provide IAM user credentials.

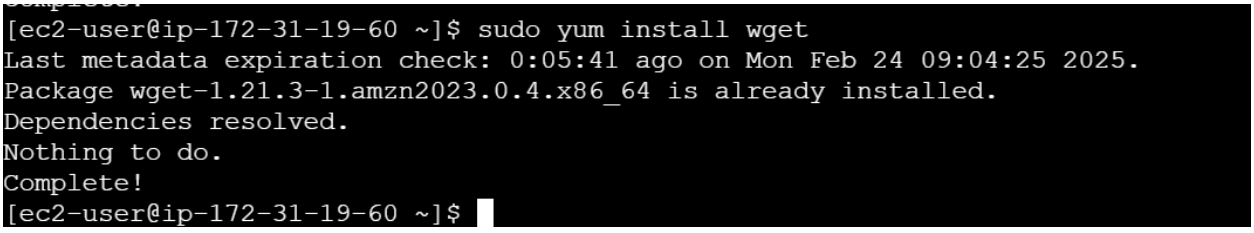
5. Install CodeDeploy Agent (Production Instance):

- Purpose: Install CodeDeploy agent on the production instance.
- Action:
  - Connect to the production instance via SSH.
  - Follow the AWS documentation:
    - yum update –y



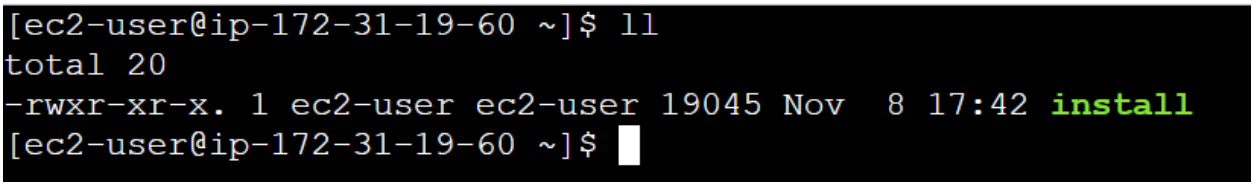
- yum install ruby

- yum install wget

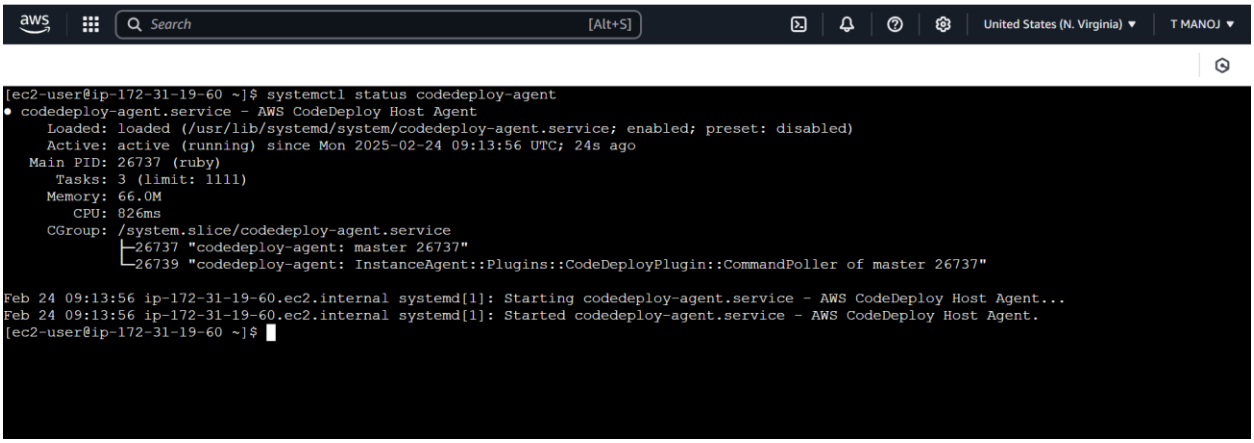


- wget <CodeDeploy agent download link>

- chmod +x ./install
- sudo ./install auto

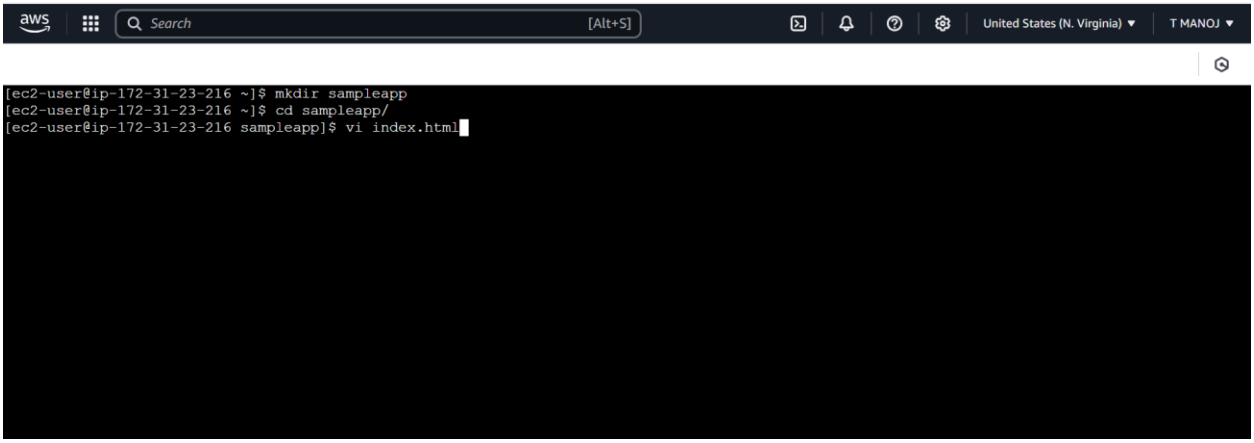


- systemctl status codedeploy-agent
- systemctl start codedeploy-agent

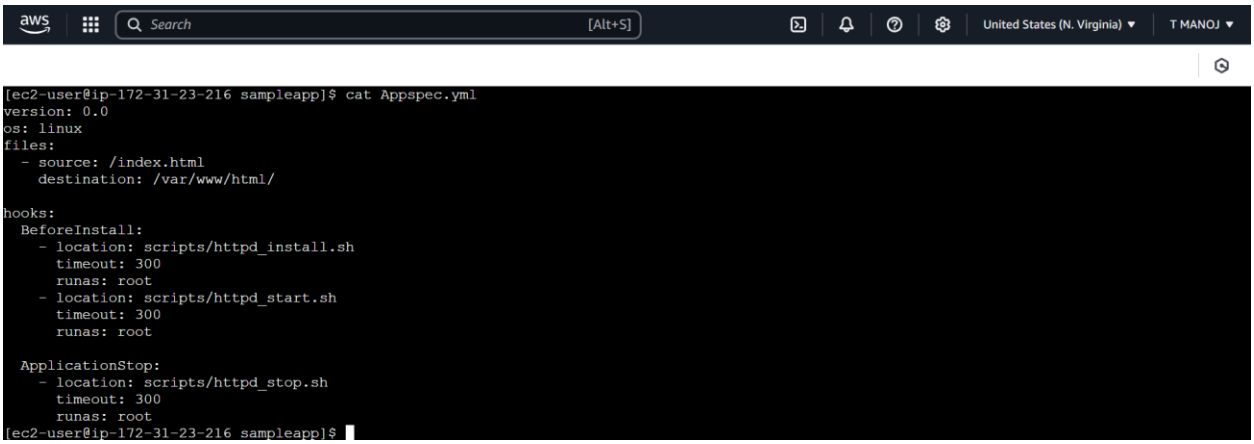


6. Create Sample Code:

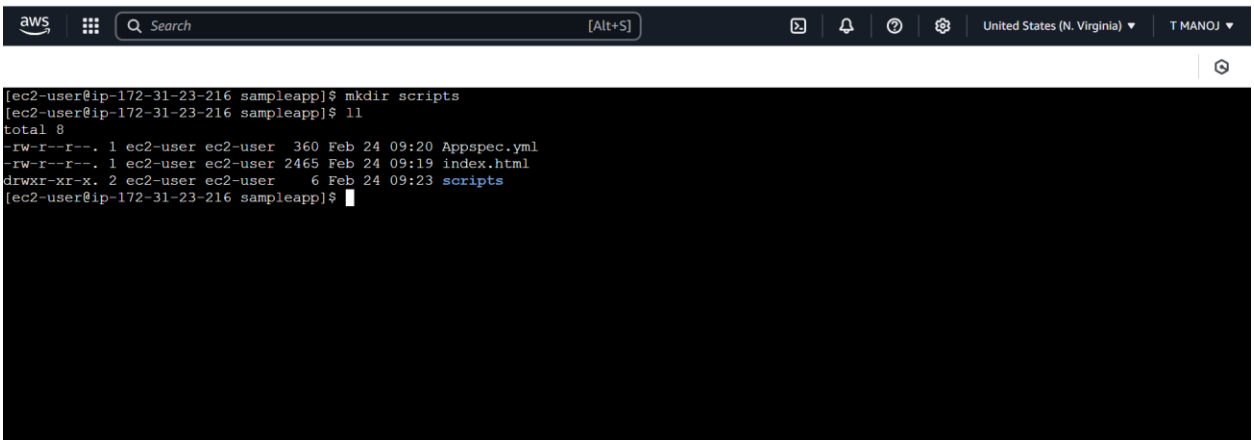
- Purpose: Create application files and deployment scripts.
- Action:
  - On the developer instance:
    - Create index.html.

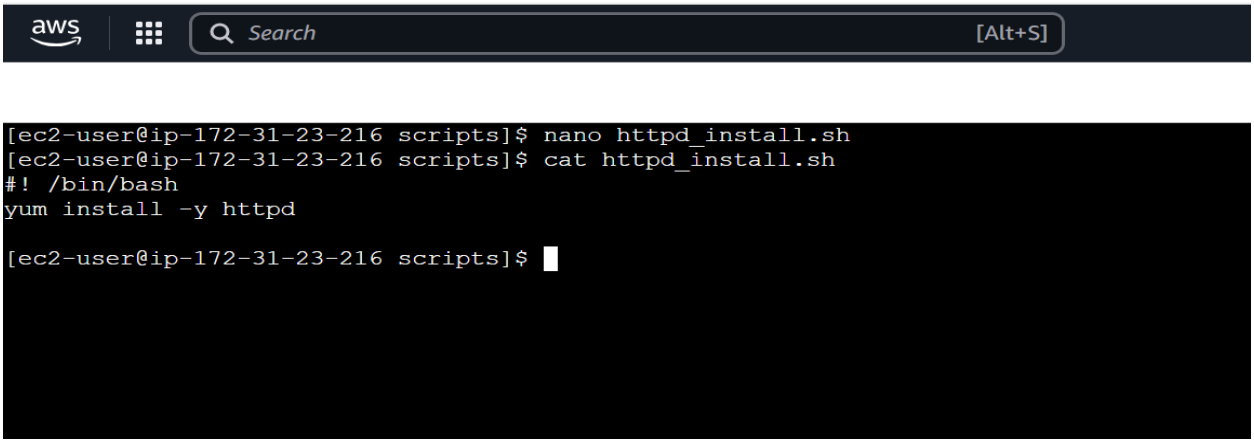


- Create appspec.yml.

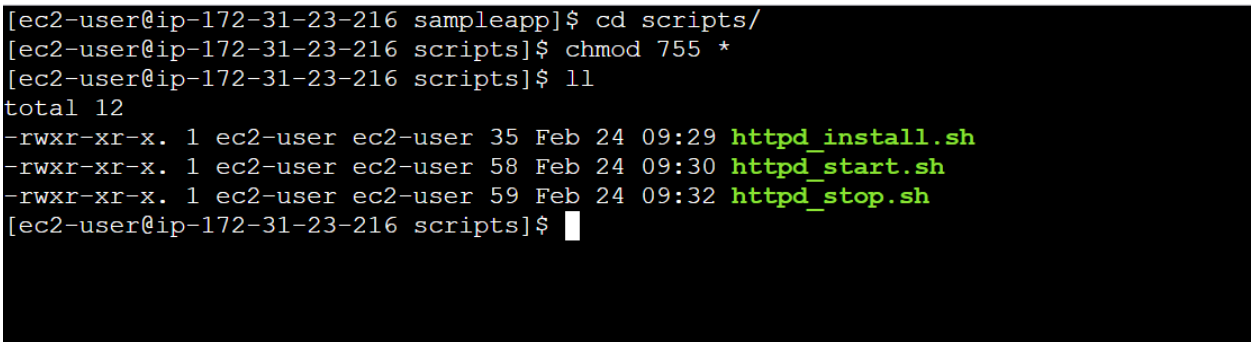


- Create httpd_start.sh and httpd_stop.sh and httpd_install.sh scripts.



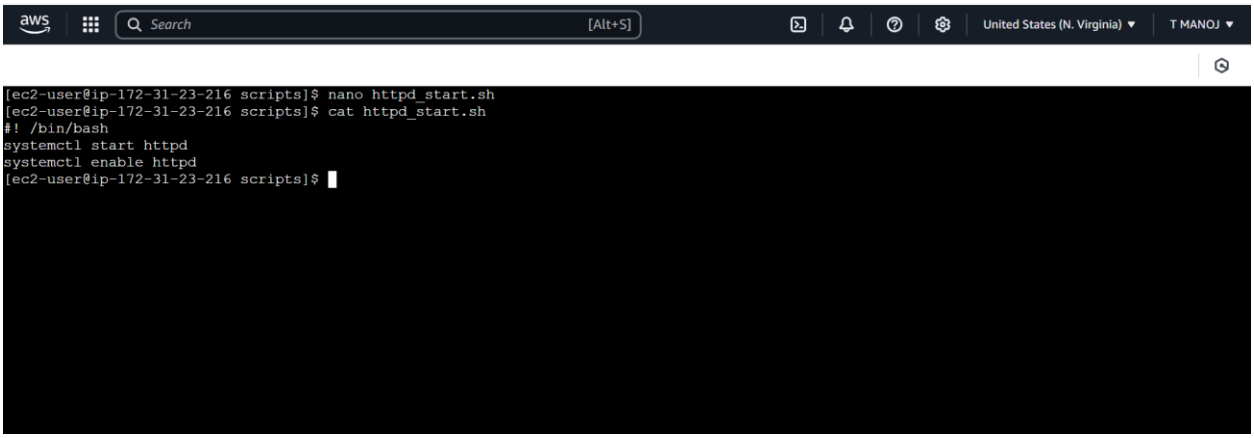


- Make scripts executable (chmod 775 *).



```
[ec2-user@ip-172-31-23-216 scripts]$ cd ../
[ec2-user@ip-172-31-23-216 sampleapp]$ ll
total 8
-rw-r--r--. 1 ec2-user ec2-user  360 Feb 24 09:20 Appspec.yml
-rw-r--r--. 1 ec2-user ec2-user 2465 Feb 24 09:19 index.html
drwxr-xr-x. 2 ec2-user ec2-user   73 Feb 24 09:32 scripts
[ec2-user@ip-172-31-23-216 sampleapp]$ cd scripts/
[ec2-user@ip-172-31-23-216 scripts]$ ll
total 12
-rw-r--r--. 1 ec2-user ec2-user 35 Feb 24 09:29 httpd_install.sh
-rw-r--r--. 1 ec2-user ec2-user 58 Feb 24 09:30 httpd_start.sh
-rw-r--r--. 1 ec2-user ec2-user 59 Feb 24 09:32 httpd_stop.sh
[ec2-user@ip-172-31-23-216 scripts]$
```

- Inside of the http_start.sh file include the commands to start the apache web server, and enable it.

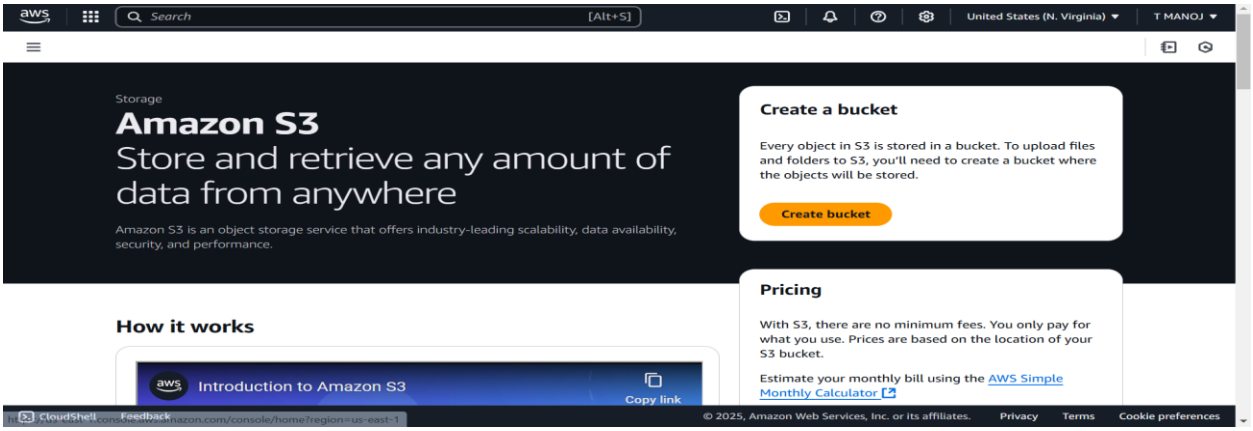


- Inside of the http_stop.sh file include the commands to stop the apache web server, and disable it.

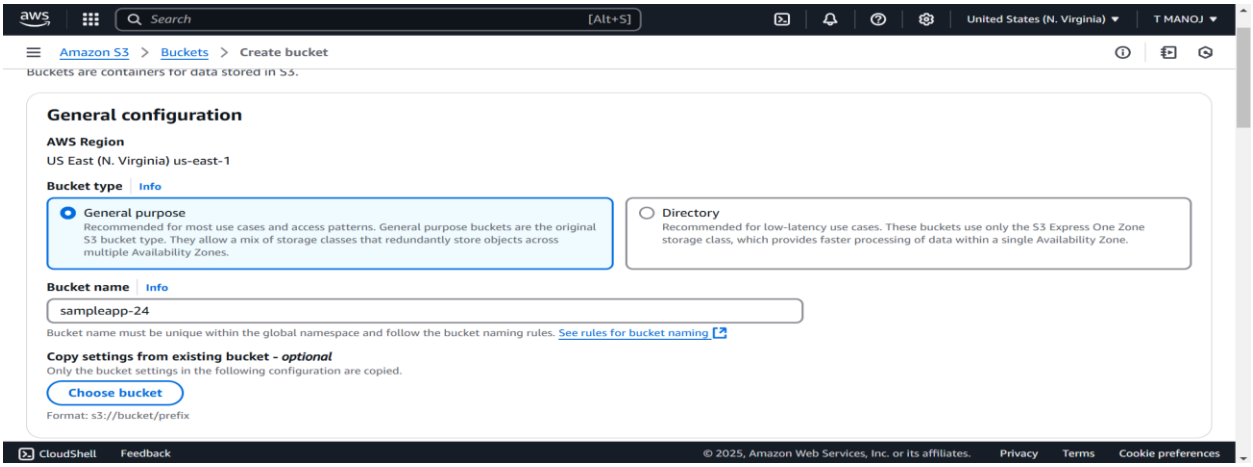
```
[ec2-user@ip-172-31-23-216 scripts]$ cat httpd_stop.sh
#! /bin/bash
systemctl stop httpd
systemctl disable httpd

[ec2-user@ip-172-31-23-216 scripts]$
```

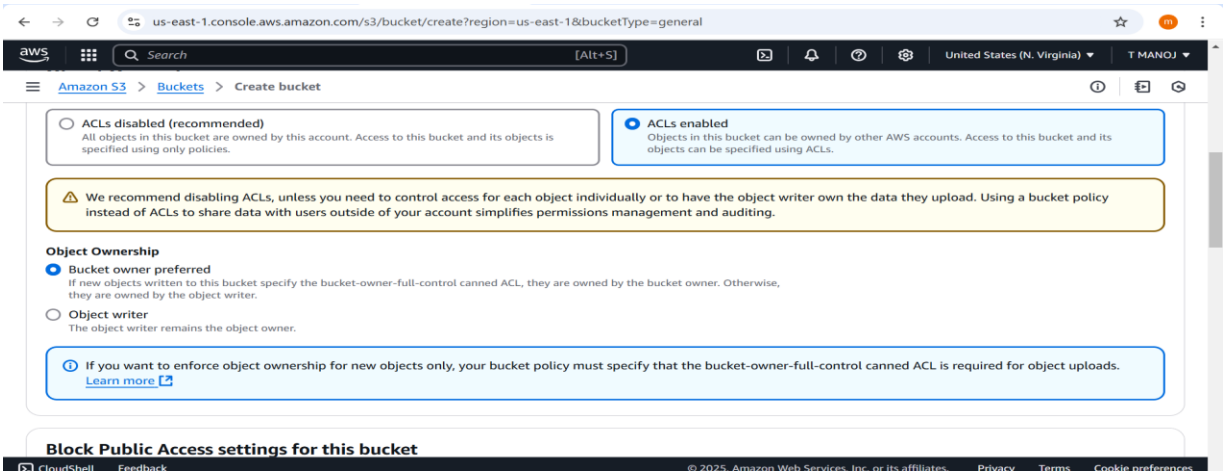
7. Create S3 Bucket:

- Purpose: Store application artifacts.
- Action:
    - Create a public S3 bucket.

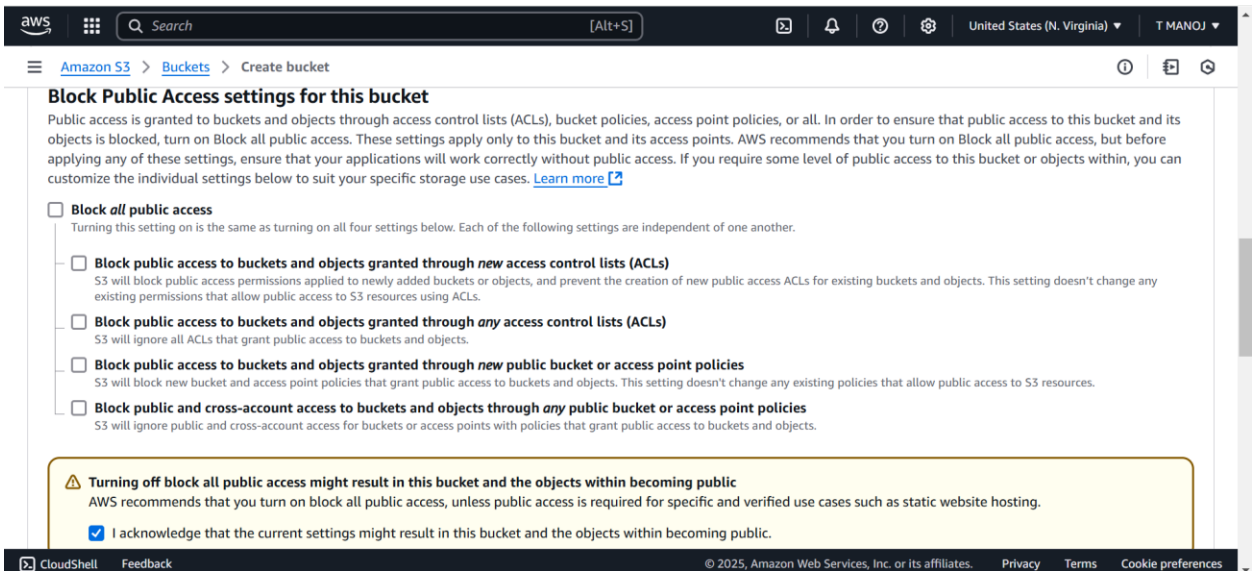


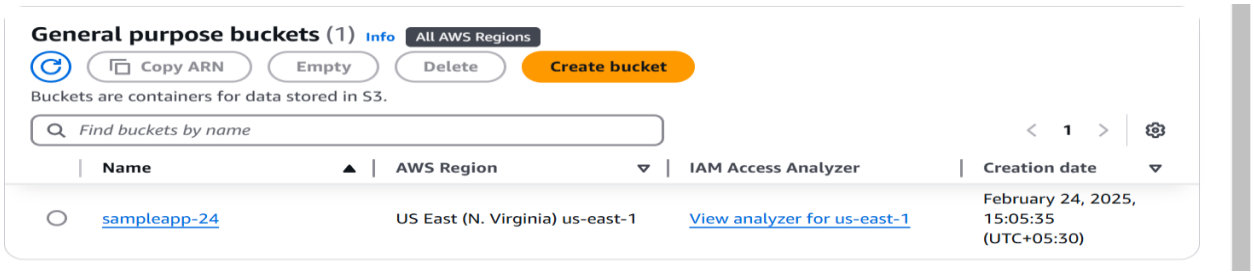
- Enable ACLs.





- Disable encryption.

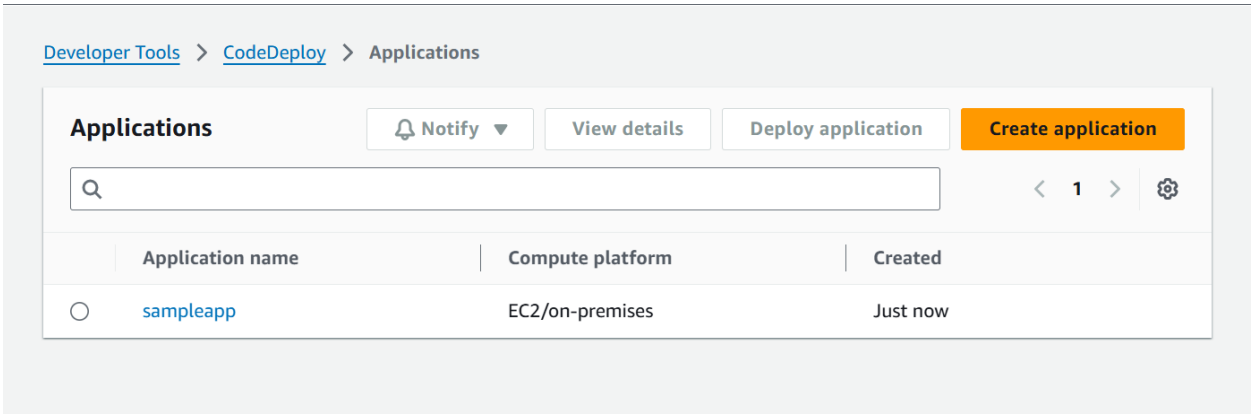




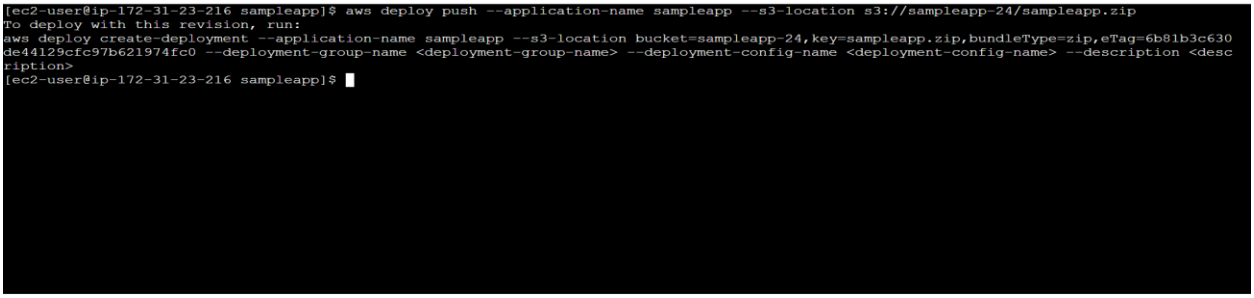
8. Create CodeDeploy Application:

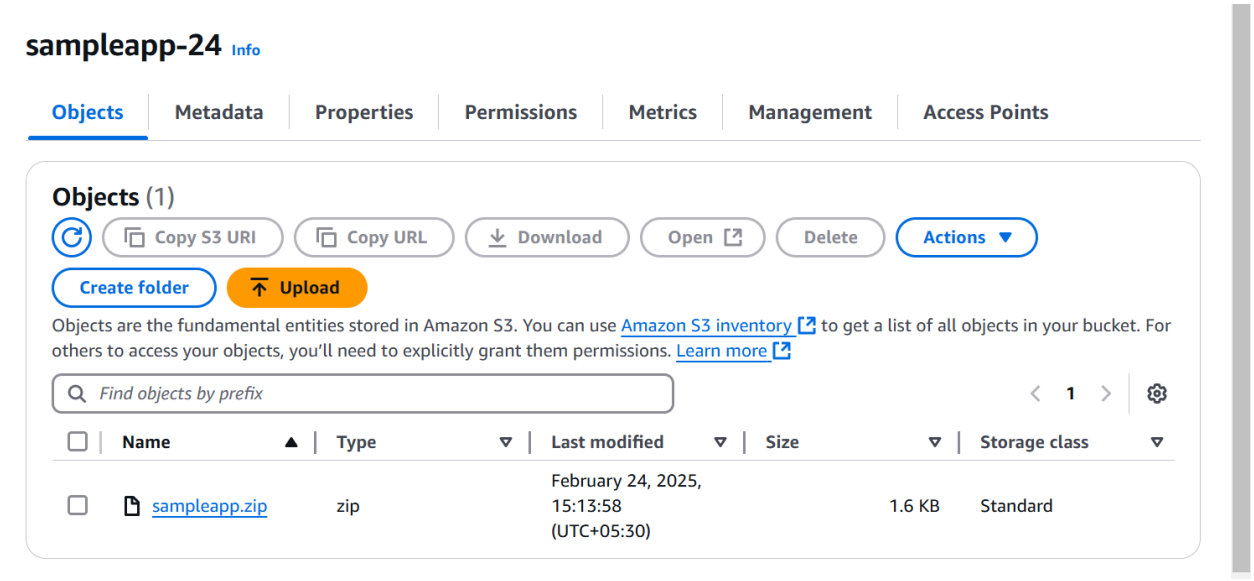
- Purpose: Manage application deployments.
- Action:
    - On the developer machine:
        - Run aws deploy create-application --application-name sampleapp.

```
[ec2-user@ip-172-31-23-216 sampleapp]$ aws deploy create-application --application-name sampleapp
{
    "applicationId": "435cbe28-c0c3-4879-b232-701f1d1231ff"
}
[ec2-user@ip-172-31-23-216 sampleapp]$
```



- Zip the application files (zip -r sampleapp.zip *).
- Upload to S3: aws deploy push --application-name sampleapp --s3-location s3://gir-sampleapp/sampleapp.zip.

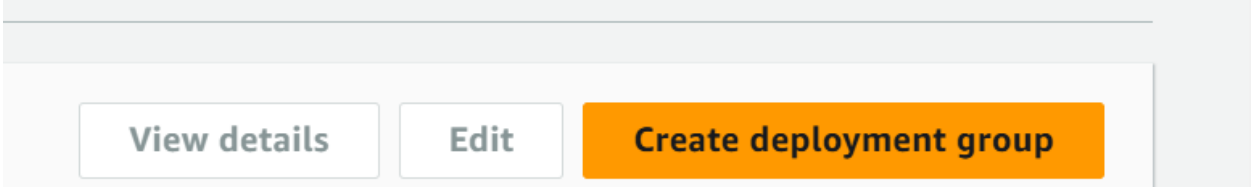




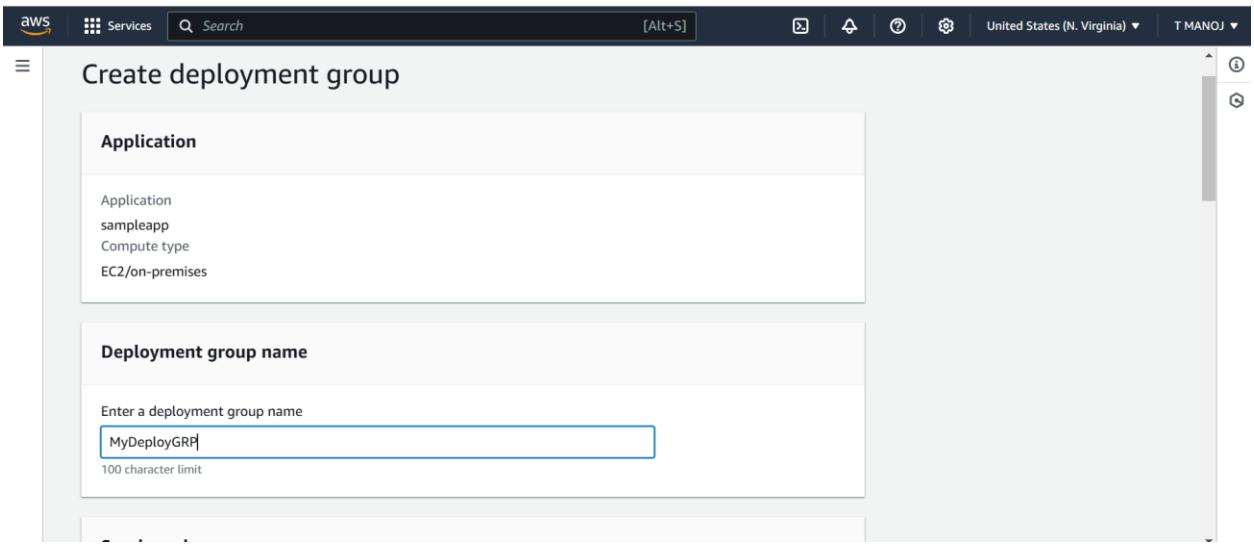
9. Create Deployment Group:

- Purpose: Define deployment settings.
- Action:

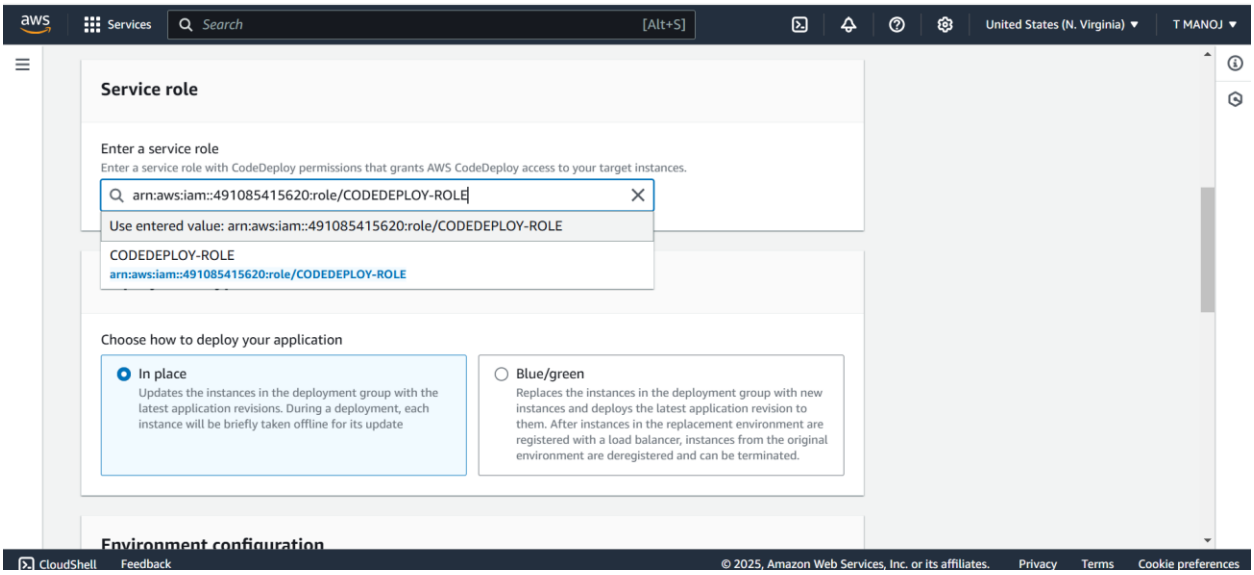
- In the CodeDeploy console:
  - Create a deployment group.



  - Provide a name.

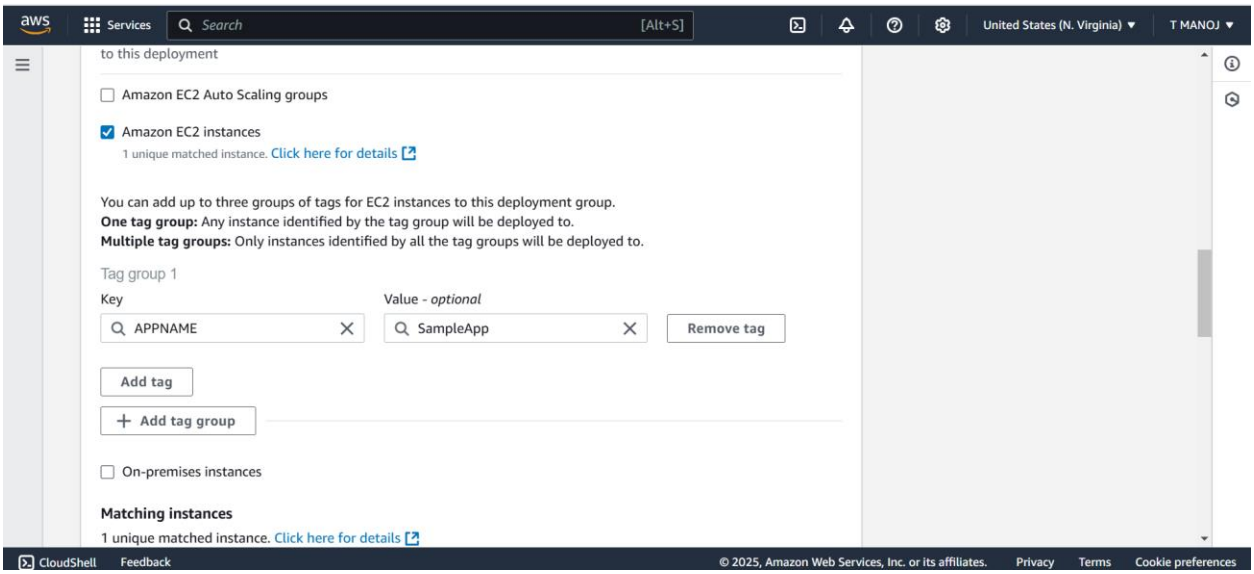


  - Attach the "codedeploy" service role.

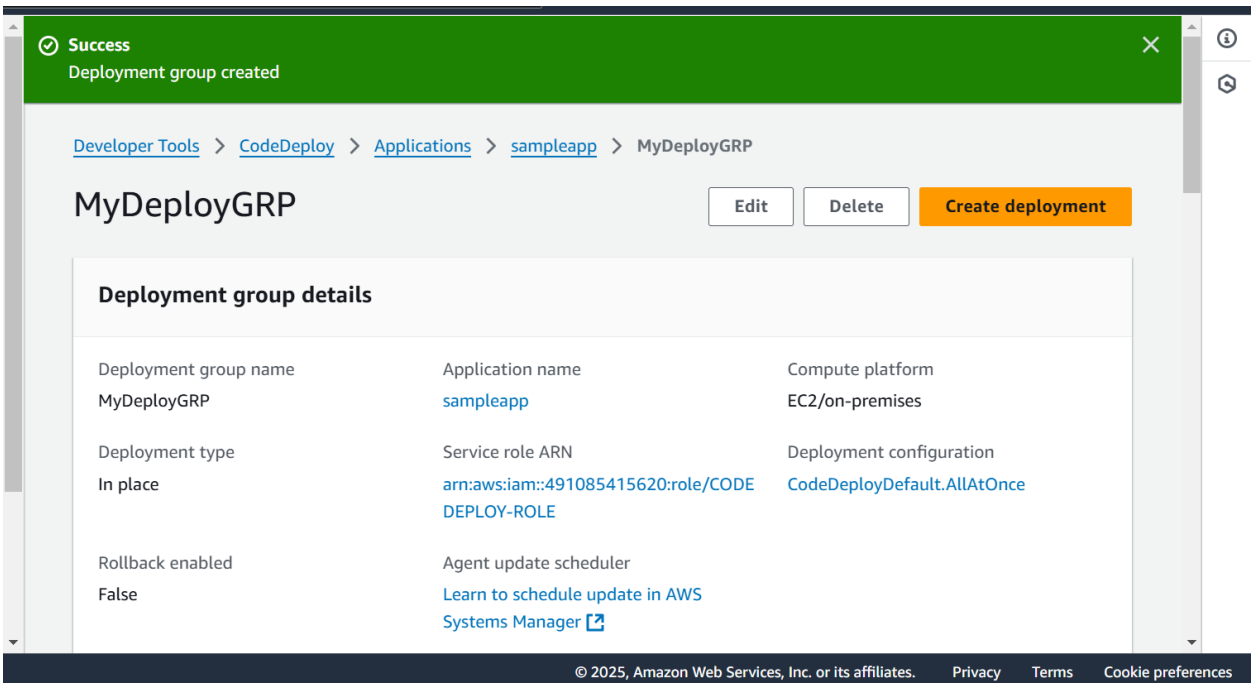


  - Select "Amazon EC2 instances."

- Choose "In-place" deployment.
- Select the production instance.



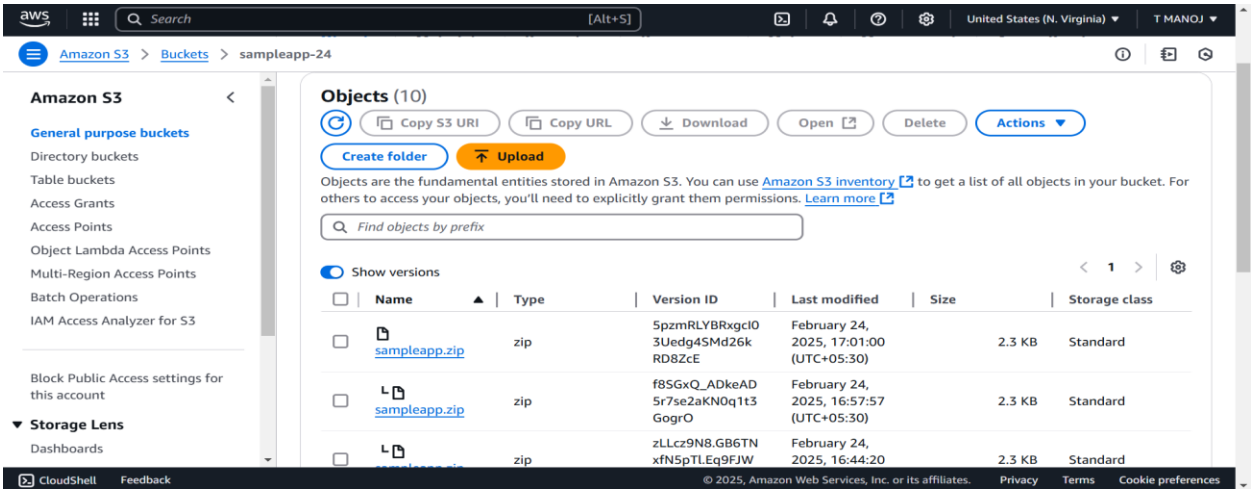
- Disable load balancing.

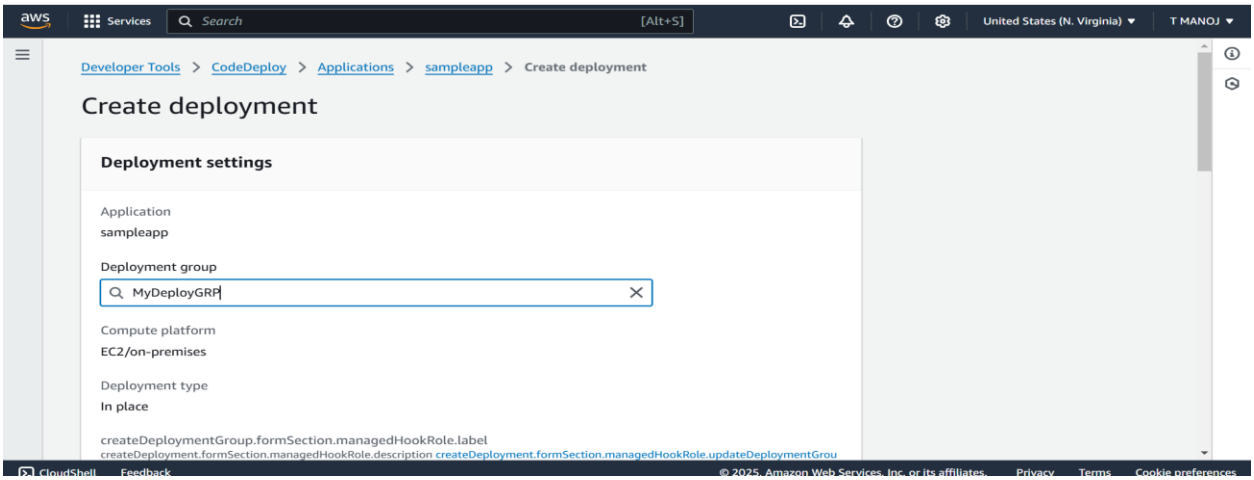


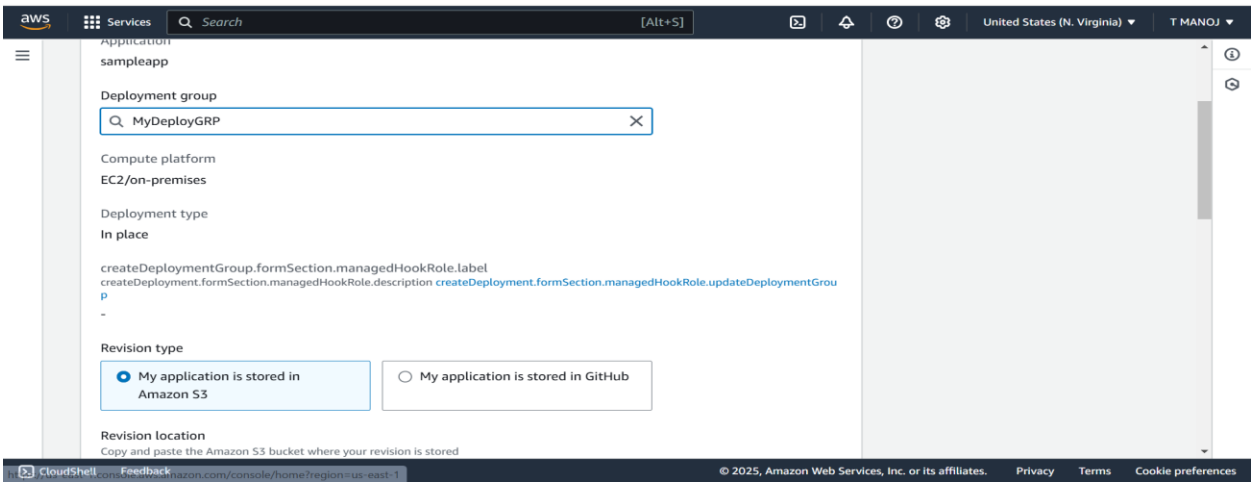
10. Create Deployment:

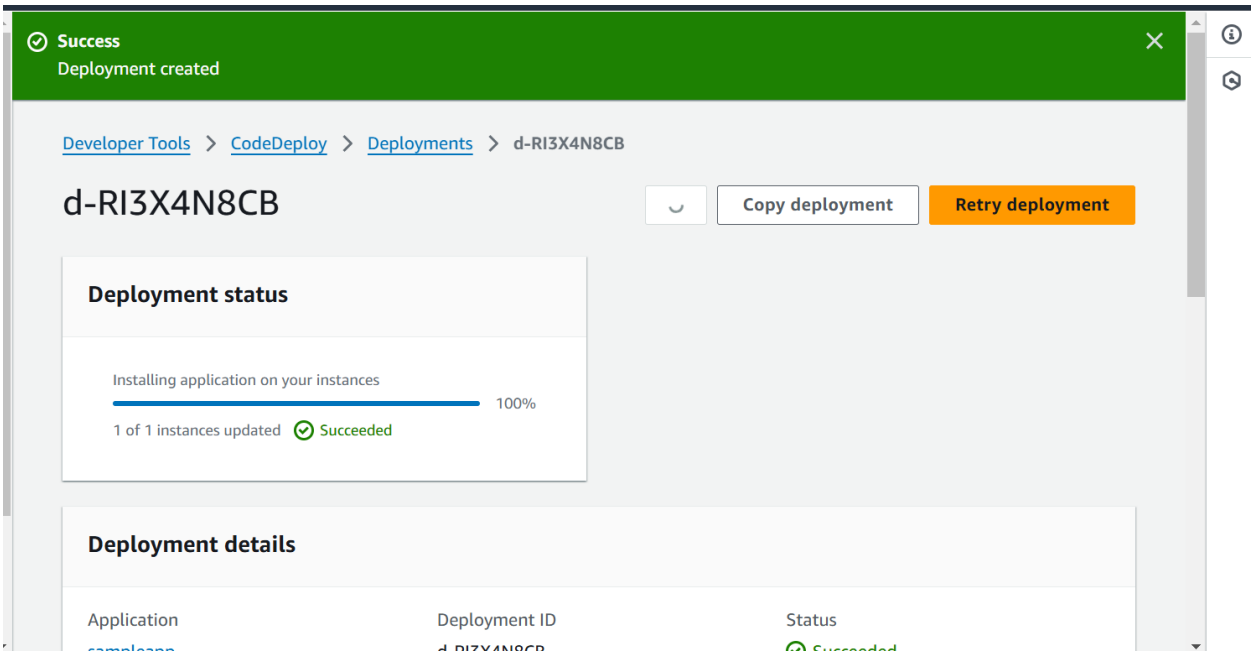
- Purpose: Deploy the application.
- Action:
  - In the CodeDeploy console:
    - Create a deployment.

- Select the S3 bucket and sampleapp.zip.



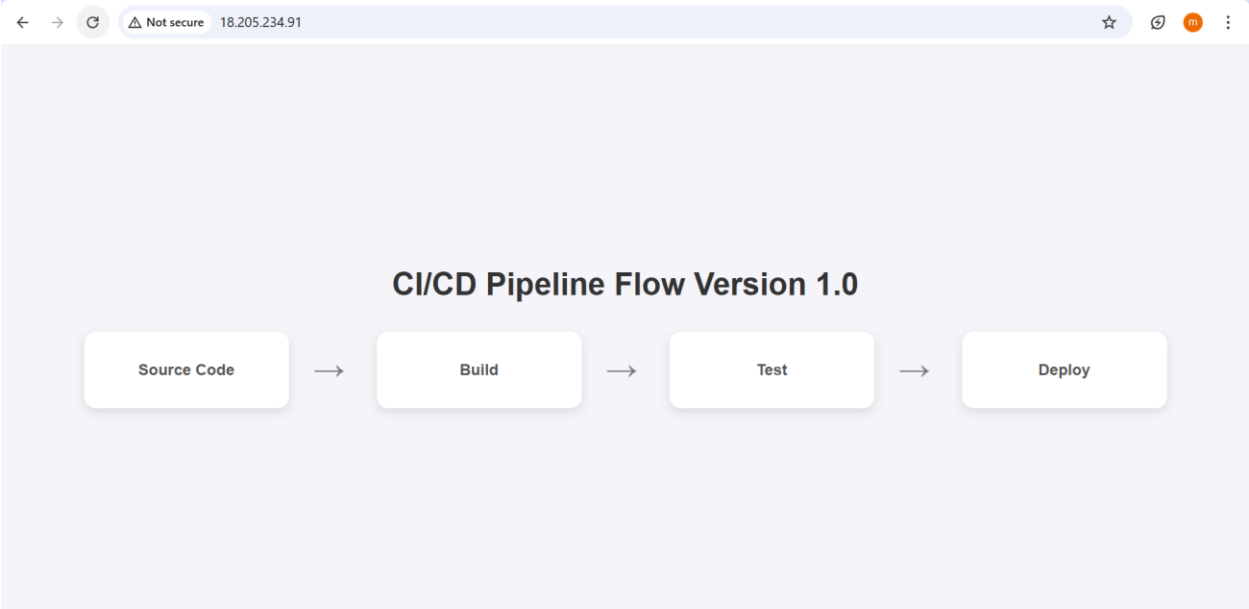






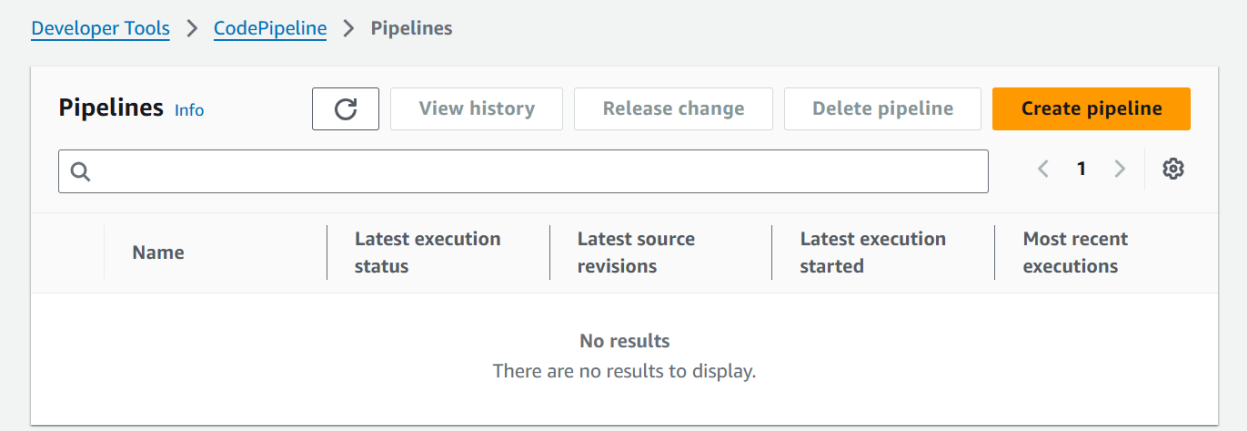
11. Test the Output:

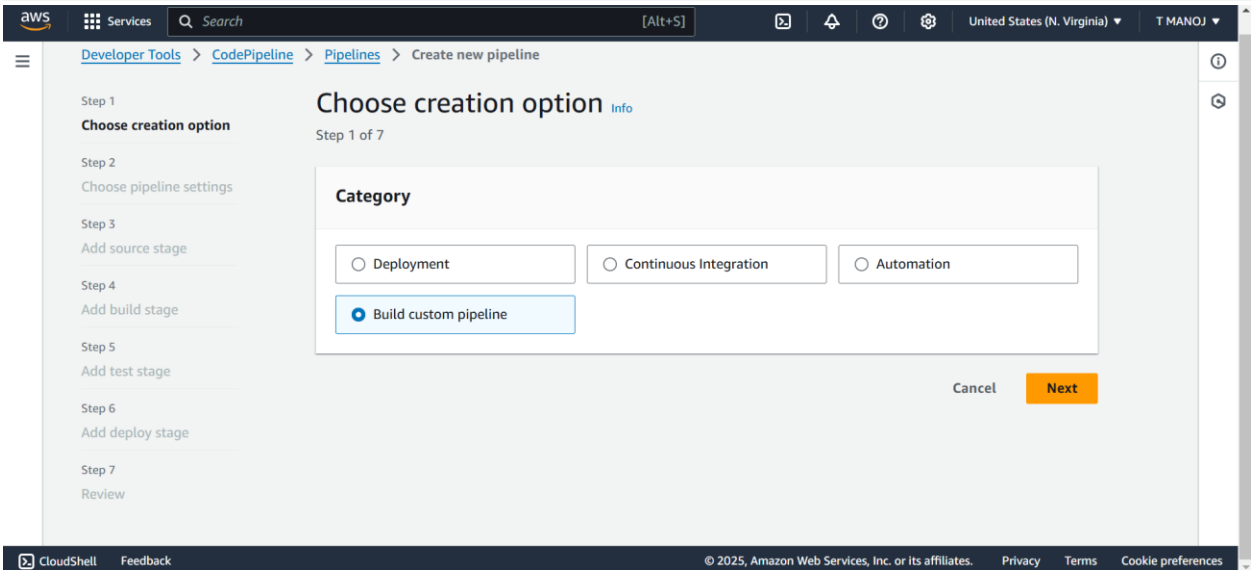
- Purpose: Verify the deployed application.
- Action:
    - Get the production instance's IP address.
    - Open a browser and enter the IP address.

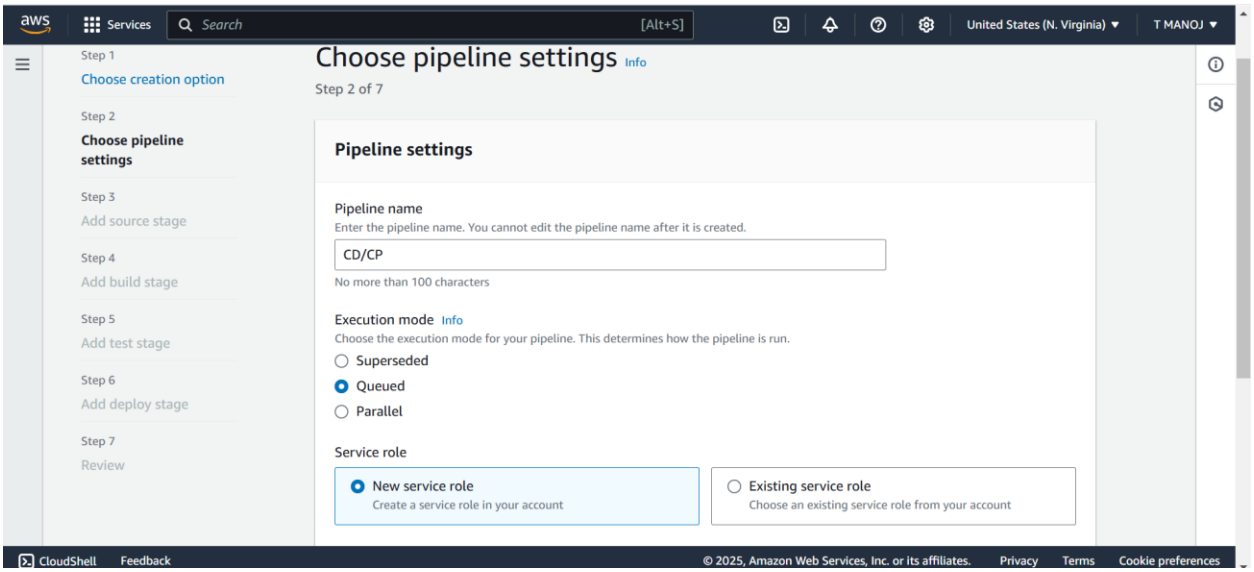


12. Create CodePipeline:

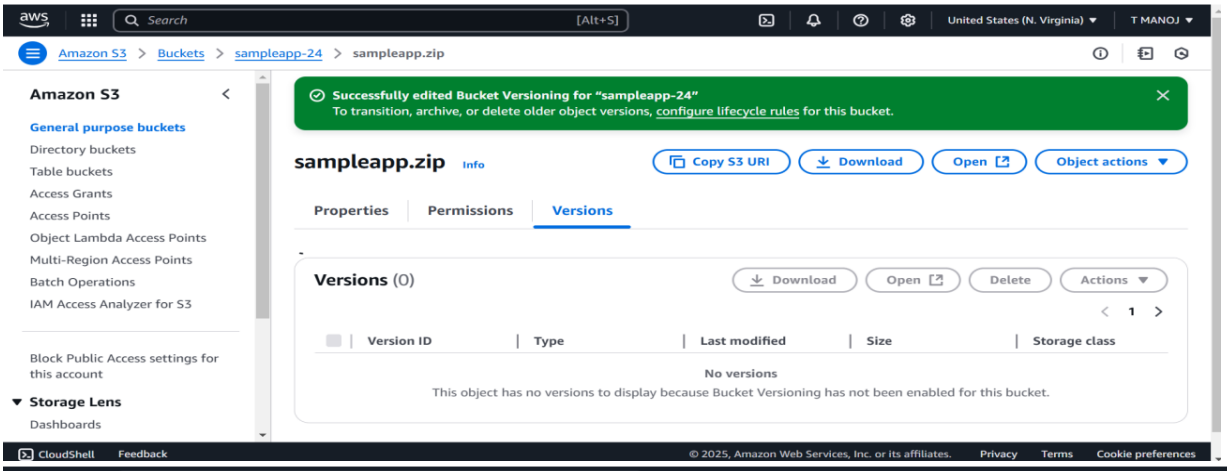
- Purpose: Automate the deployment pipeline.
- Action:
    - In the CodePipeline console:
        - Create a pipeline.

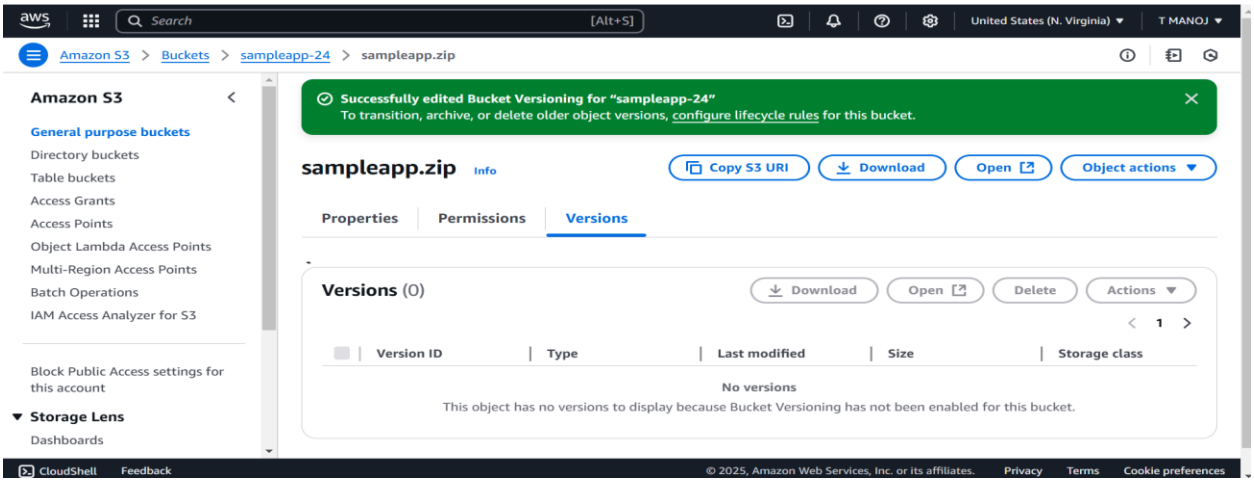


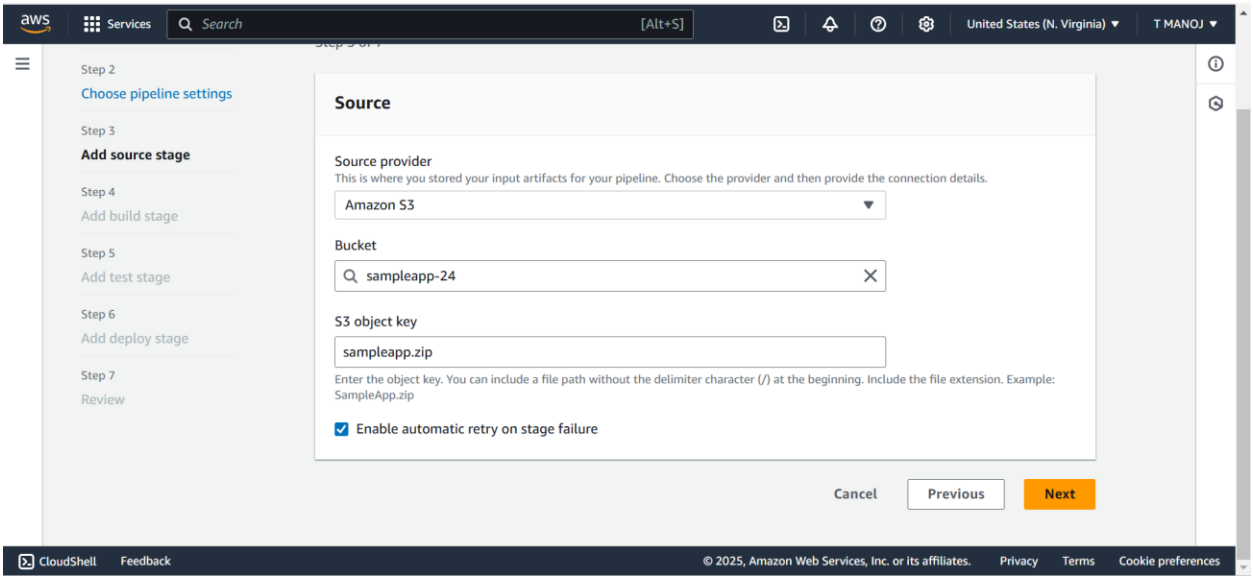




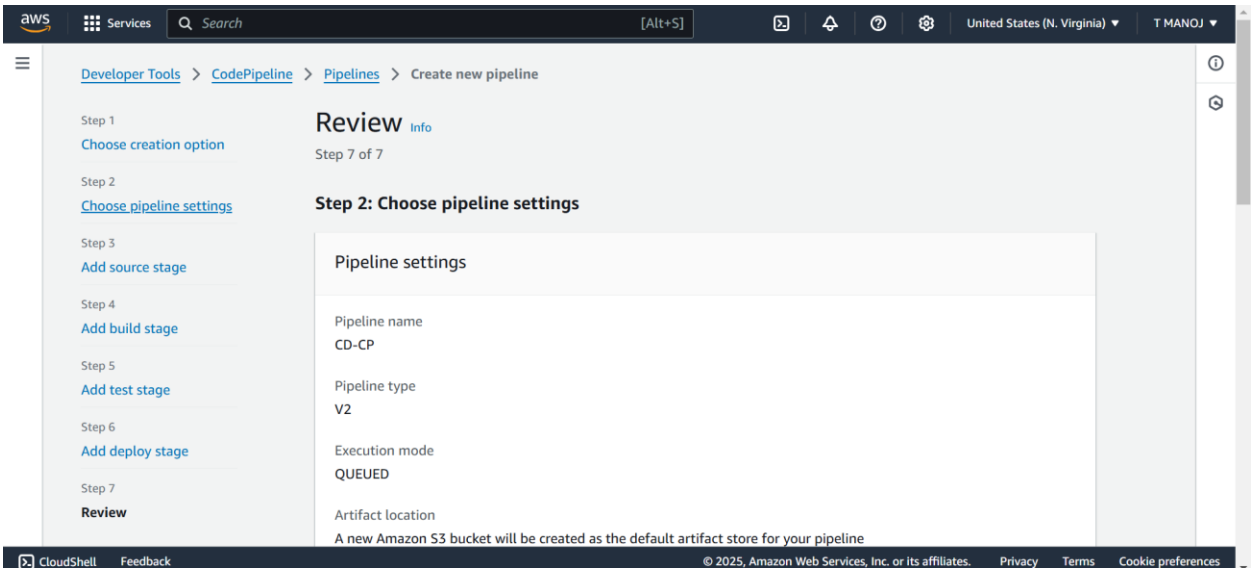
- Source stage: Select S3, enable bucket versioning.

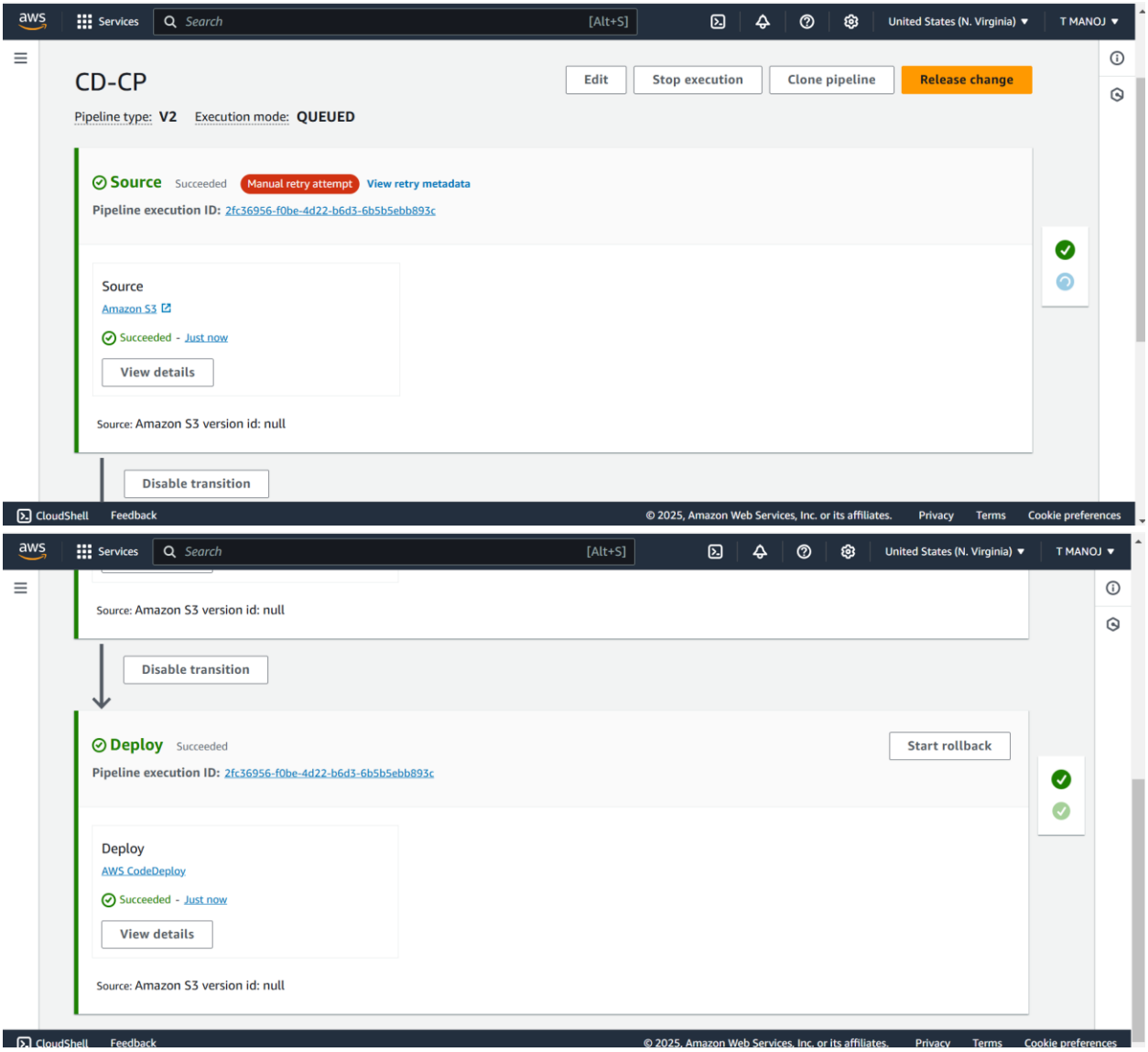






- Skip build and test stages.
- Deploy stage: Select CodeDeploy, choose the application and deployment group.



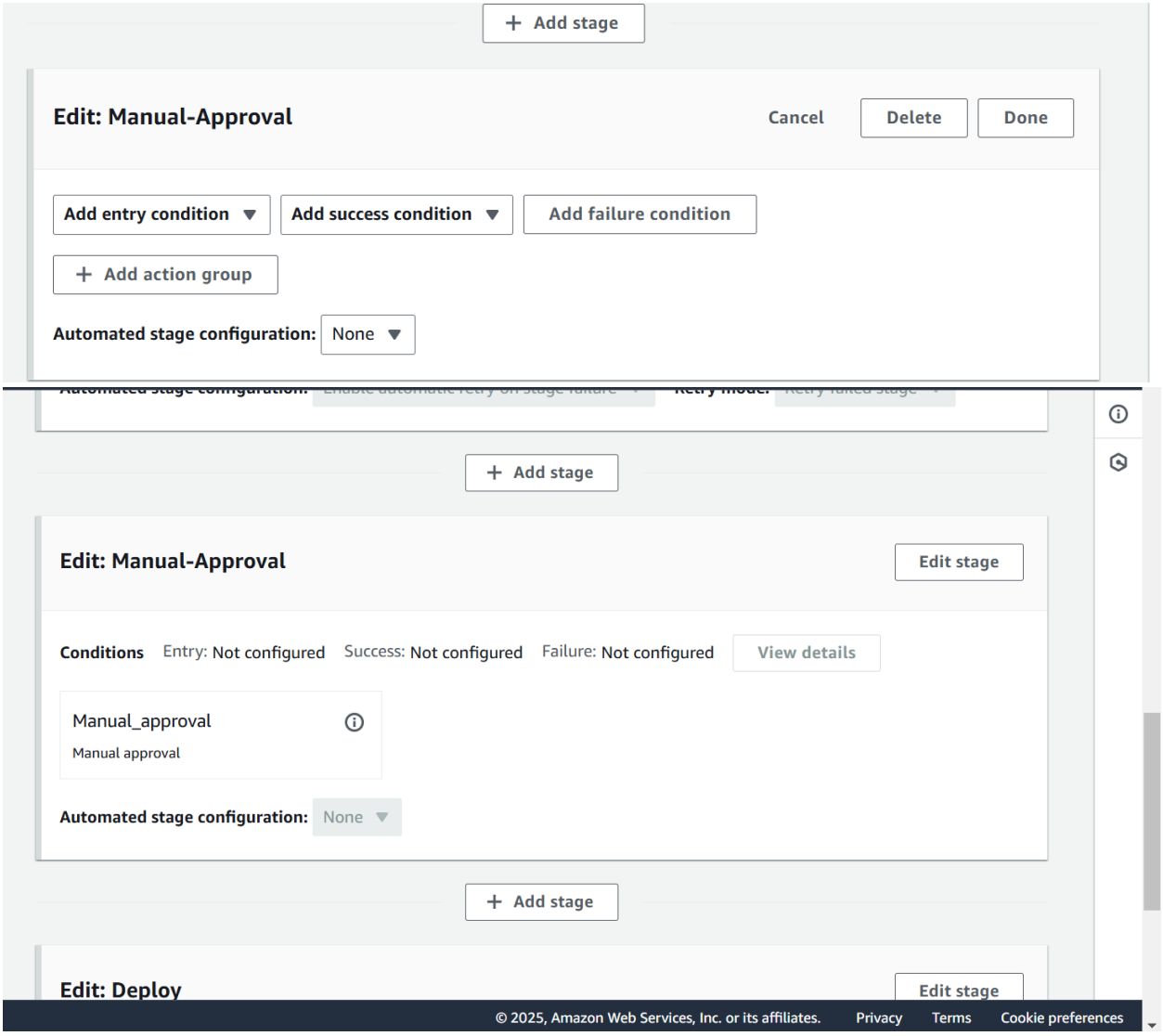


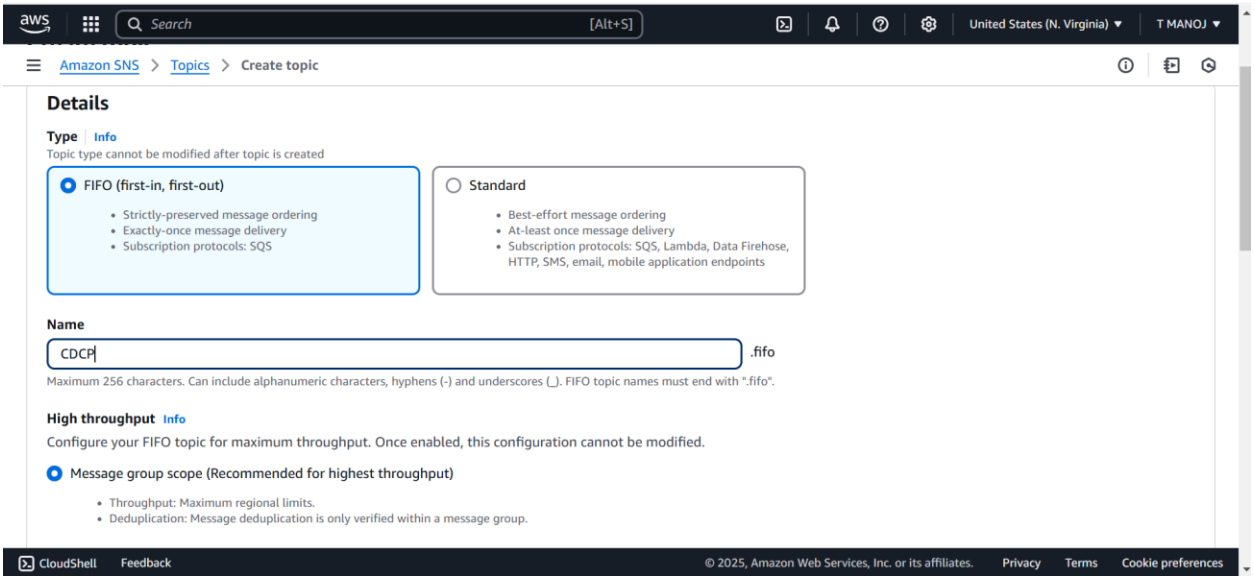
13. Testing:

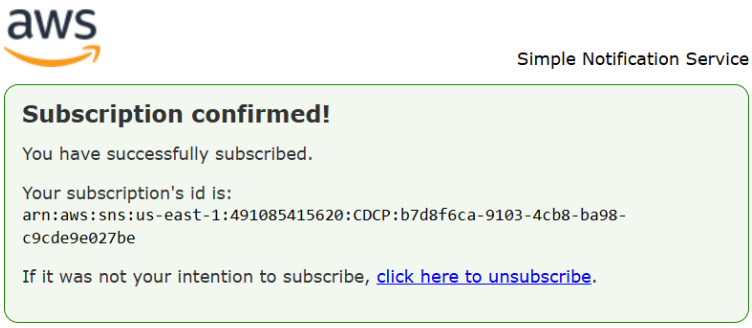
- Purpose: Test the CodePipeline.
- Action:
    - On the developer instance:
        - Change the source code.
        - Zip the application.
        - Upload the zip to S3.
        - The CodePipeline should trigger a deployment.

Enhancements:

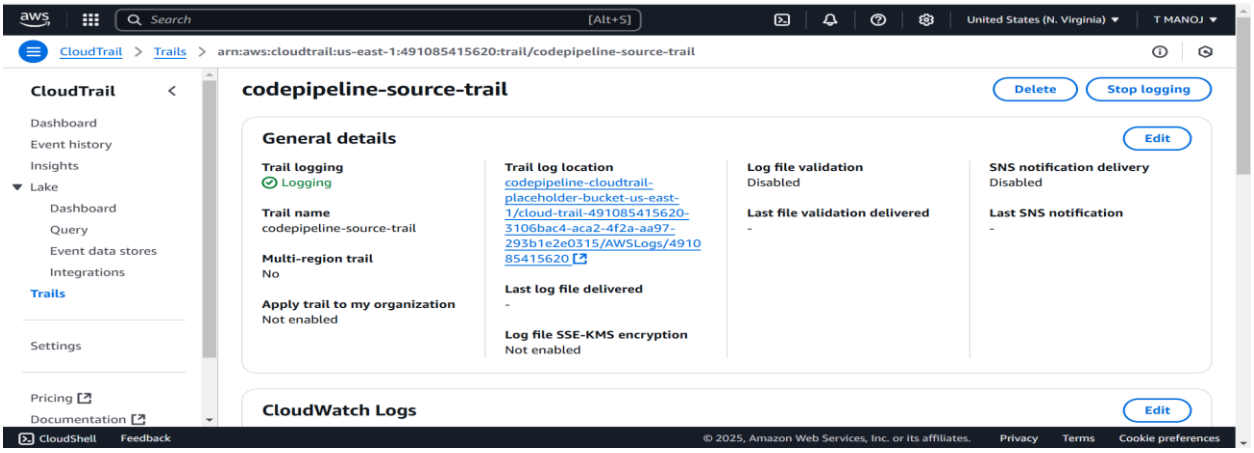
- Manual Approval: Add an approval stage in CodePipeline.

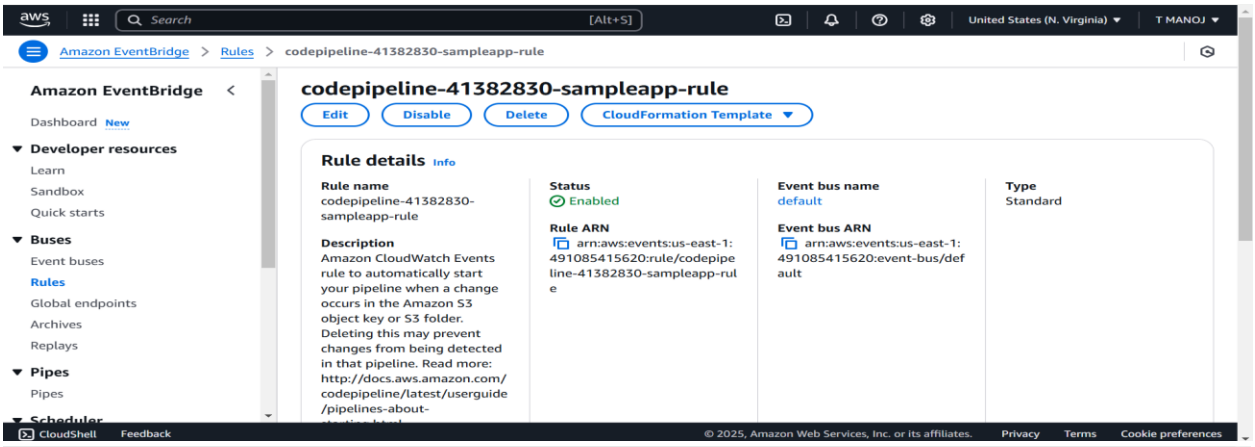


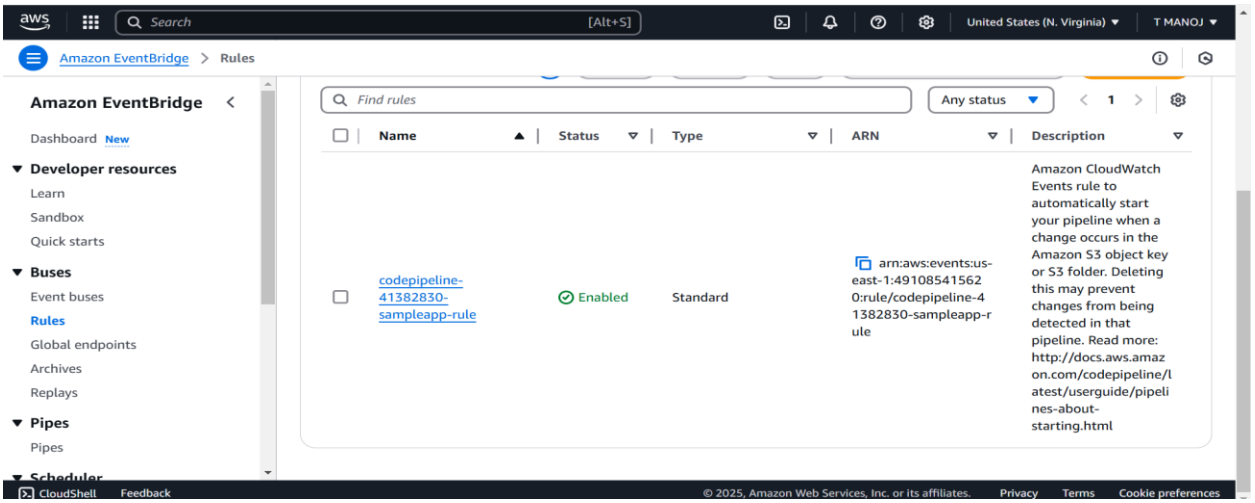




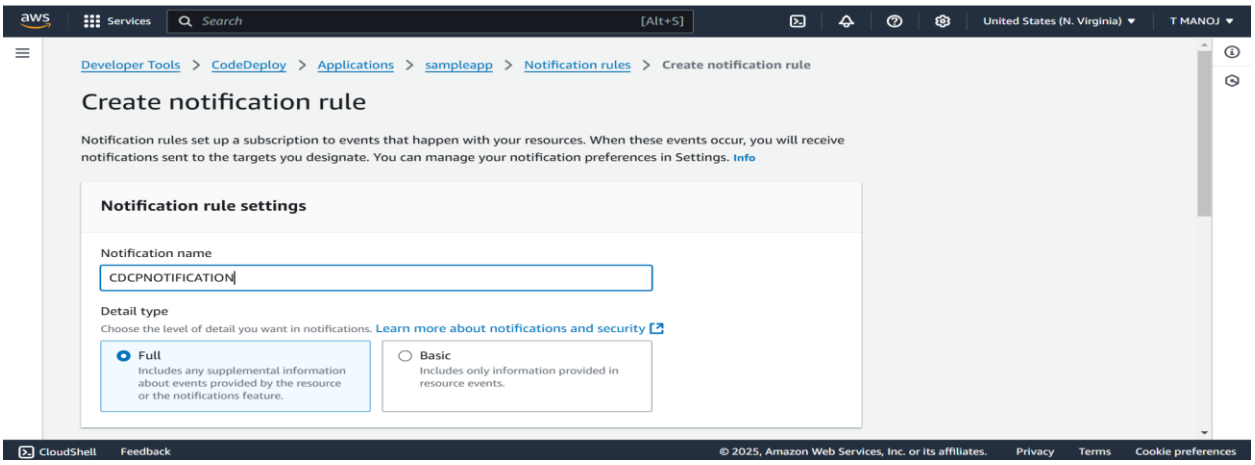
- CloudWatch Monitoring and Notifications: Set up CloudWatch alarms and SNS notifications for CodeDeploy and CodePipeline.

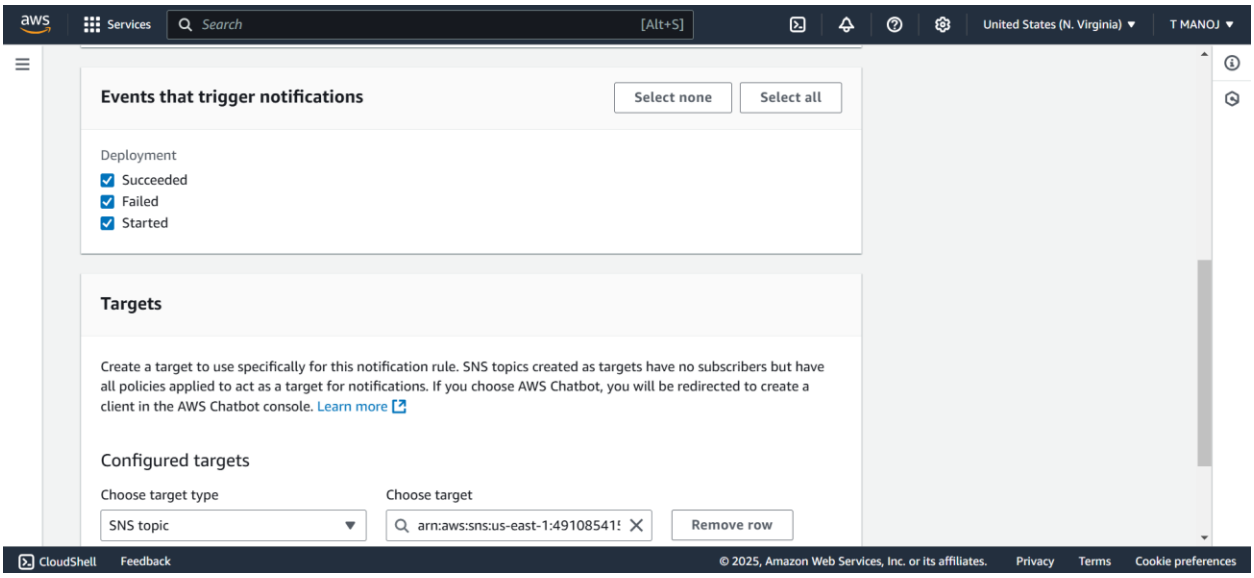


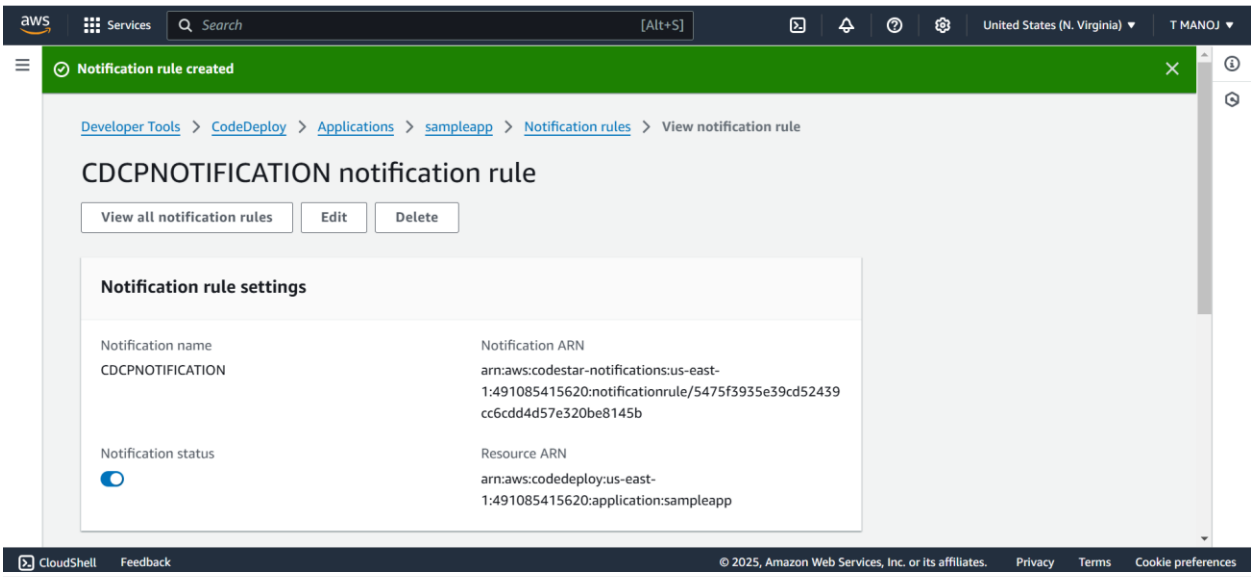


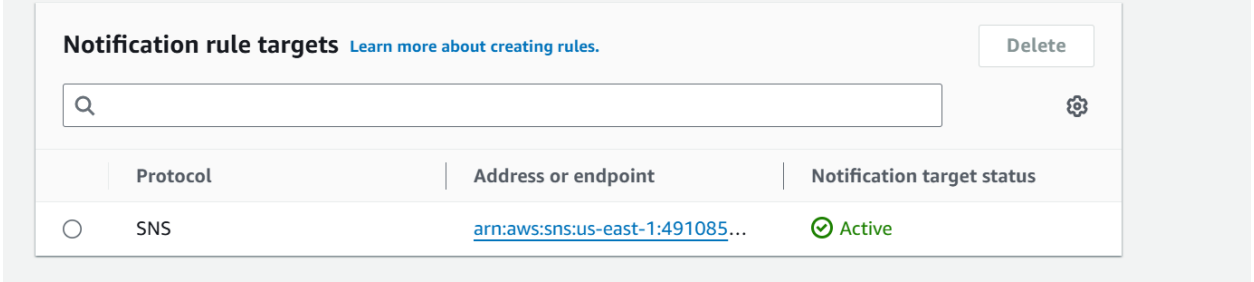


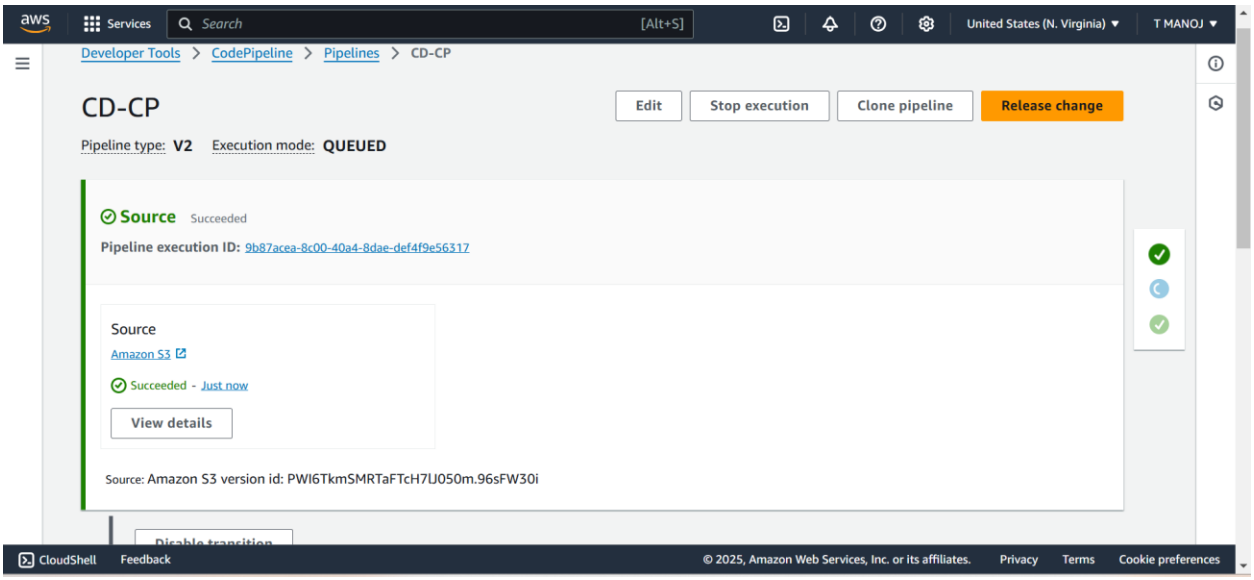
- Notifications: Configure SNS notifications for deployment success and failure.

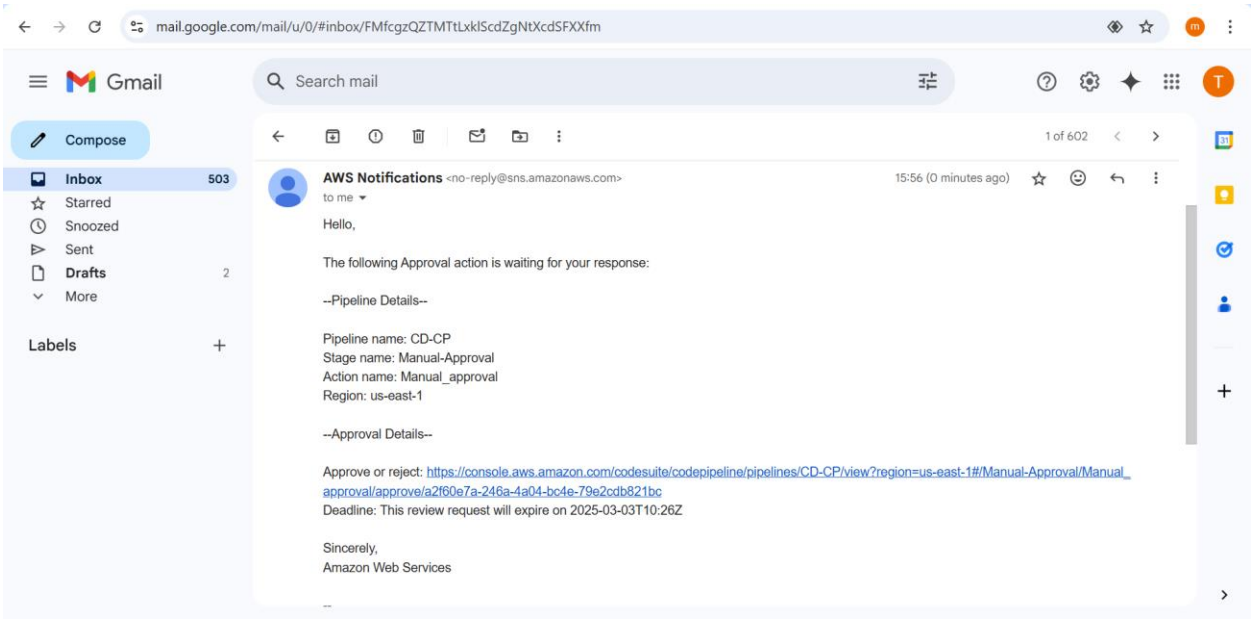



aws
Services
Search
[Alt+S]
United States (N. Virginia)
T MANOJ
Events that trigger notifications
Select none
Select all
Deployment
Succeeded
Failed
Started
Targets
Create a target to use specifically for this notification rule. SNS topics created as targets have no subscribers but have all policies applied to act as a target for notifications. If you choose AWS Chatbot, you will be redirected to create a client in the AWS Chatbot console. Learn more
Configured targets
Choose target type
SNS topic
Choose target
arn:aws:sns:us-east-1:49108541
Remove row
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
United States (N. Virginia)
T MANOJ
Notification rule created
Developer Tools > CodeDeploy > Applications > sampleapp > Notification rules > View notification rule
CDCPNOTIFICATION notification rule
View all notification rules
Edit
Delete
Notification rule settings
Notification name
CDCPNOTIFICATION
Notification ARN
arn:aws:codestar-notifications:us-east-1:491085415620:notificationrule/5475f3935e39cd52439cc6cdd4d57e320be8145b
Notification status
Resource ARN
arn:aws:codedeploy:us-east-1:491085415620:application:sampleapp
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Notification rule targets
Learn more about creating rules.
Delete
Protocol
Address or endpoint
Notification target status
SNS
arn:aws:sns:us-east-1:491085...
Active


aws
Services
Search
[Alt+S]
United States (N. Virginia)
T MANOJ
Developer Tools > CodePipeline > Pipelines > CD-CP
CD-CP
Edit
Stop execution
Clone pipeline
Release change
Pipeline type: V2
Execution mode: QUEUED
Source
Succeeded
Pipeline execution ID: 9b87acea-8c00-40a4-8dae-def4f9e56317
Source
Amazon S3
Succeeded - Just now
View details
Source: Amazon S3 version id: PWI6TkmSMRTaFTcH7IJ050m.96sFW30i
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


mail.google.com/mail/u/0/#inbox/FMfcgzQZTMTtLxklScdZgNtXcdSFXXfm
Gmail
Search mail
Compose
Inbox 503
Starred
Snoozed
Sent
Drafts 2
More
Labels
1 of 602
AWS Notifications <no-reply@sns.amazonaws.com>
15:56 (0 minutes ago)
to me
Hello,
The following Approval action is waiting for your response:
--Pipeline Details--
Pipeline name: CD-CP
Stage name: Manual-Approval
Action name: Manual_approval
Region: us-east-1
--Approval Details--
Approve or reject: https://console.aws.amazon.com/codesuite/codepipeline/pipelines/CD-CP/view?region=us-east-1#/Manual-Approval/Manual_approval/approve/a2f60e7a-246a-4a04-bc4e-79e2cdb821bc
Deadline: This review request will expire on 2025-03-03T10:26Z
Sincerely,
Amazon Web Services

Review

Action name: Manual_approval Status: Waiting for approval

Details | Revisions

Trigger

CloudWatchEvent - codepipeline-41382830-sampleapp-rule

Comments about this action

-

URL for review

Decision

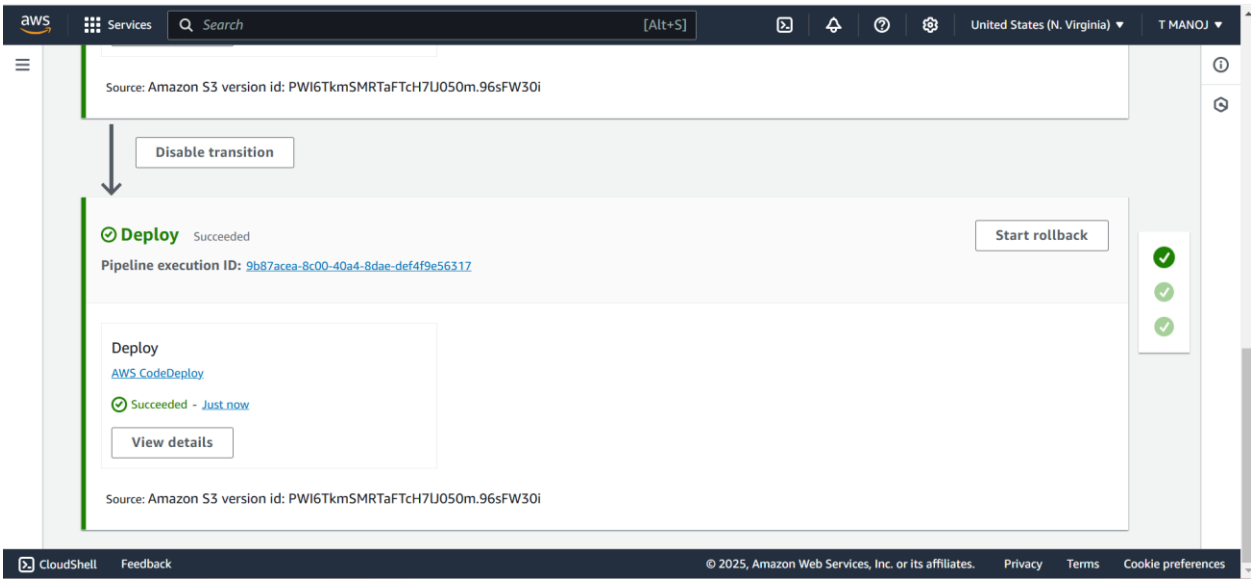
- (•) Approve
  Approving will resume the pipeline execution.
- ( ) Reject
  Rejecting will stop the pipeline execution with a failed status.

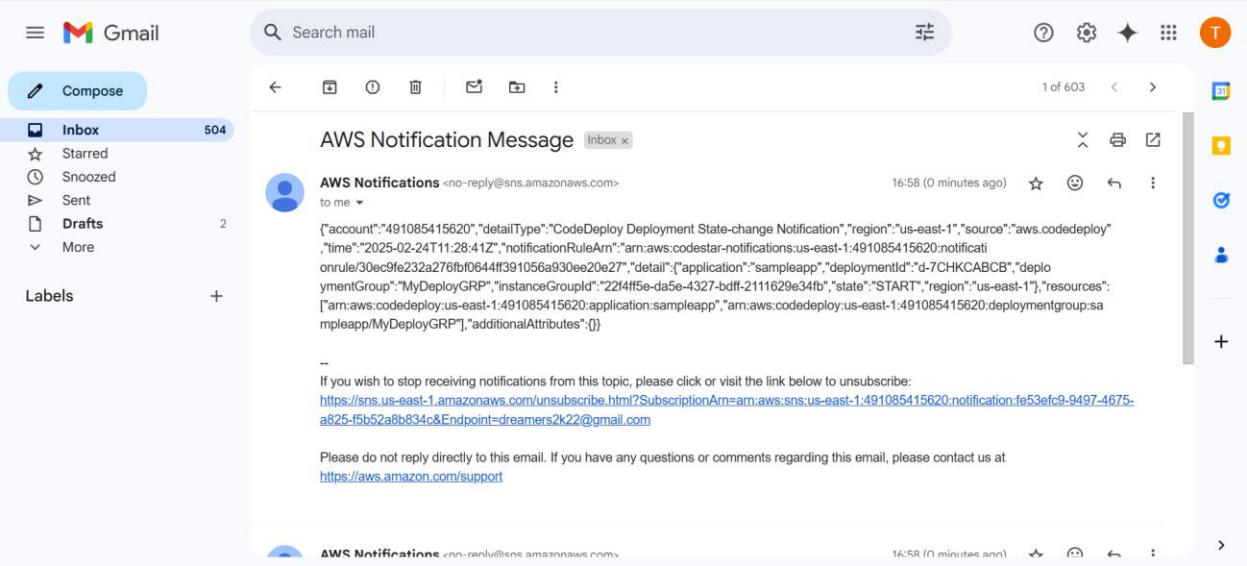
Comments - *optional*

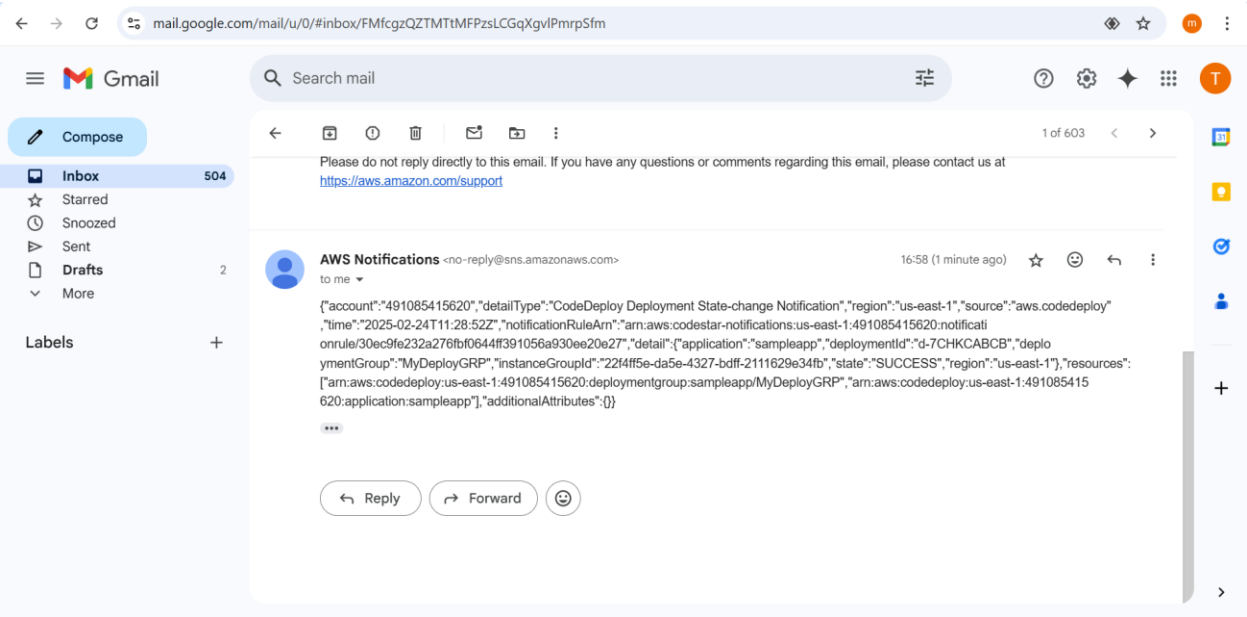
Preview markdown

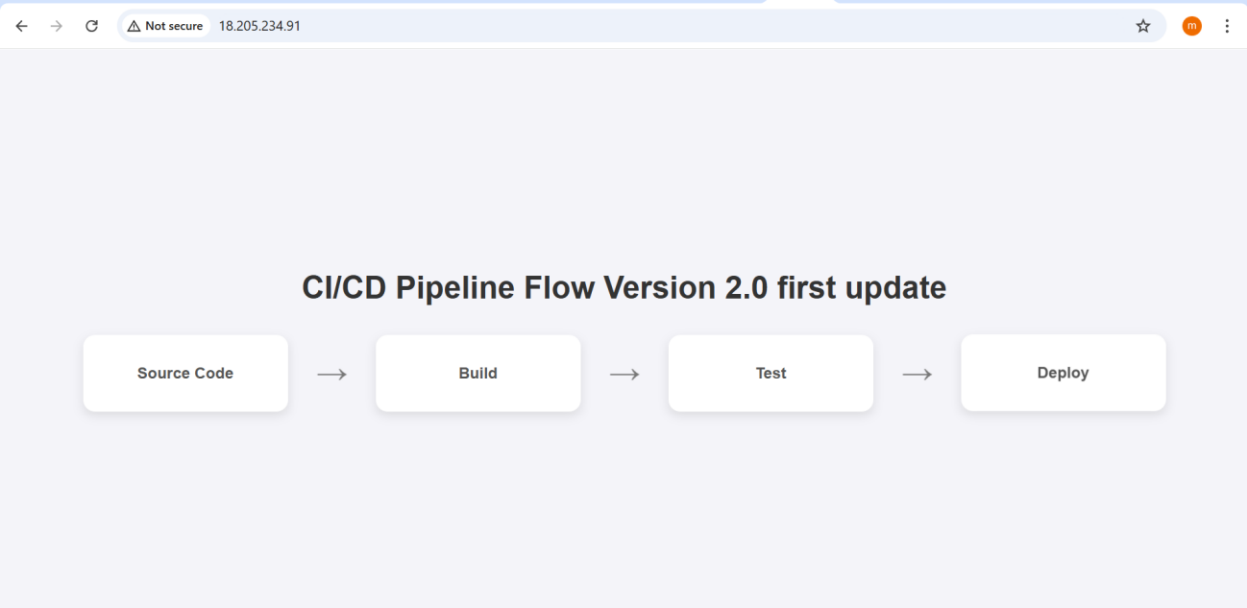
Learn more

Cancel Submit

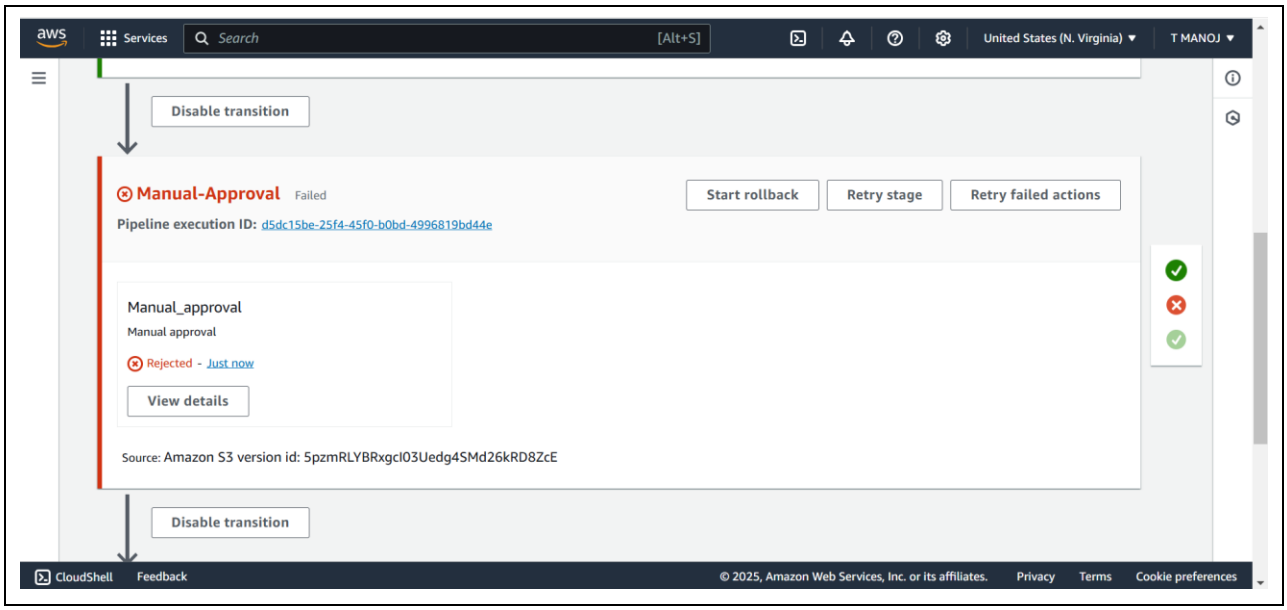


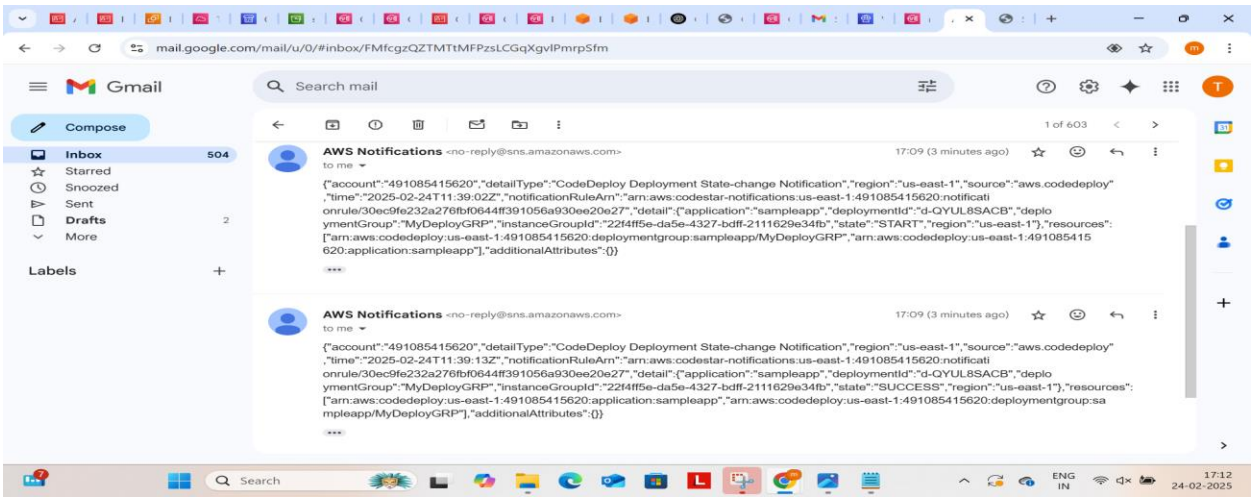

Gmail
Search mail
Compose
Inbox 504
Starred
Snoozed
Sent
Drafts 2
More
Labels
1 of 603
AWS Notification Message
Inbox
AWS Notifications <no-reply@sns.amazonaws.com>
16:58 (0 minutes ago)
to me
{"account":"491085415620","detailType":"CodeDeploy Deployment State-change Notification","region":"us-east-1","source":"aws.codedeploy","time":"2025-02-24T11:28:41Z","notificationRuleArn":"arn:aws:codestar-notifications:us-east-1:491085415620:notificationrule/30ec9fe232a276fbf0644ff391056a930ee20e27","detail":{"application":"sampleapp","deploymentId":"d-7CHKCABCB","deploymentGroup":"MyDeployGRP","instanceGroupId":"22f4ff5e-da5e-4327-bdff-2111629e34fb","state":"START","region":"us-east-1"},"resources":["arn:aws:codedeploy:us-east-1:491085415620:application:sampleapp","arn:aws:codedeploy:us-east-1:491085415620:deploymentgroup:sampleapp/MyDeployGRP"],"additionalAttributes":{}}
--
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-1:491085415620:notification:fe53efc9-9497-4675-a825-f5b52a8b834c&Endpoint=dreamers2k22@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


mail.google.com/mail/u/0/#inbox/FMfcgzQZTMTtMFPzsLCGqXgvlPmrpSfm
Gmail
Search mail
Compose
Inbox 504
Starred
Snoozed
Sent
Drafts 2
More
Labels
1 of 603
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
AWS Notifications <no-reply@sns.amazonaws.com>
16:58 (1 minute ago)
to me
{"account":"491085415620","detailType":"CodeDeploy Deployment State-change Notification","region":"us-east-1","source":"aws.codedeploy","time":"2025-02-24T11:28:52Z","notificationRuleArn":"arn:aws:codestar-notifications:us-east-1:491085415620:notificationrule/30ec9fe232a276fbf0644ff391056a930ee20e27","detail":{"application":"sampleapp","deploymentId":"d-7CHKCABCB","deploymentGroup":"MyDeployGRP","instanceGroupId":"22f4ff5e-da5e-4327-bdff-2111629e34fb","state":"SUCCESS","region":"us-east-1"},"resources":["arn:aws:codedeploy:us-east-1:491085415620:deploymentgroup:sampleapp/MyDeployGRP","arn:aws:codedeploy:us-east-1:491085415620:application:sampleapp"],"additionalAttributes":{}}
Reply
Forward



Rejecting the Manual Approval :





Advantages

- Automation: Reduces manual tasks through automated pipelines.
- Faster Deployments: Enables rapid and frequent releases.
- Reliability: Minimizes human errors and ensures consistent deployments.
- Scalability: Easily adaptable for scaling applications across multiple environments.
- Monitoring & Alerts: Integrated CloudWatch and SNS for real-time monitoring and notifications.
- Cost-Effective: Pay-as-you-go AWS services optimize resource usage and costs.
- Improved Collaboration: Facilitates seamless integration between development and operations teams.

Conclusion

Implementing a CD/CP pipeline on AWS optimizes the deployment process by automating critical stages, ensuring faster time-to-market, and enhancing application reliability. The integration of AWS services like CodeDeploy, CodePipeline, and CloudWatch provides robust monitoring and efficient management. With proper configuration, this pipeline becomes a vital asset in any DevOps workflow, promoting agility, consistency, and scalability in application deployment cycles.