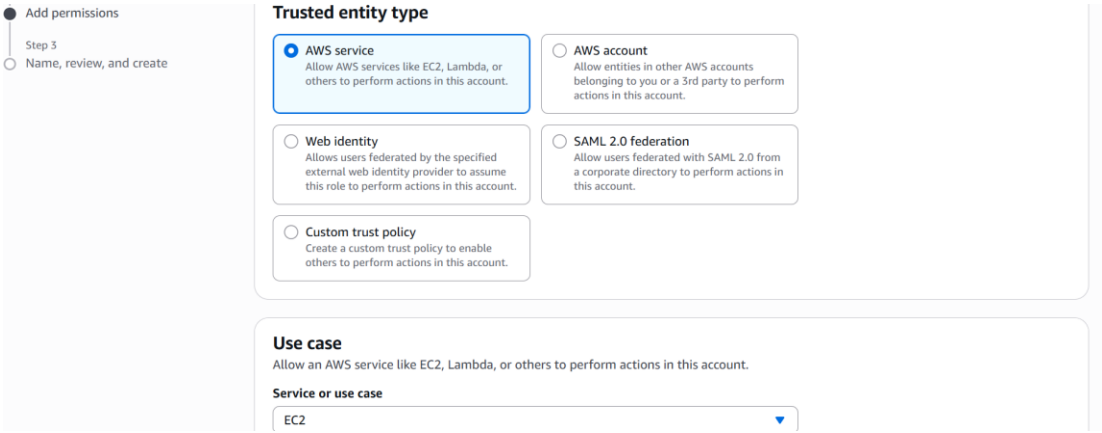
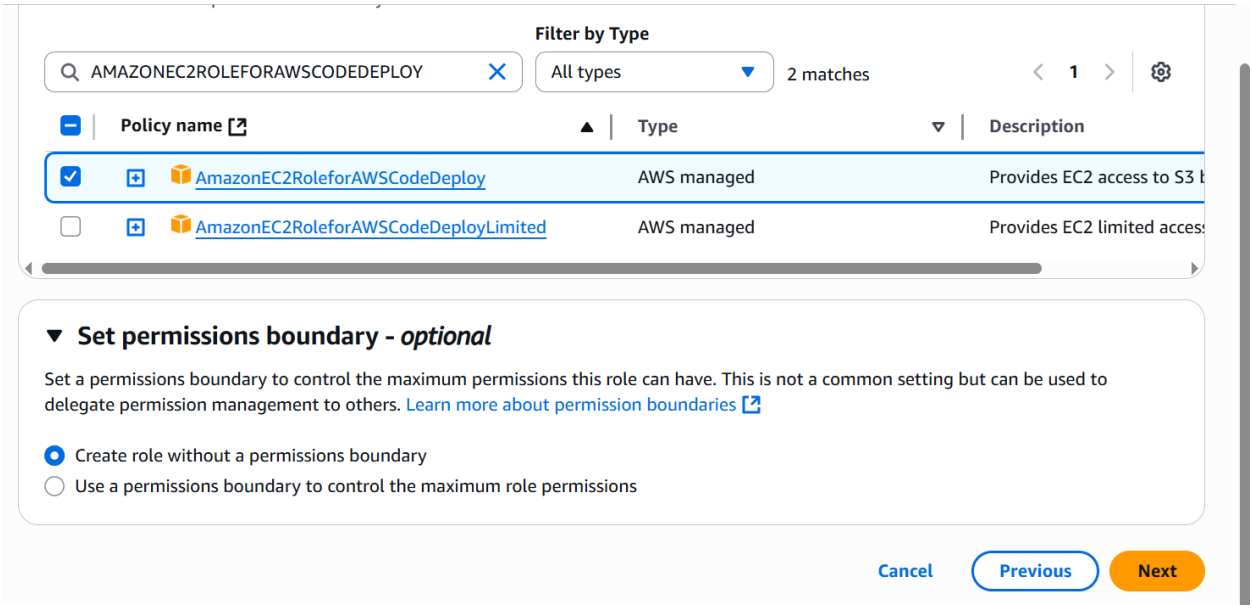
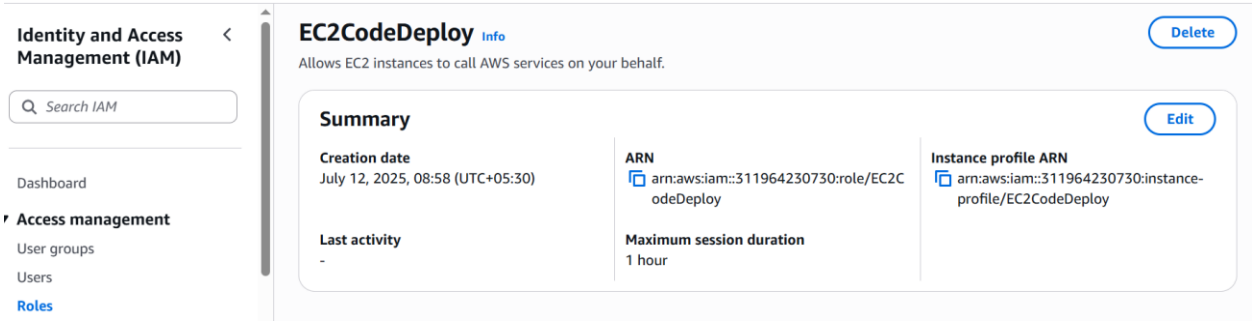
Task-1:

- IAM Role
  - Access IAM on Console and select roles from the left navigation pane.
  - Create two roles.
  - Select create role and leave the trusted entity type as AWS service.

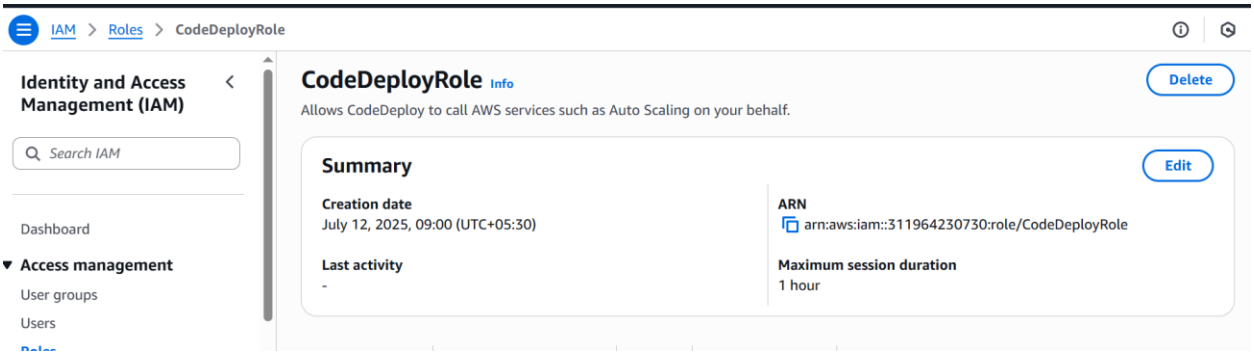


  - Under use case select EC2 followed by plain EC2 and hit next.
  - In add permissions page in the search box enter Code deploy and select AMAZONEC2ROLEFORAWSCODEDEPLOY and hit next.
  - Give the name to the role as EC2codedeploy.

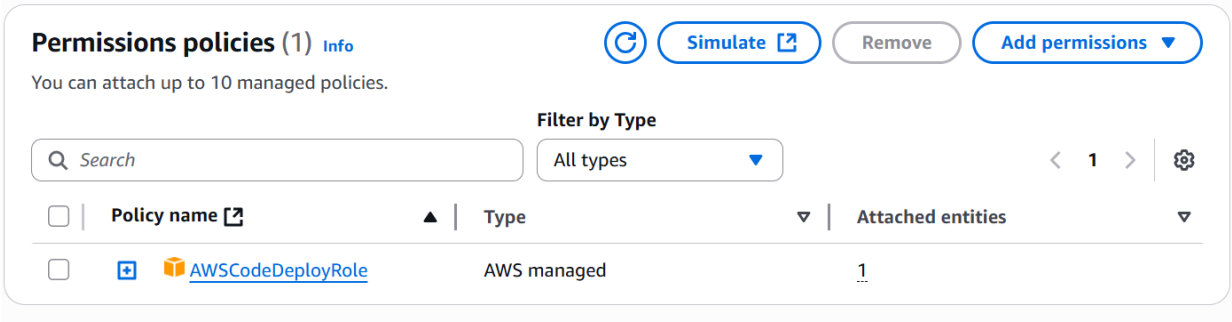


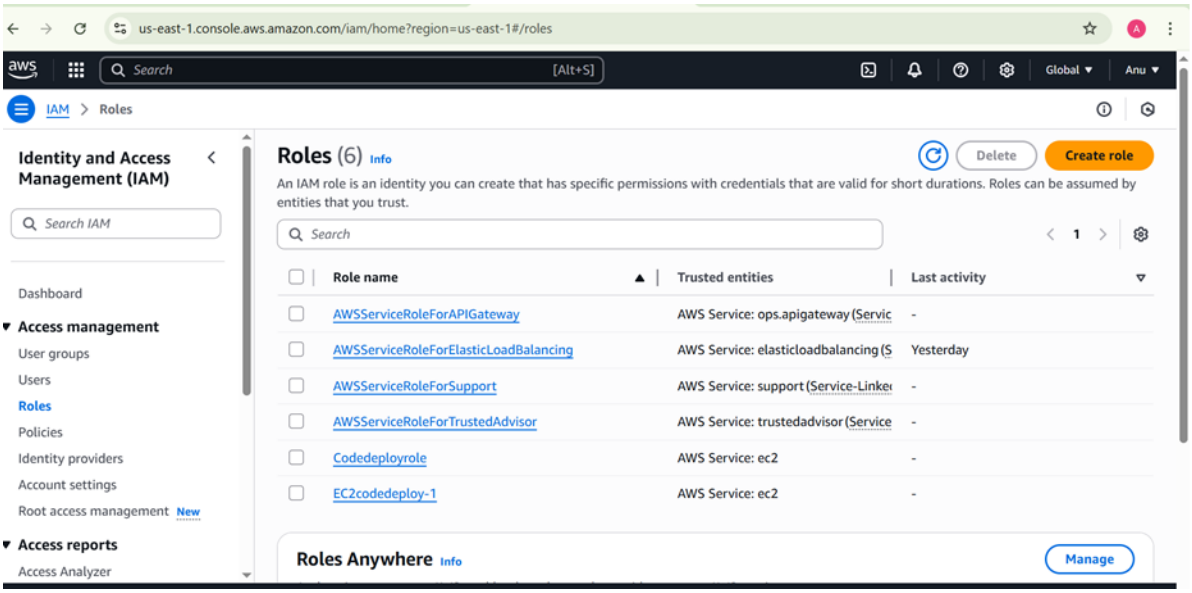


- Click on create role.
- Create one more role for code deployment purpose.
- Follow the same process to create a role under use cases for other AWS services from dropdown select Codedeploy and select plain codedeploy below and hit next.



- You can see AWSCodedeploy role.
- Click on next.
- Give role name as Codedeployrole. Click on create role.
- You can notice two roles have been created.

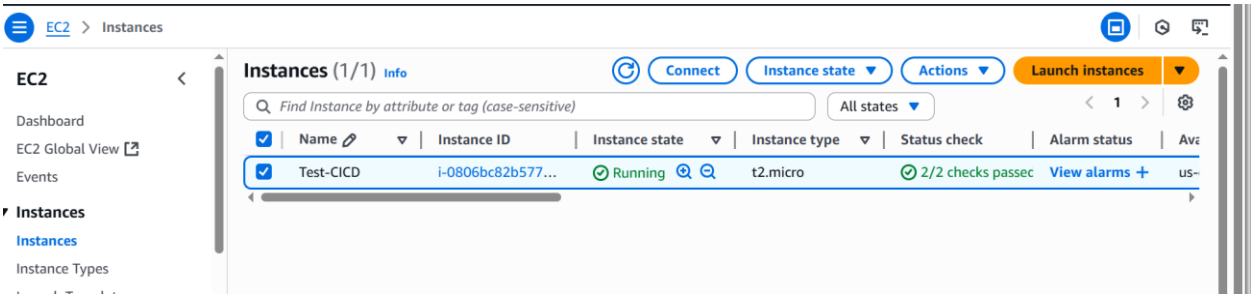




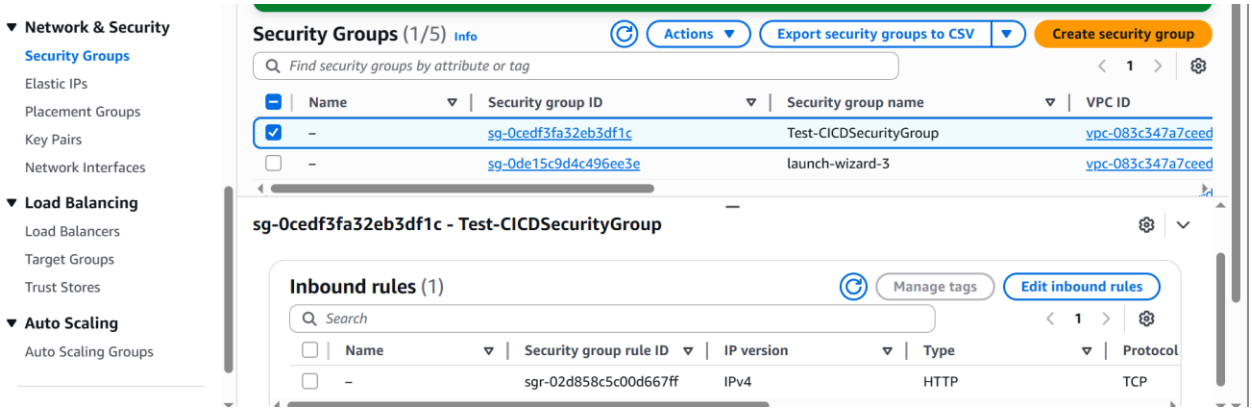
Task-2:

- EC2 Instance Creation

  - Access EC2 on Console and select launch instances.
  - Select the region as N.Virginia.
  - Give the instance name as Test-CICD.
  - Select OS as Linux.
  - Instance type as t2.micro
  - Create a key pair.
  - In the network settings tab select edit and under create security group give the name as TestCICD security group
  - Description as Security group for my Test-CICD
  - Remove the existing SSH security group rule.
  - Expand advanced details and in IAM instance profile field select EC2codedeploy role. Scroll down to user data box and copy the code from Git repository and paste it in the user data box.
  - Click on launch instances.



- Access EC2 on console and the select the instance created by you.
- From the left navigation pane under networking select security groups and select the security group created by you.
- Select inbound rules followed by edit inbound rules.
- Click on add rule and take the source as HTTP and Anywhere IPV4 and save rules.



Task-3:

- Code Pipeline

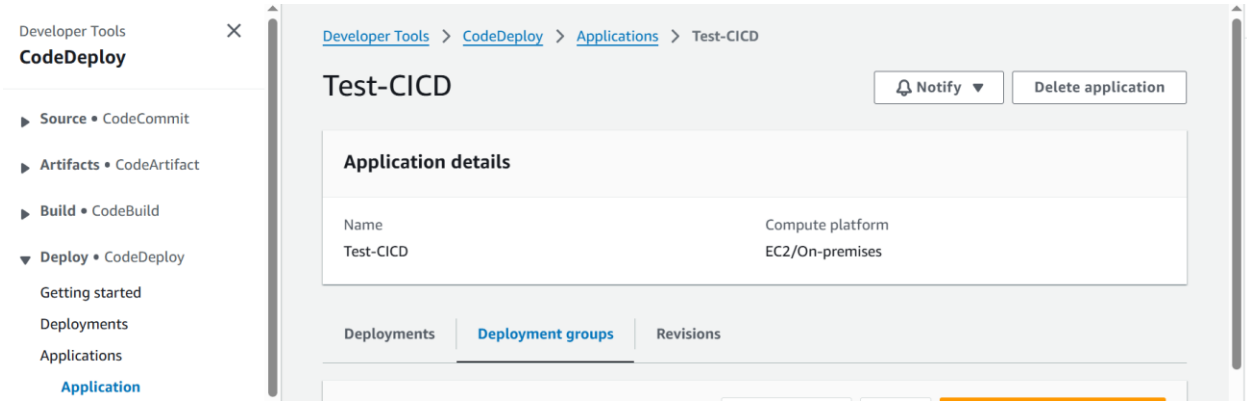
  - Access code pipeline on Console and select code deploy from left navigation pane.
  - Expand deploy and select applications.
  - Click on create application. Give the name as Test-CICD.
  - Compute platform should be EC2/on premises and click on create application.
  - Application created and you can see create deployment group.
  - Click on create deployment group and give deployment group name as Test-CICD-DP.
  - Service role select as codedeployrole.
  - Under environment configuration, select amazon EC2 instances.
  - Under amazon EC2 instances for key select name and value as Test-CICD.
  - Under deployment settings, select codedeploydefault.allatonce .
  - Load balancer, disable loadbalancing by unselecting the check box
  - Create deployment group

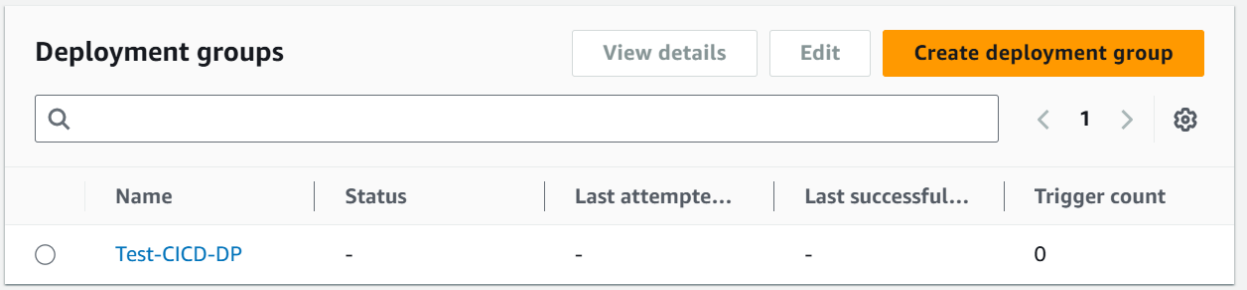
Task-4:

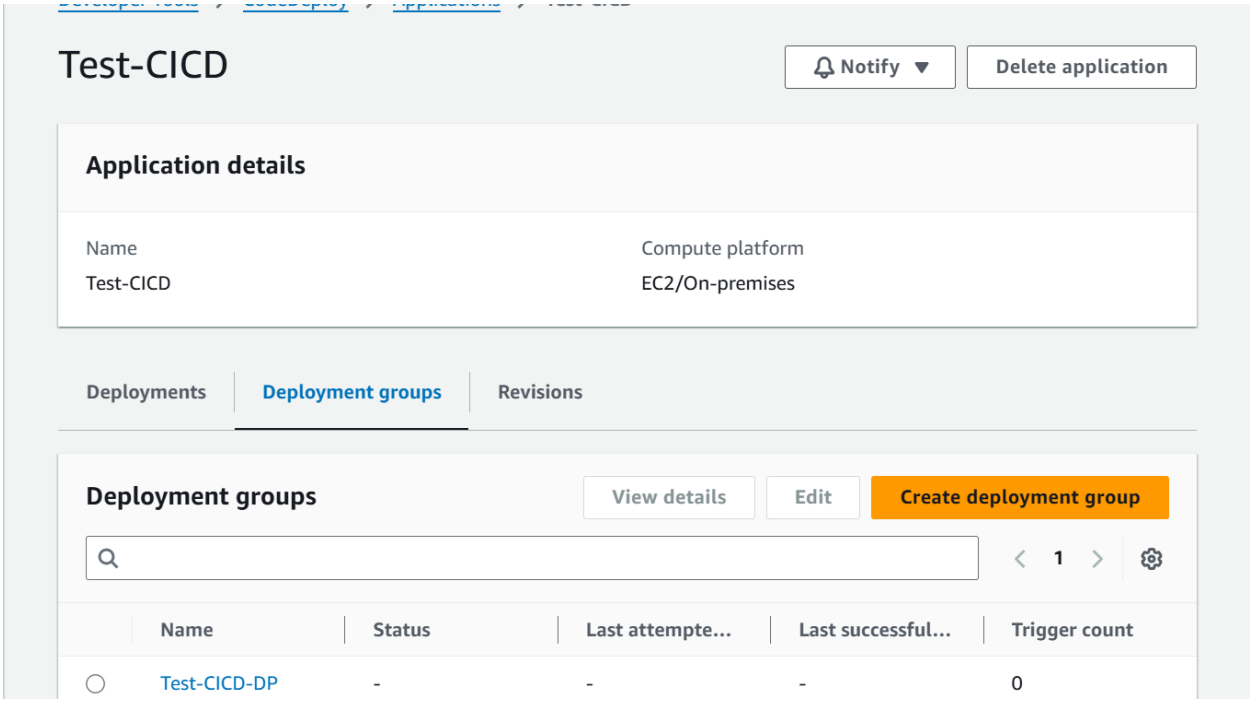
- Code Deploy

  - From left navigation pane of code pipeline expand code pipeline and click on pipelines.
  - Click on create pipeline.
  - Select build custom pipeline and hit next.
  - Give pipeline name as Test-CICD-pipeline
  - Leave the service role as New service role

- Expand advanced settings, observe default location as S3 bucket and click on next
- It asks for source provider
- Select GitHub(Via GitHub App) from drop down and click on connect to GitHub.
- Give connection name as Test-CICD-Git.
- Click on connect to GitHub.
- In GitHub apps field it says authorize AWS to GitHub. Select it.
- In connect to GitHub page, give connection name as Test-CICD-Git and click on install a newapp
- Select install and authorize and enter GitHub password
- Click on connect. It directs to pipeline page and you can see your git has been synced.
- Under repository name select the repository created by you from clicking in the field.
- Output field take default as code pipelinedefault
- Click on next
- In build provider as we are using AWS cofde deploy, just skip the build stage at the end of the page
- Also skip the test stage.
- In deploy state page, select deploy provider as AWS codedeploy.
- Region N.Virginia
- In application name field select Test-CICD
- In deployment group field select Test-CICD-DP
- Click on next
- Review all pipeline settings and click on create a pipeline.
- You can see deploy in progress.







Task-5:

- Access EC2 on console and the select the instance created by you.
- From the left navigation pane under networking select security groups and select the security group created by you.
- Select inbound rules followed by edit inbound rules
- Click on add rule and take the source as HTTP and Anywhere IPV4 and save rules.
- Go to the instance oage select the instance created by you and copy the public IPV4 address and paste in a new tab to see the message from the git code which you uploaded on EC2.