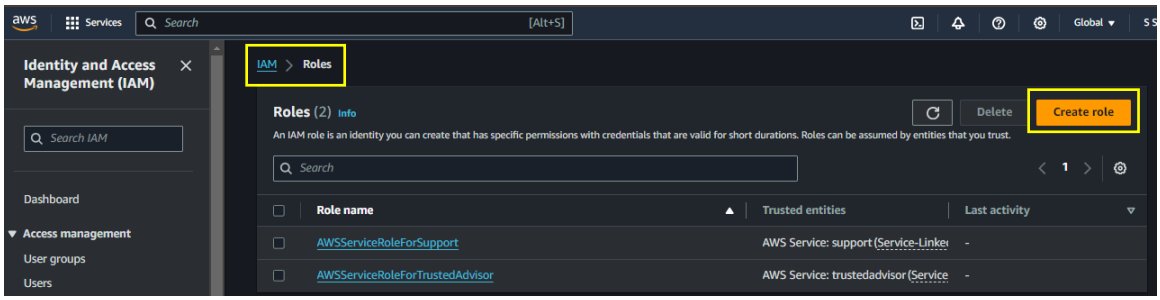
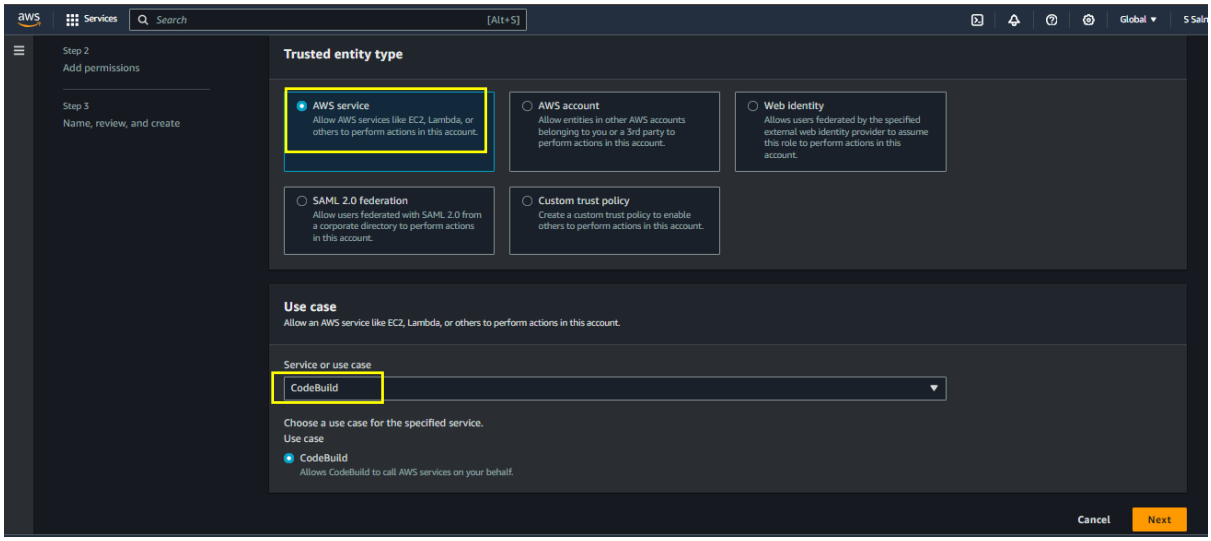
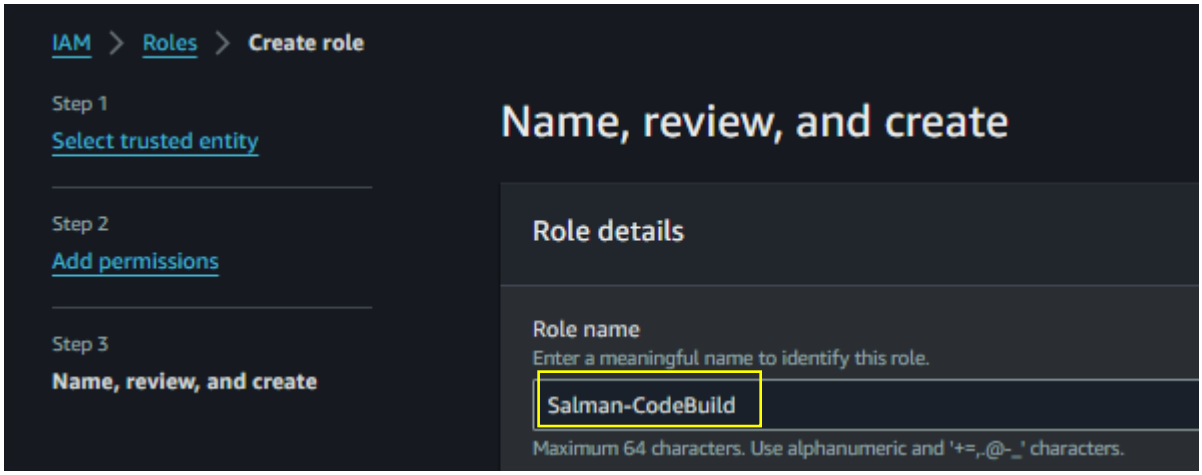
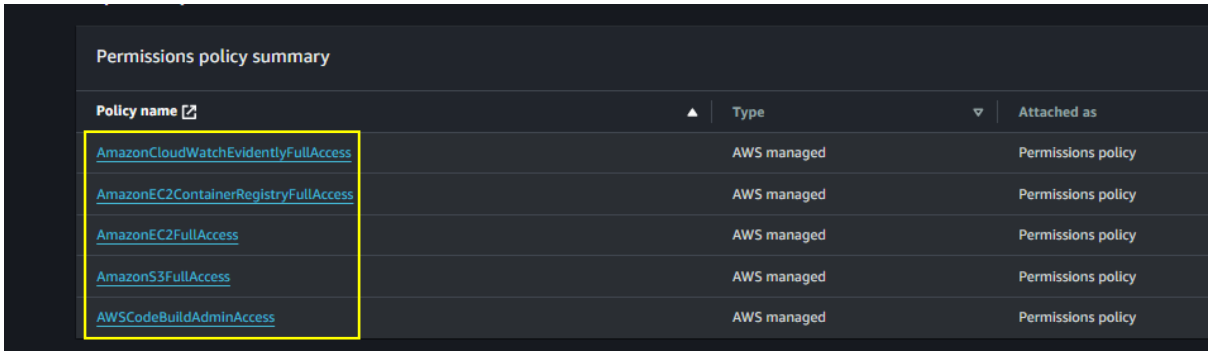
Code Build Demo:

Go to IAM

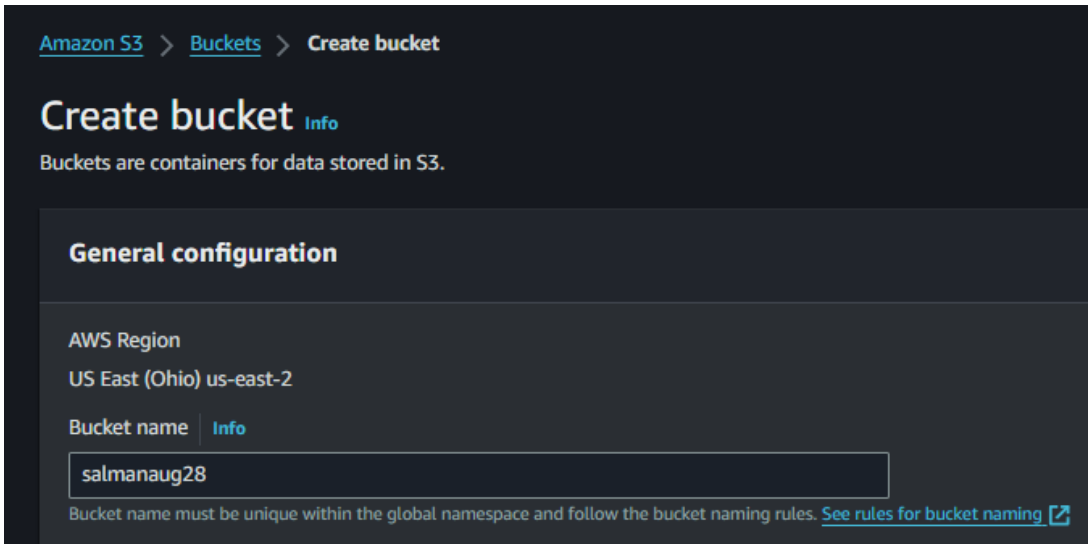


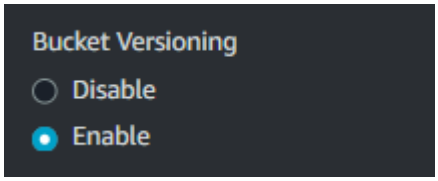


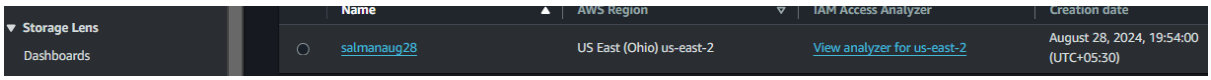




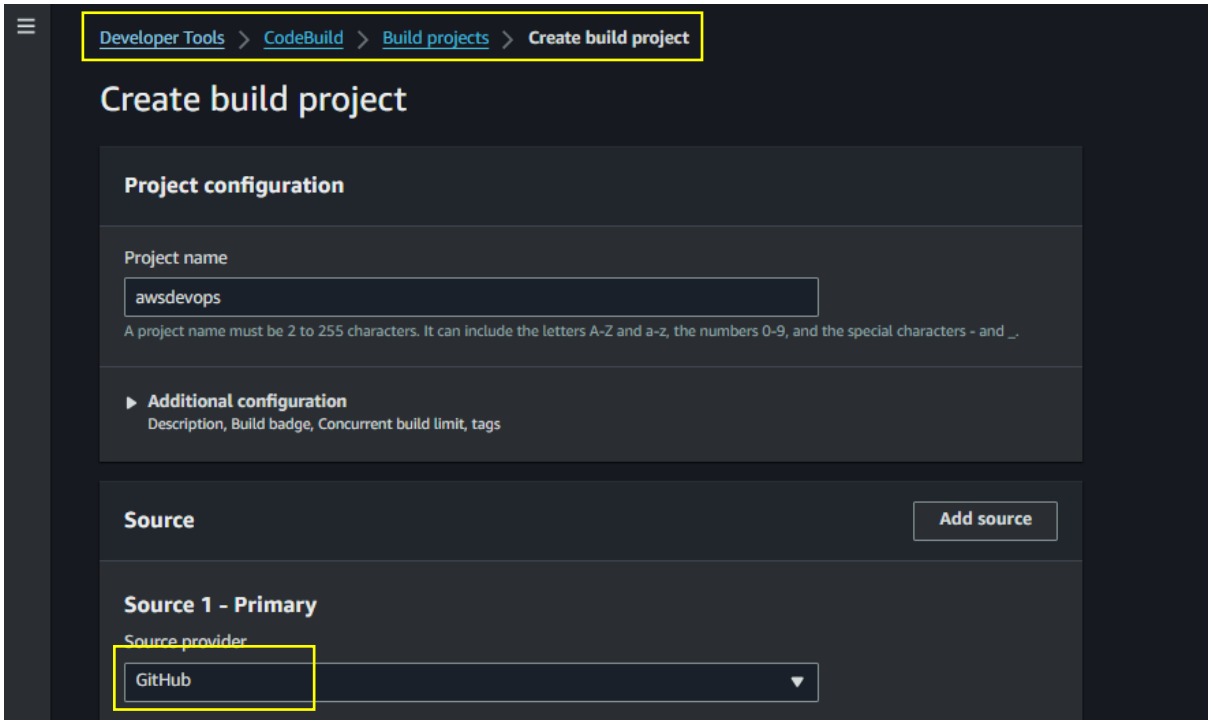
Create a Bucket for After Build the Job it will be store all the build artifacts in bucket, we can go through with s3, RDS..... so I chosen s3 bucket



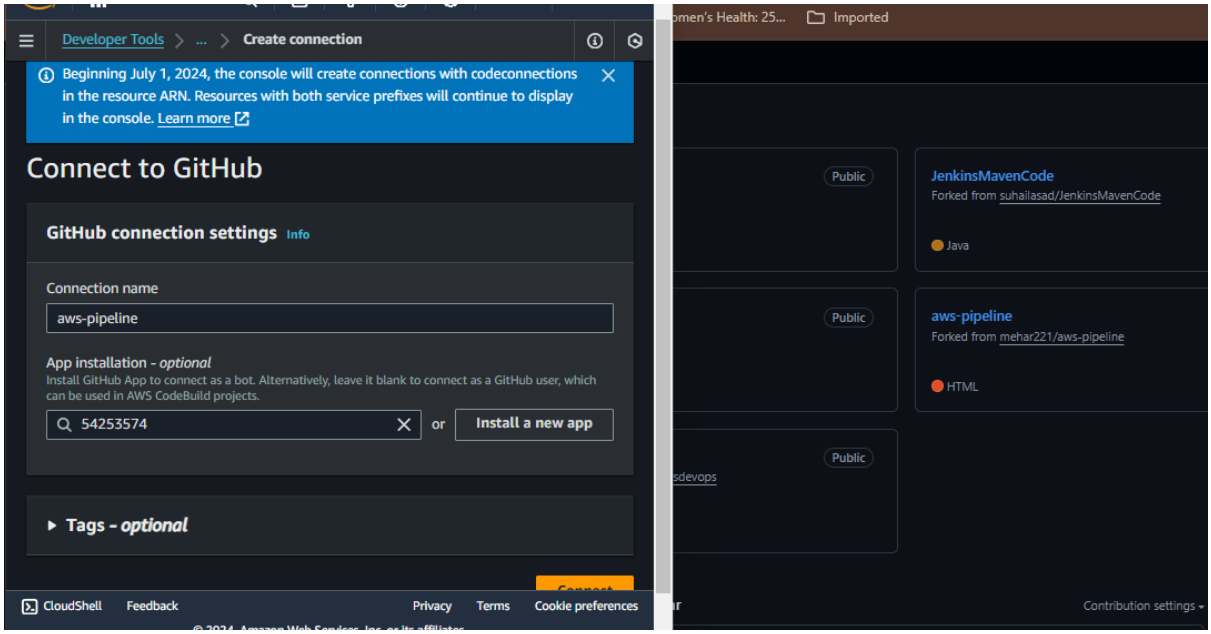




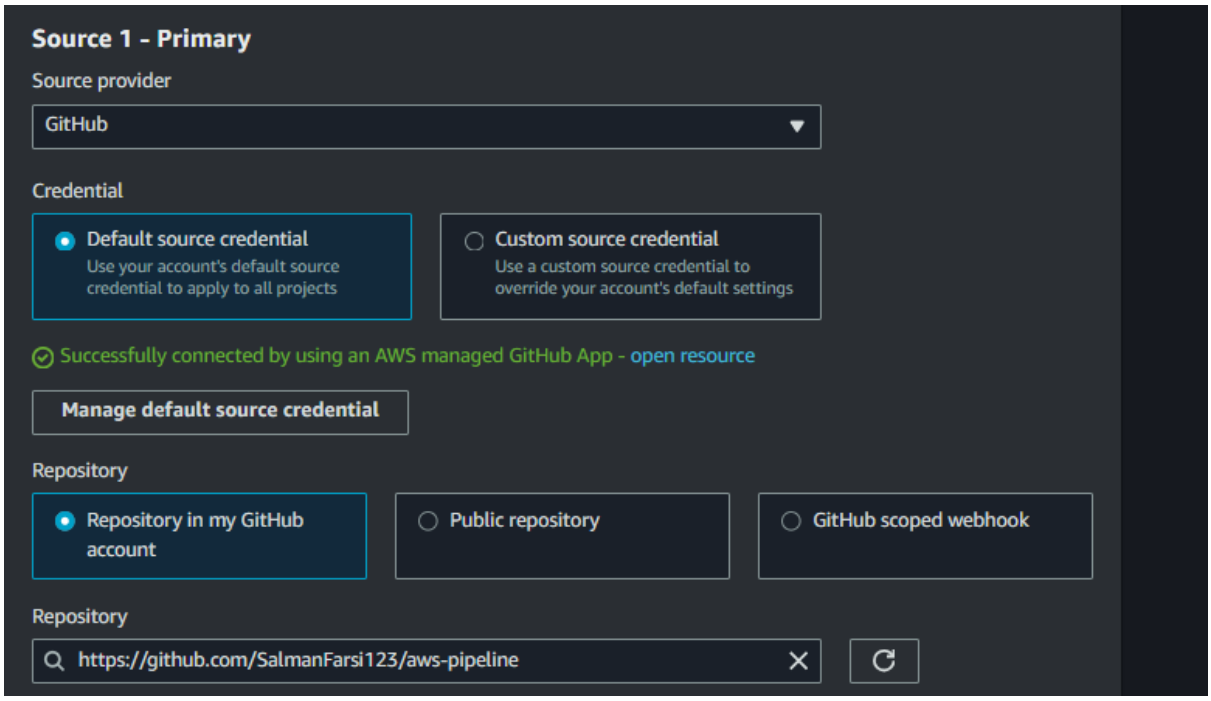
Now Create a CodeBuild Setup and after Select Source Provider and Click Manage



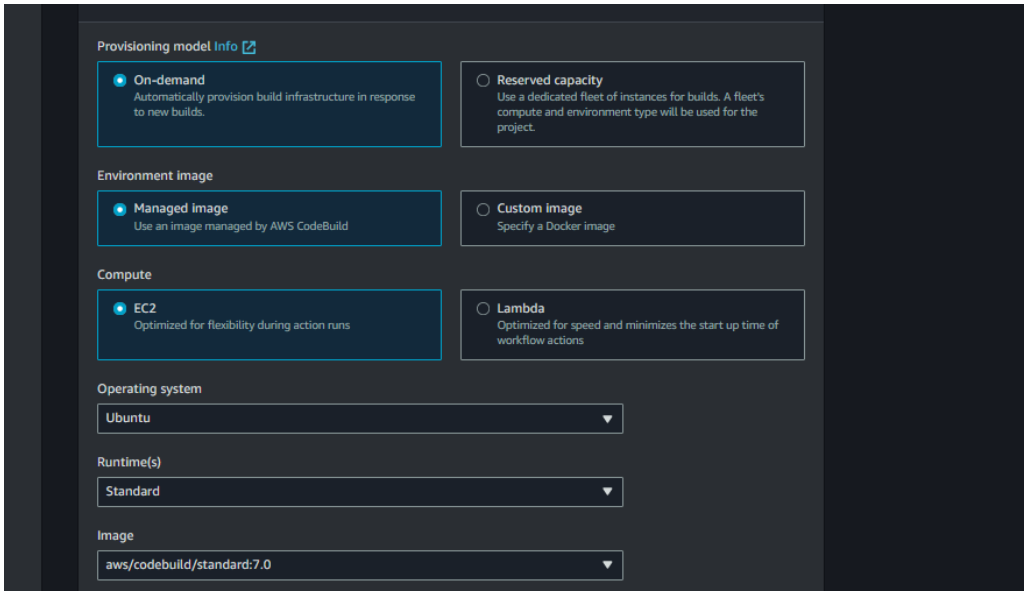
Give the Connection name and login into your git hub and connect it.



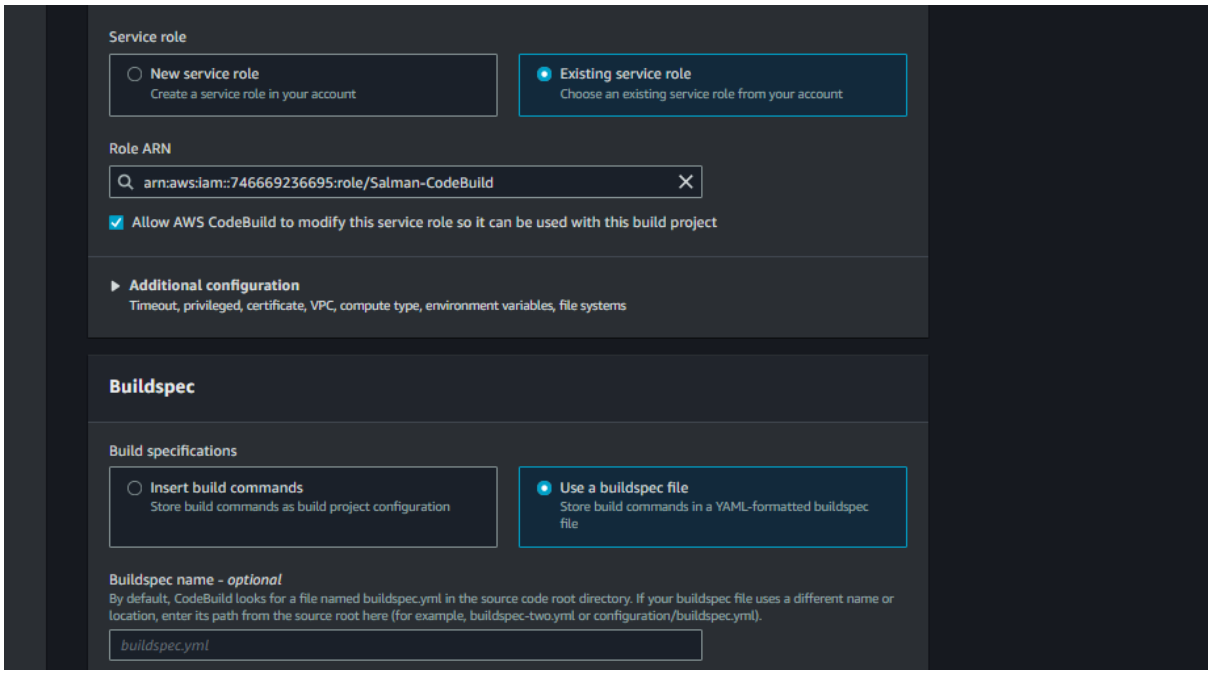
Successfully Connected, In the Repository, I chosen aws-pipeline repo and remove .git



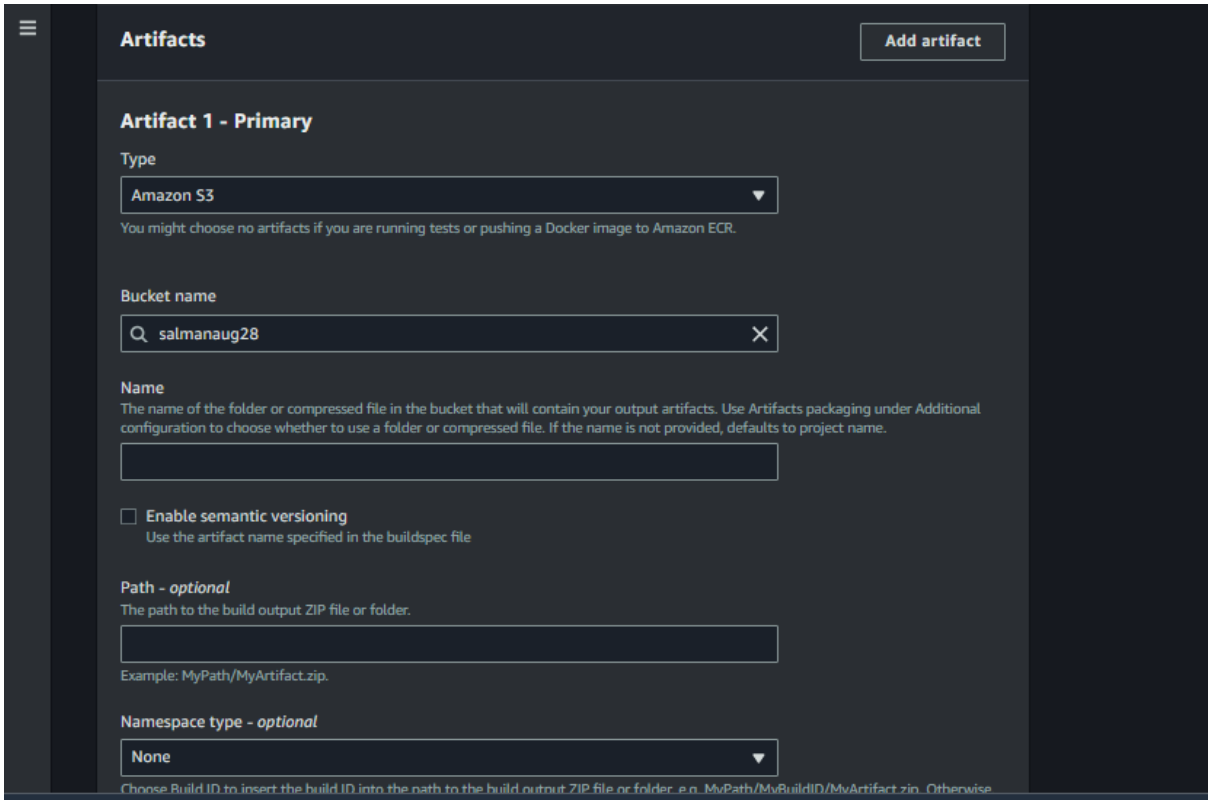
Select Operation system as ubuntu



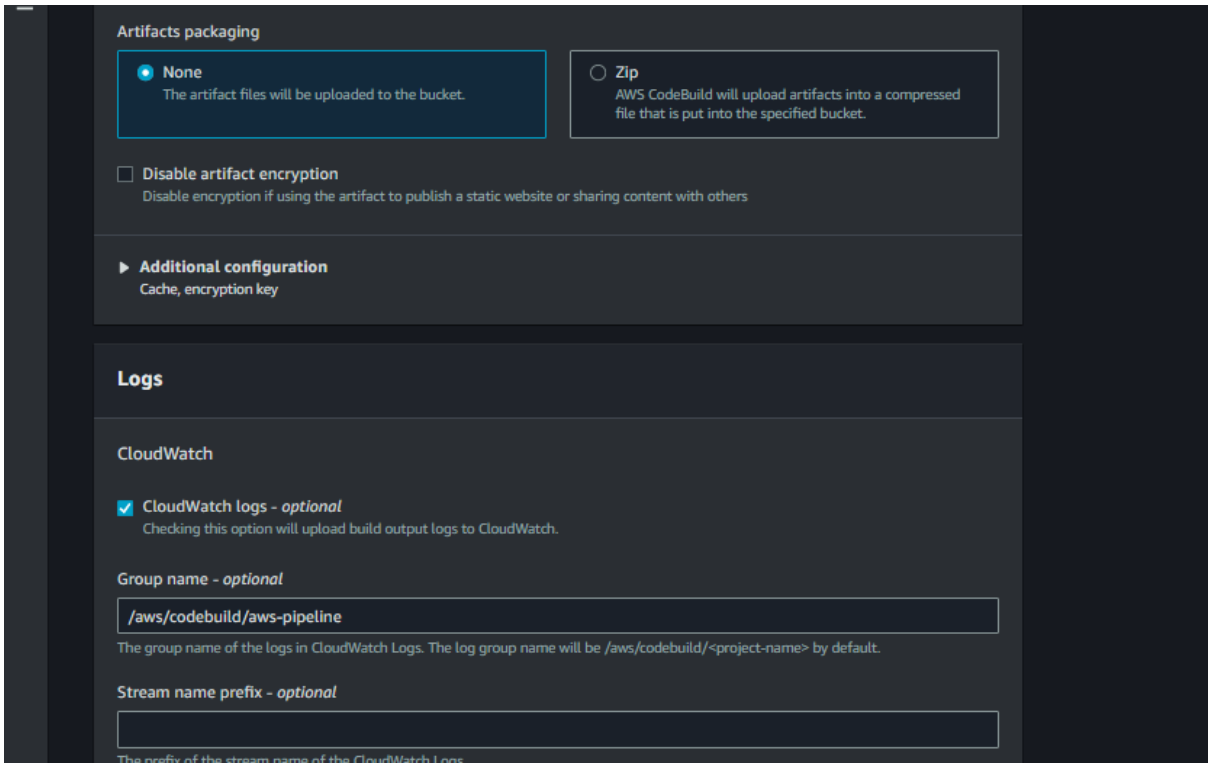
Previously We Created IAM Role for CodeBuild Select it



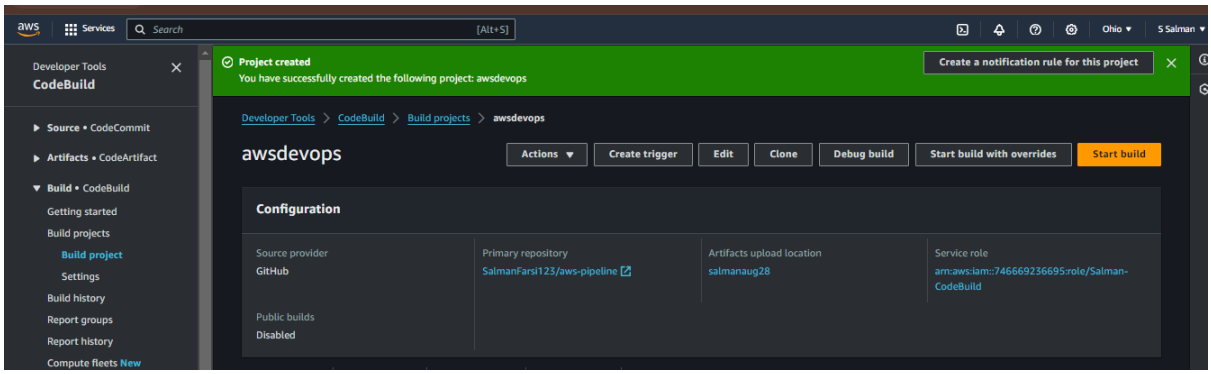
And Artifacts select s3 and select created Bucket



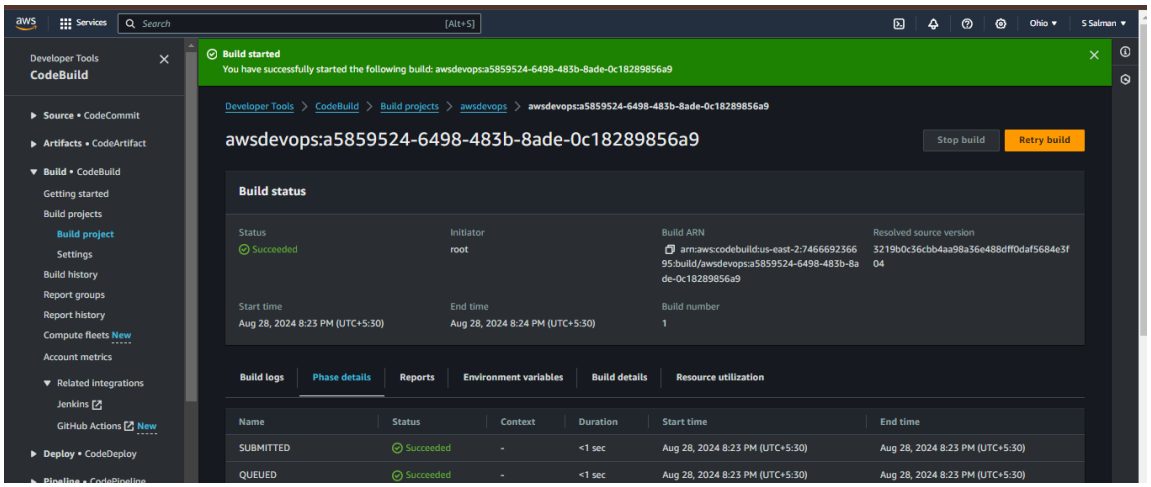
In Logs /aws/codebuild/aws-pipeline, yellow mark name keep it as default and blue mark is your repo name which we are building now, and hit the create button



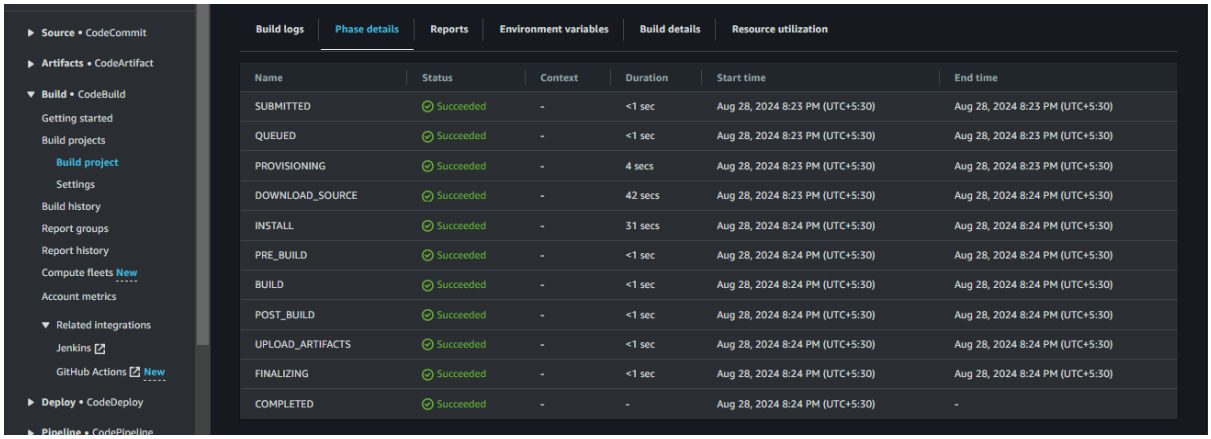
Now Start Build



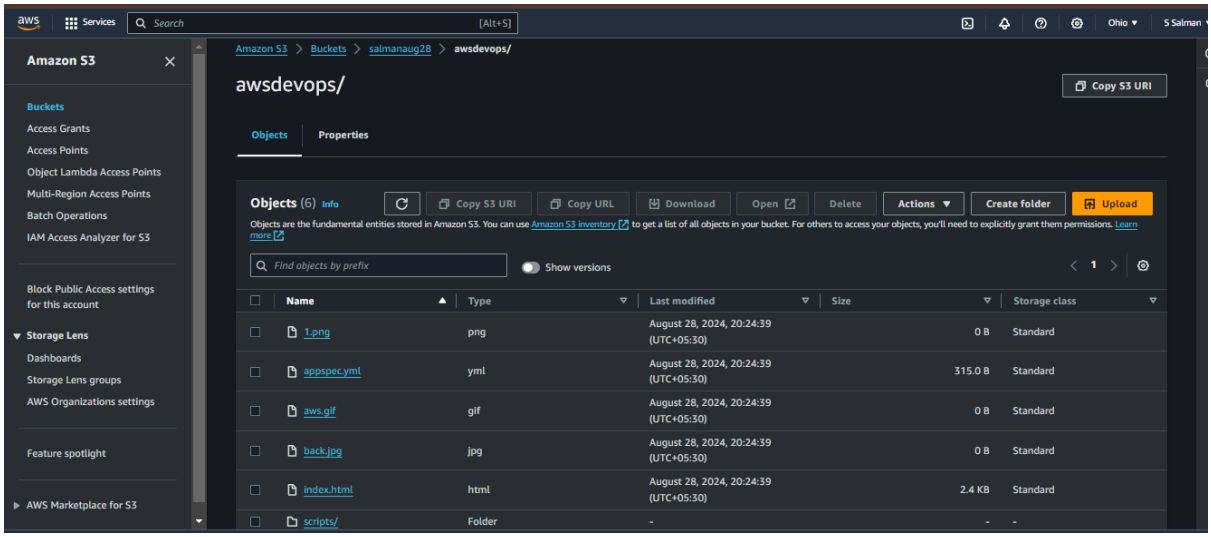
The Build was Succeeded



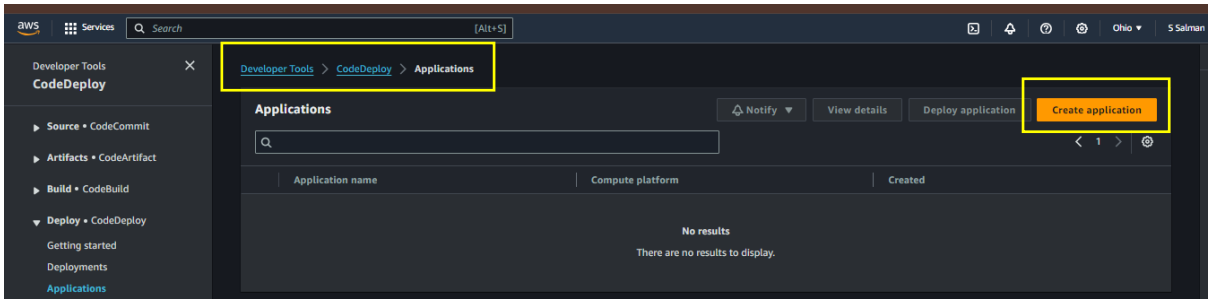
Phase Details

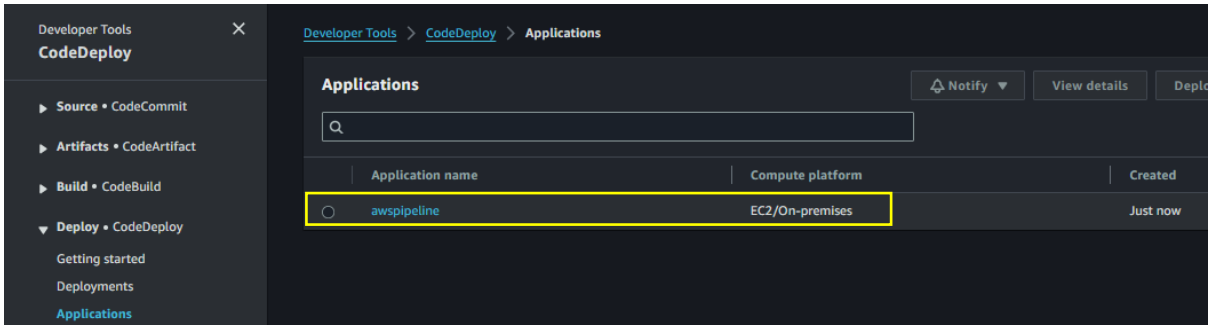


Artifacts get stored in s3 bucket, this is codebuild demo.

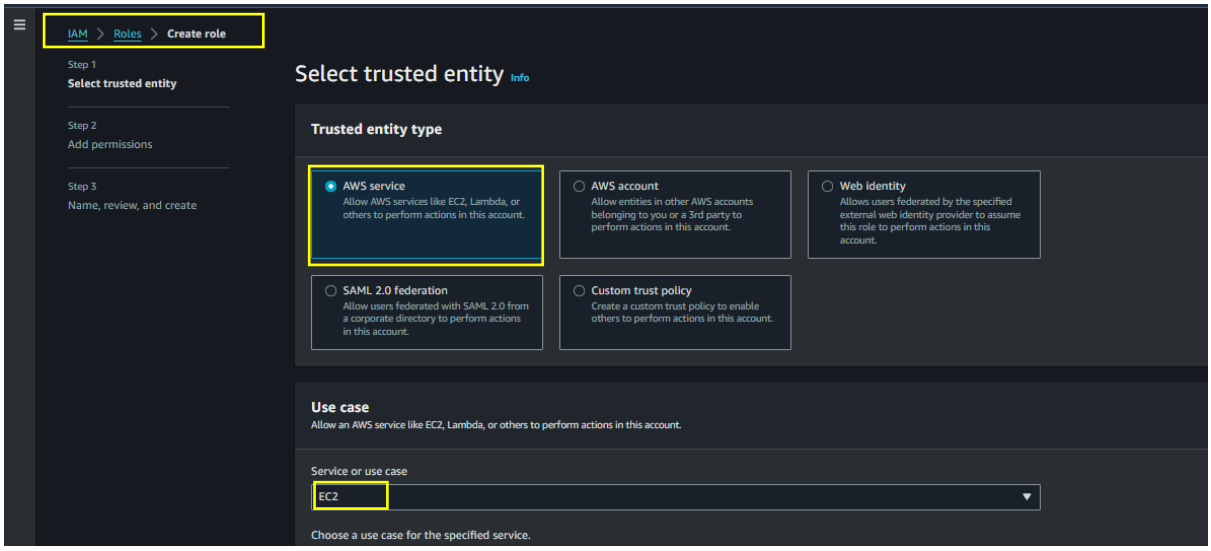


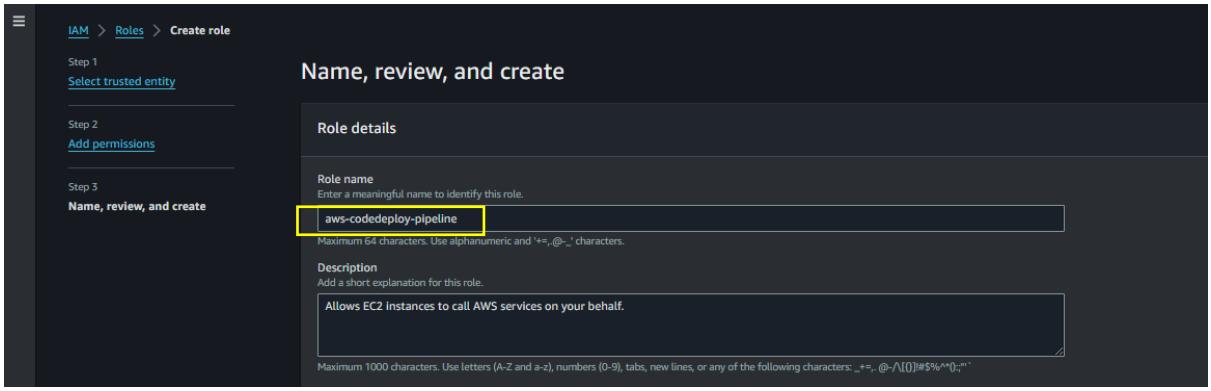
CODE Deploy Demo:



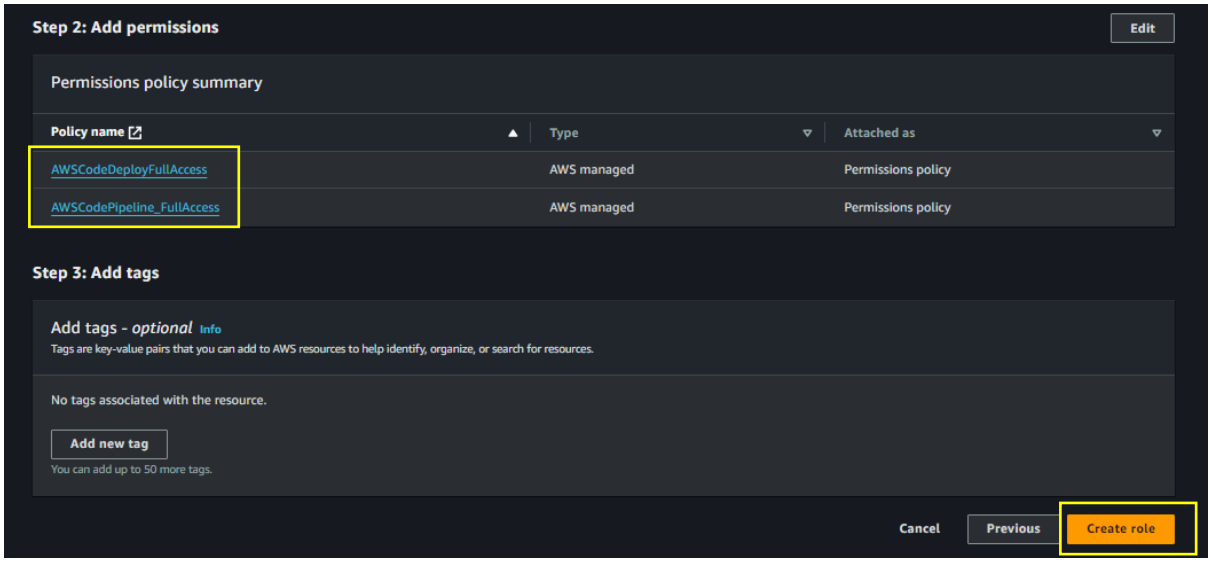


Now Creating IAM Role for Code Deploy Demo, Providing Permissions for EC2 Instances

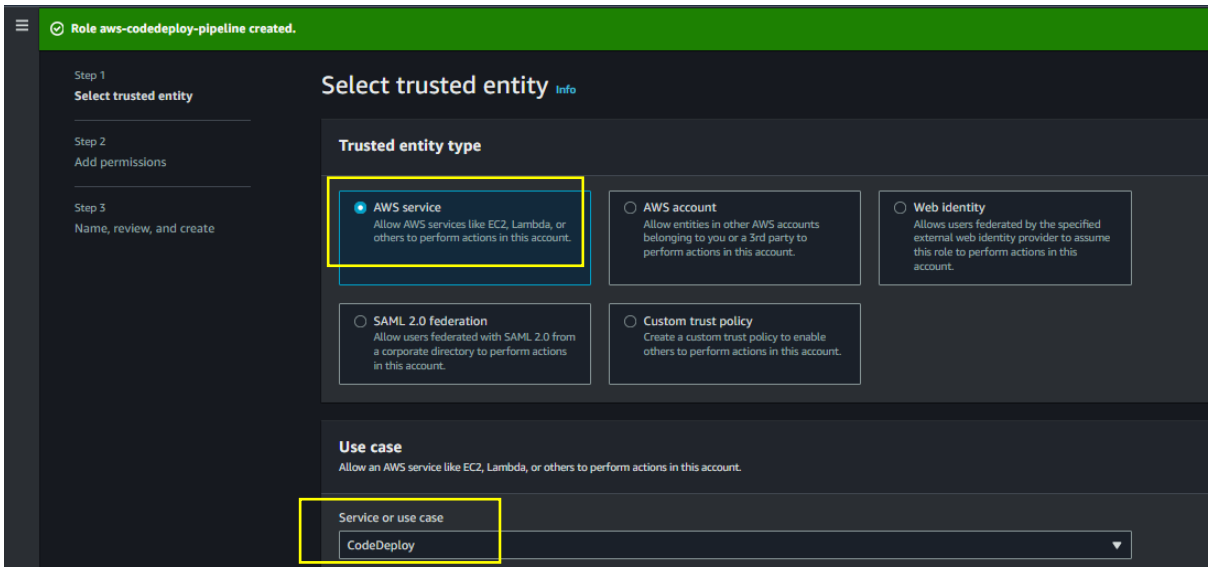




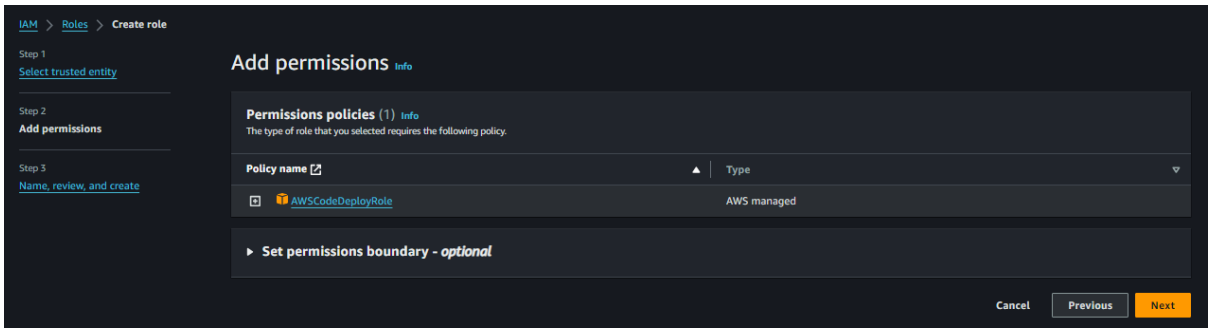
Here We are allowing CodeDeploy and CodePipeline Full Access to EC2 Instances



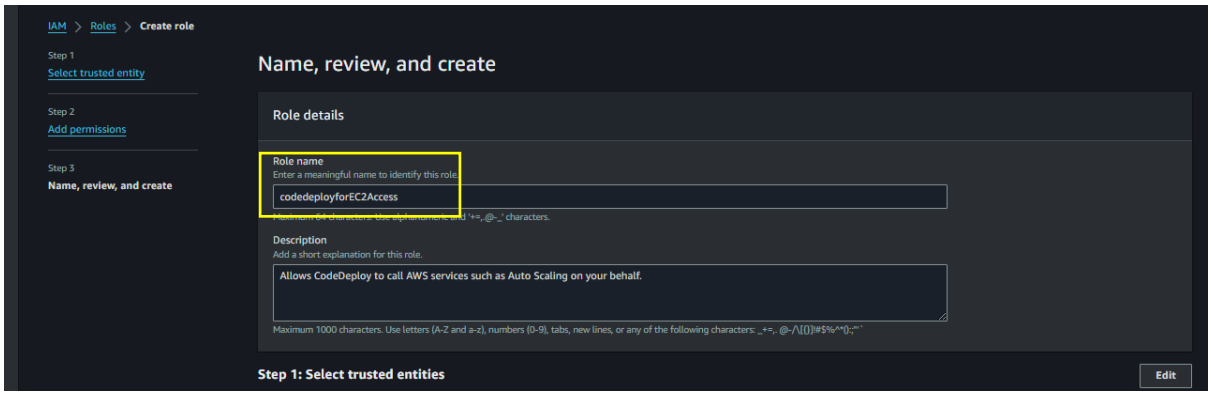
Create Another Role for CodeDeploy



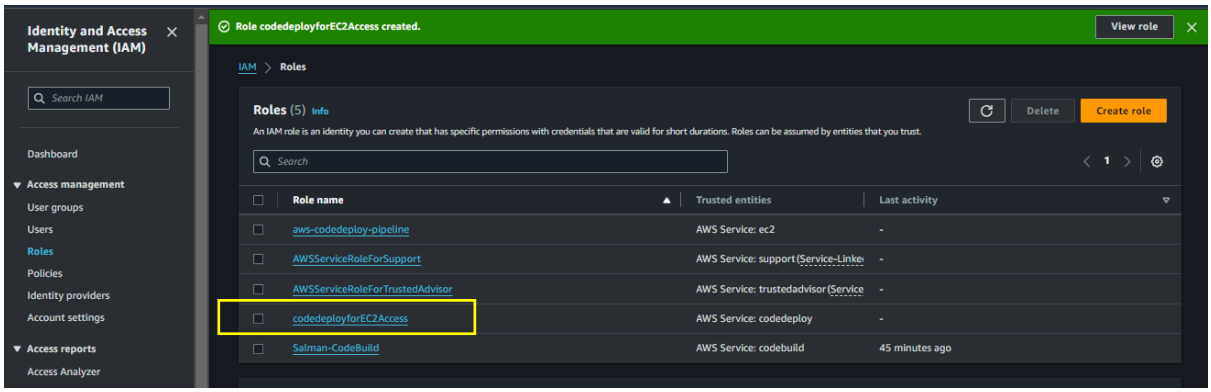
Here We are not able to add permission Just click Next

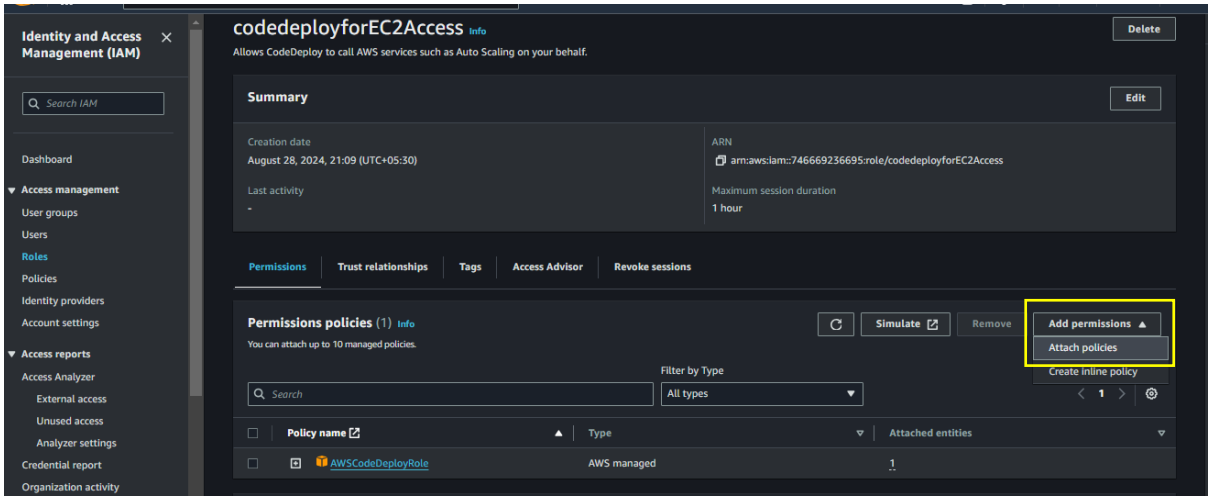


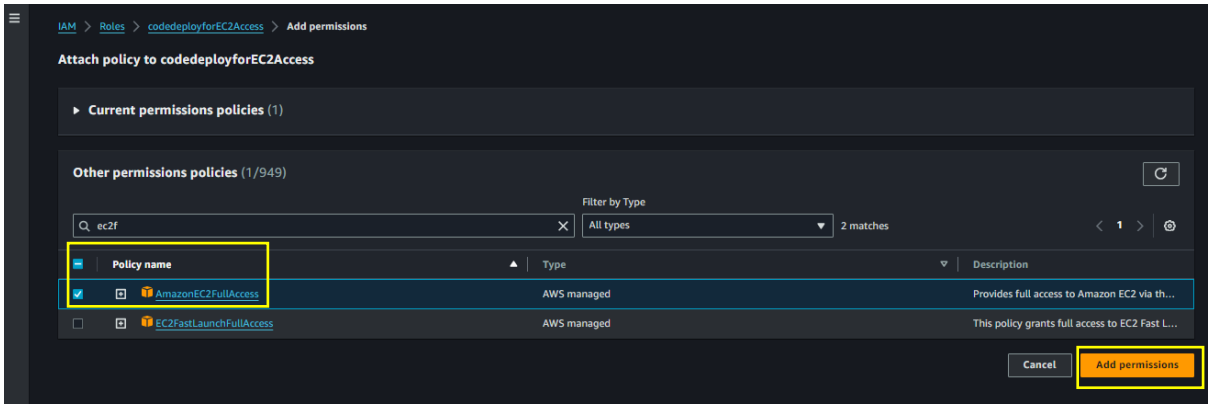
Give Role Name and Create

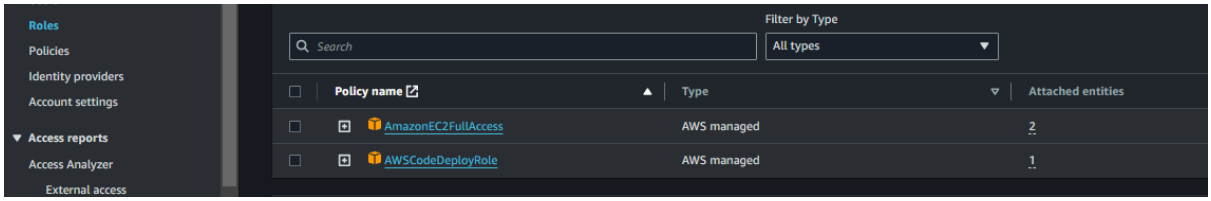


And Now Open That Codedeployroleforec2access

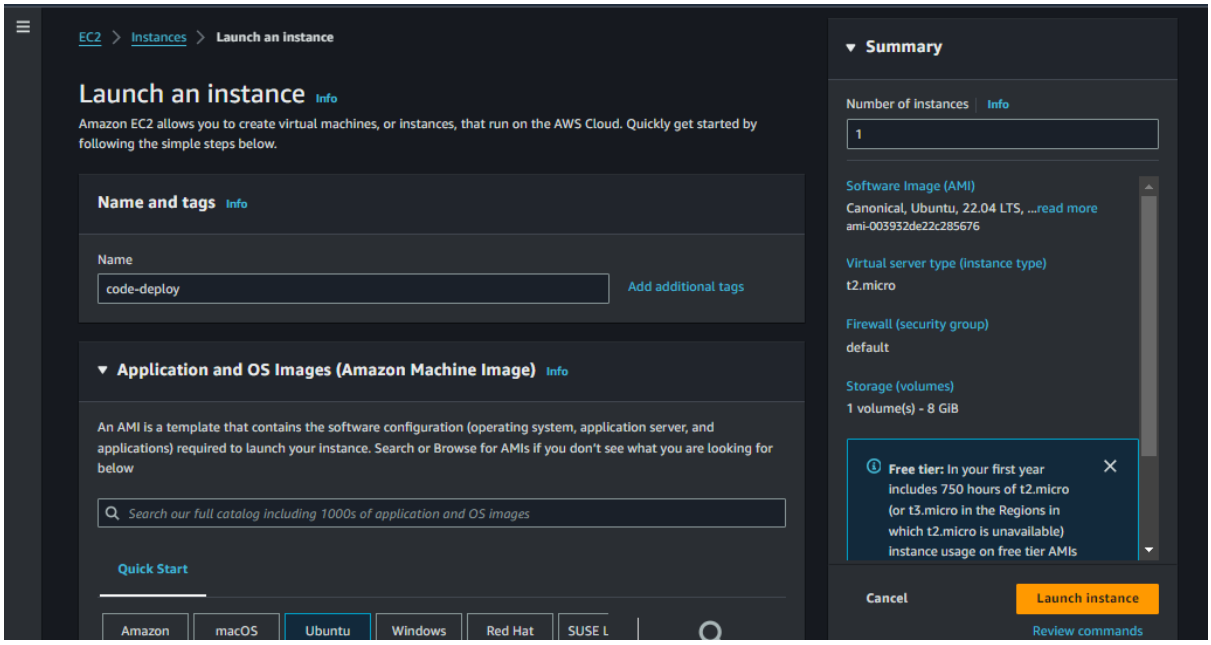




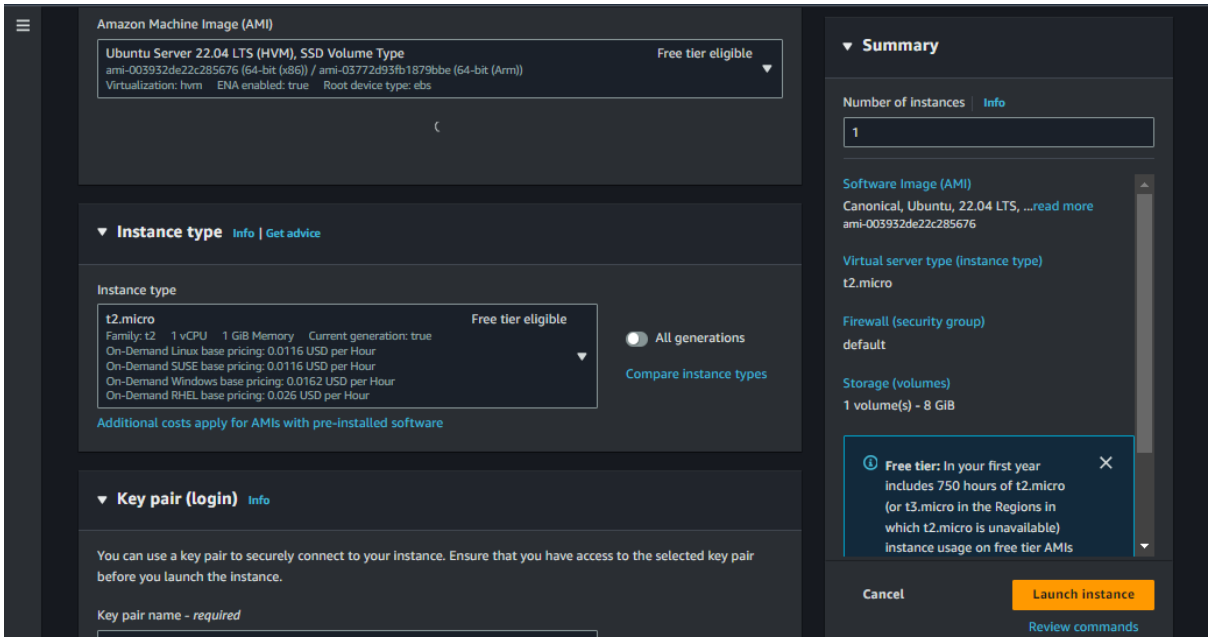




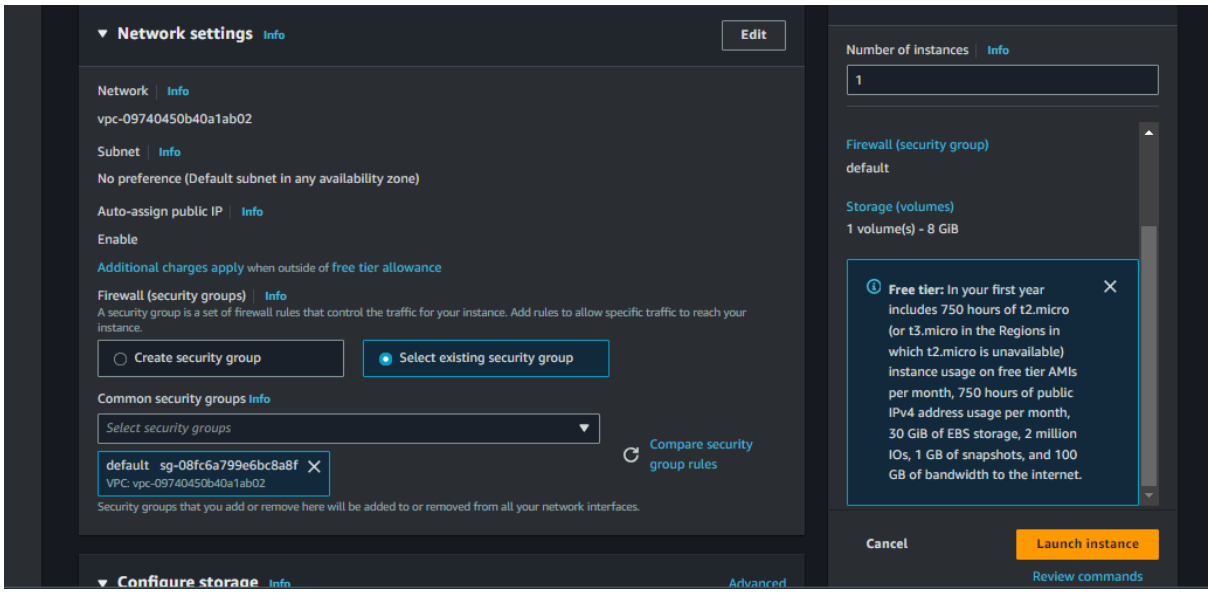
Now Launch t2.medium Instances for Code Deploy, Our Application will be going to deploy/run on this instances select ubuntu 22.04, do not select 20.04, I here launched with t2 micro but upcoming screenshot I will switch into t2.medium



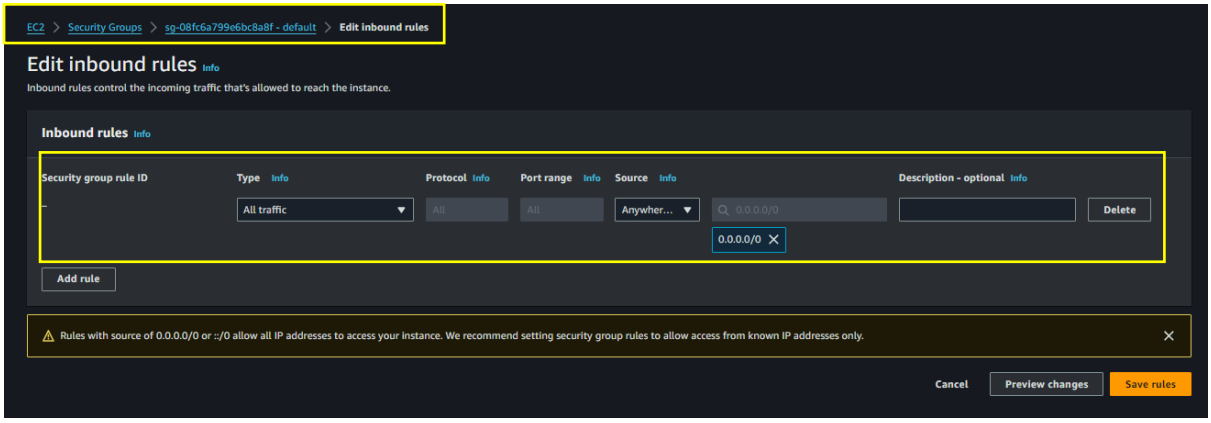
We know that how to launch instances



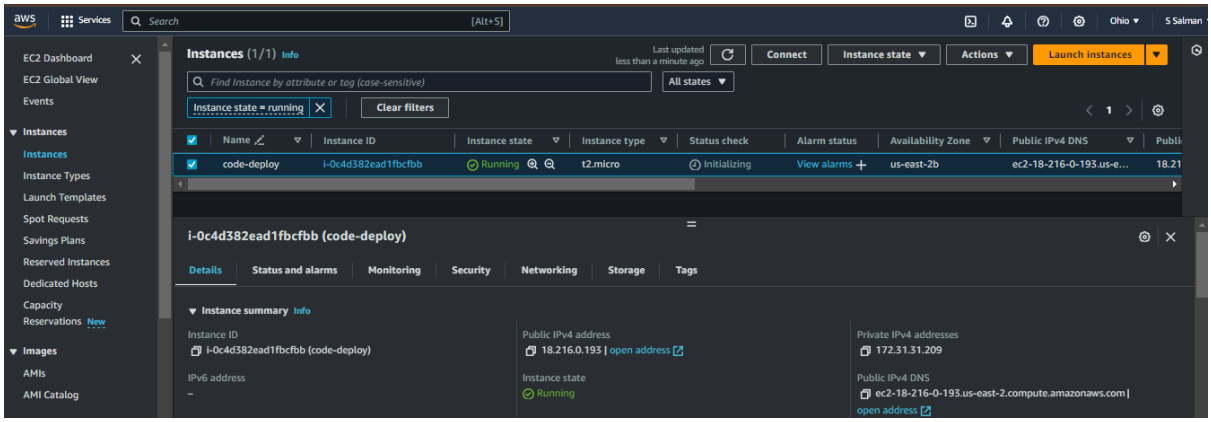
Everything Default I am using here and Launch it



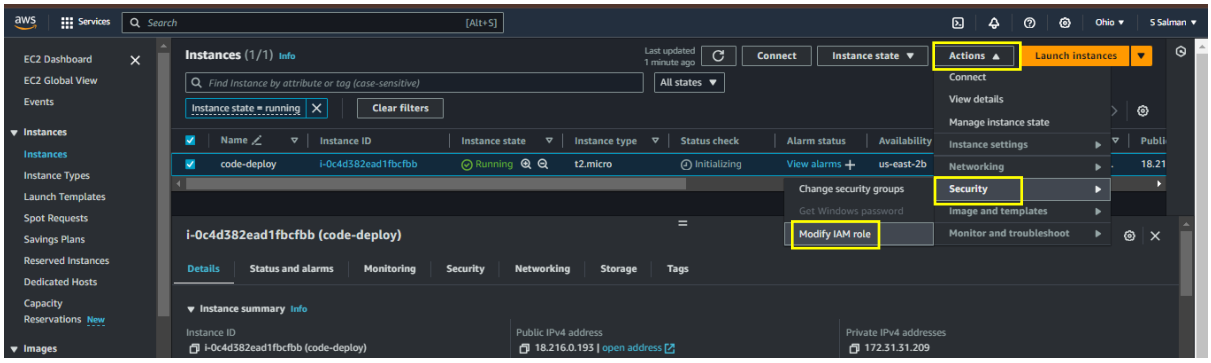
Go To Security Groups, delete and add the rule, Allow all traffic and anywhere and save rules



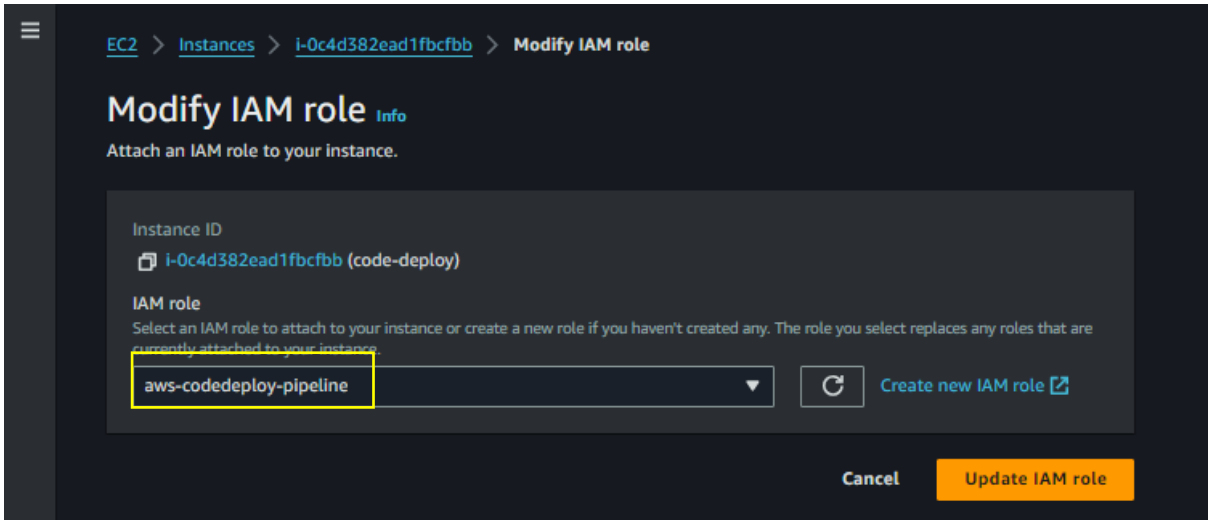
This our instances running successfully



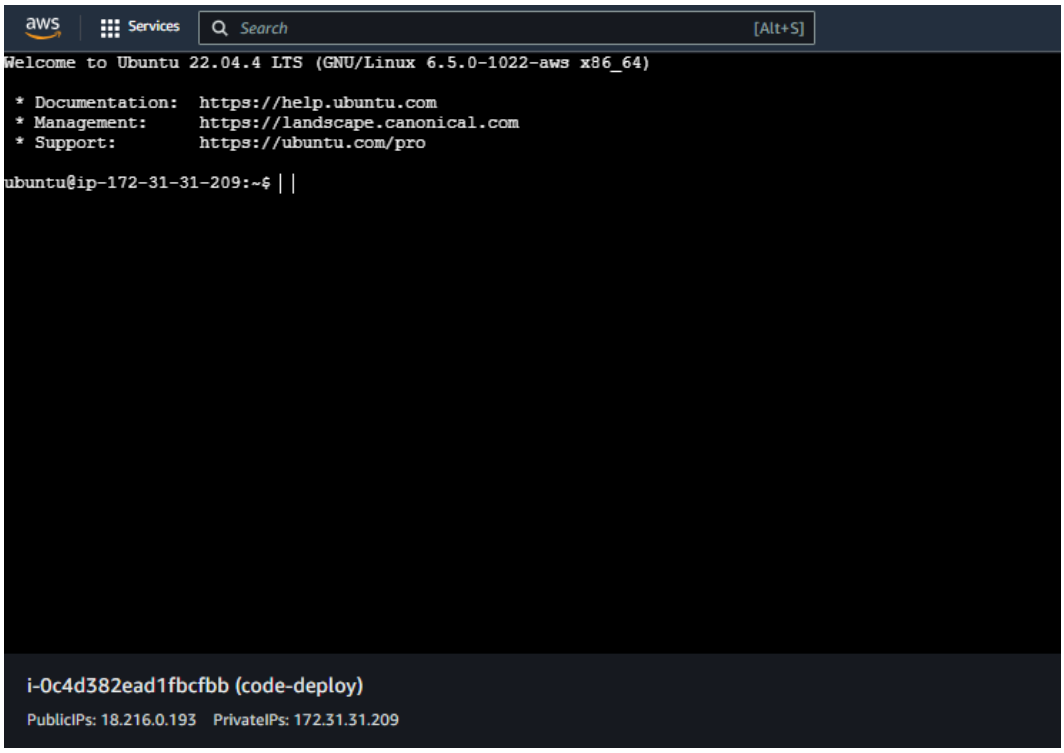
Now here I am Attaching IAM role for Ec2 instances



Update IAM Role



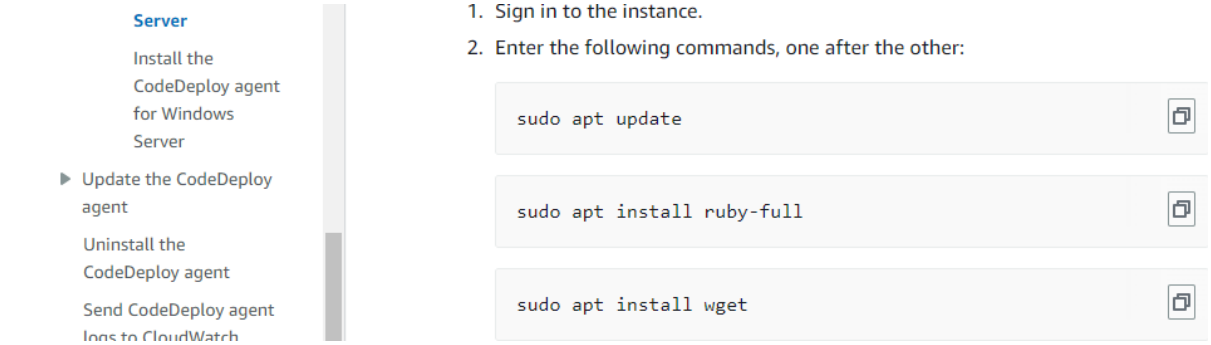
Connect Instances



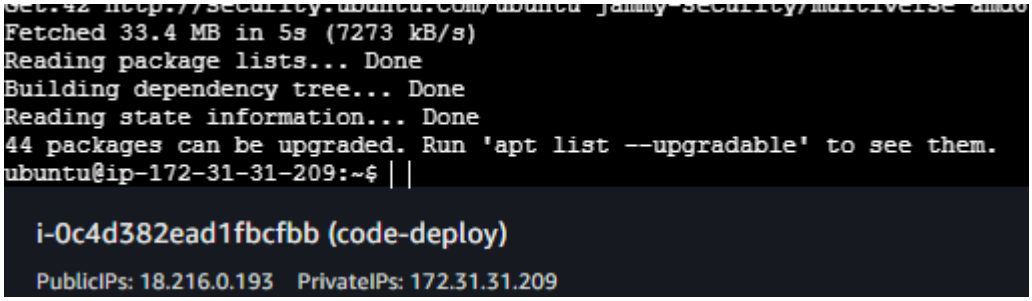
Go Through below the link to follow the instructions to install codedeploy agent into instances

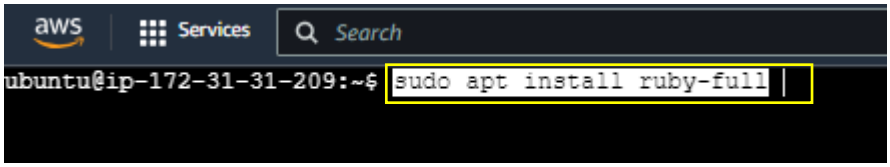
https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-ubuntu.html#:~:text=To%20install%20the%20CodeDeploy%20agent%20on%20Ubuntu%20Server,sudo%20apt%20install%20ruby-full%20sudo%20apt%20install%20wget

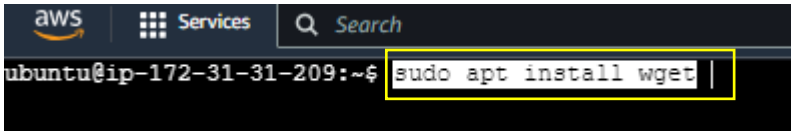
Copy 3 commands



Run in instances



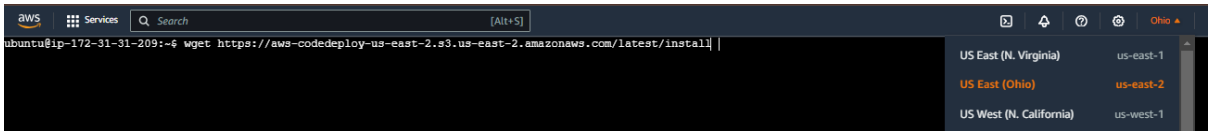




bucket-name is the name of the Amazon S3 bucket that contains the CodeDeploy Resource Kit files for your region, and *region-identifier* is the identifier for your region.

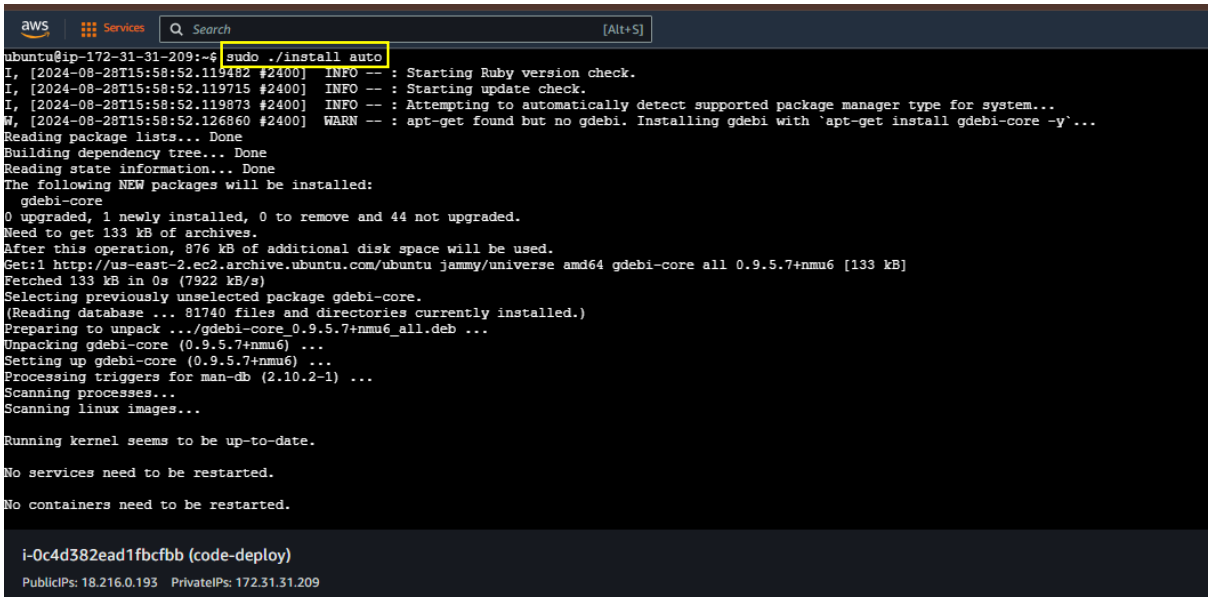
This Command works for to access our bucket codedeploy files so my region is us-east-2 so accordingly u have to change it

Copy the Command paste it

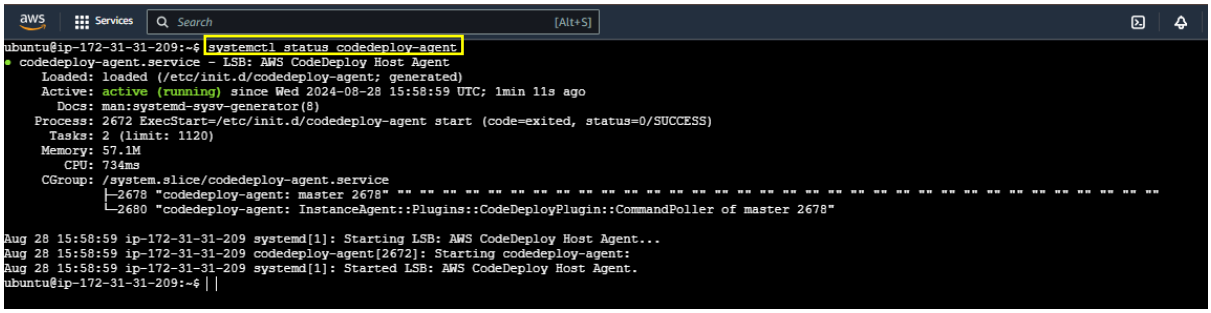


After Successful run Provide Execute Permission to install

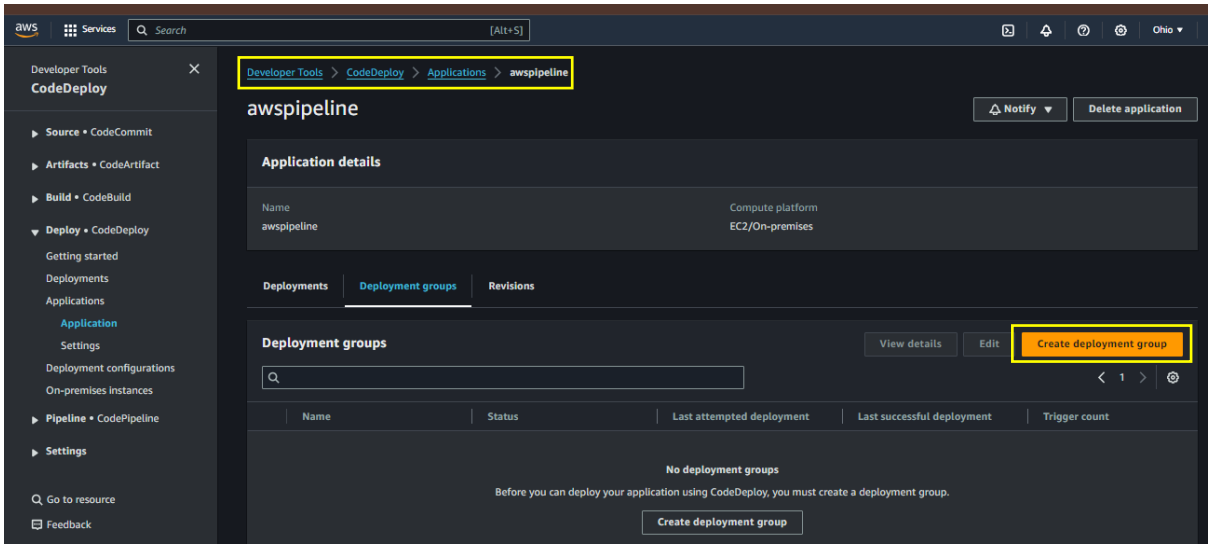
Sudo ./install auto, it will install the agent

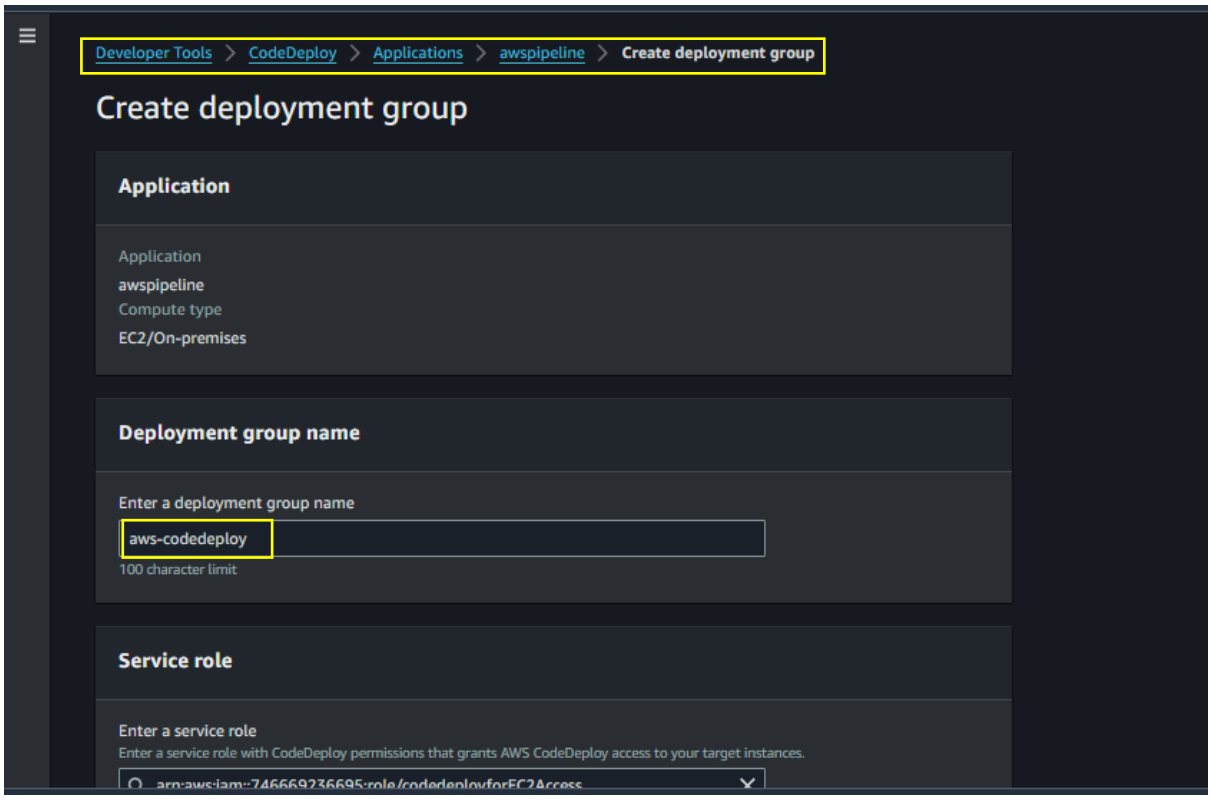


Status of codedeploy-agent, its active

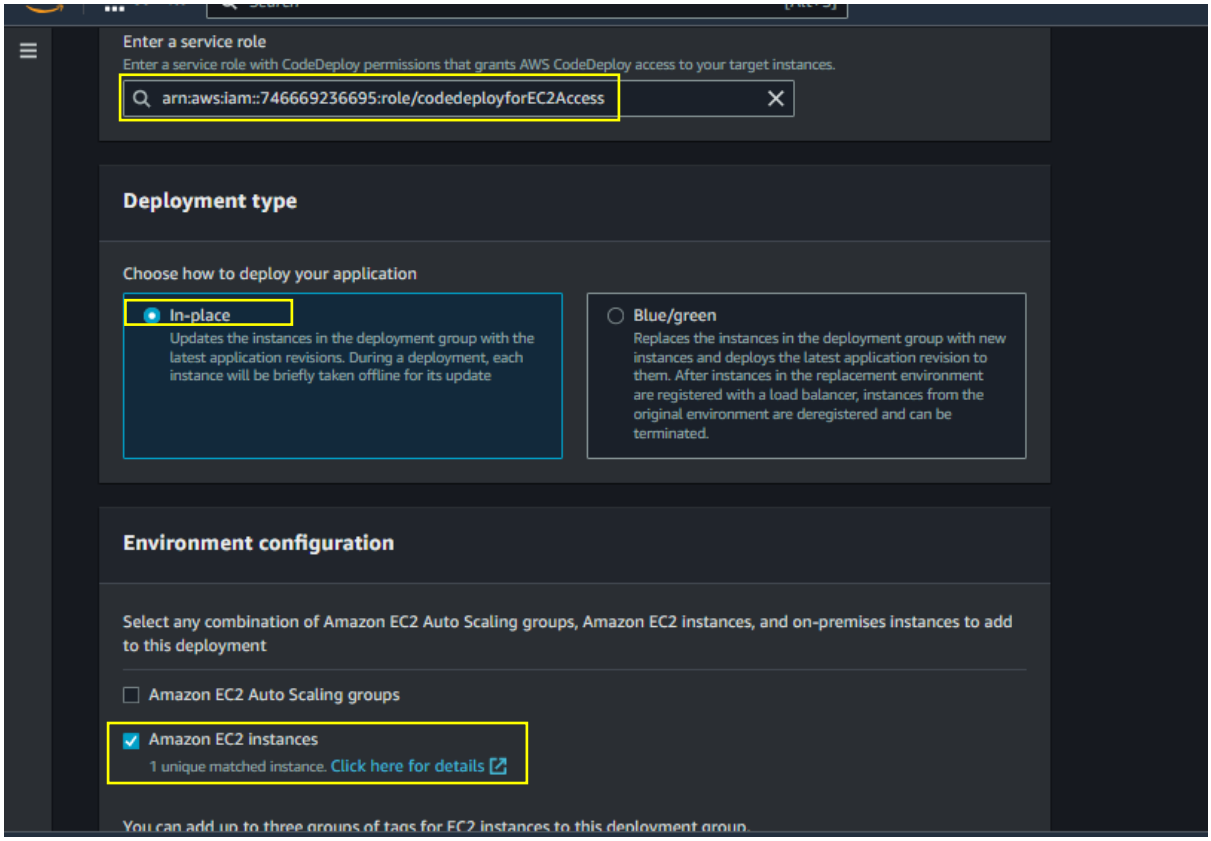


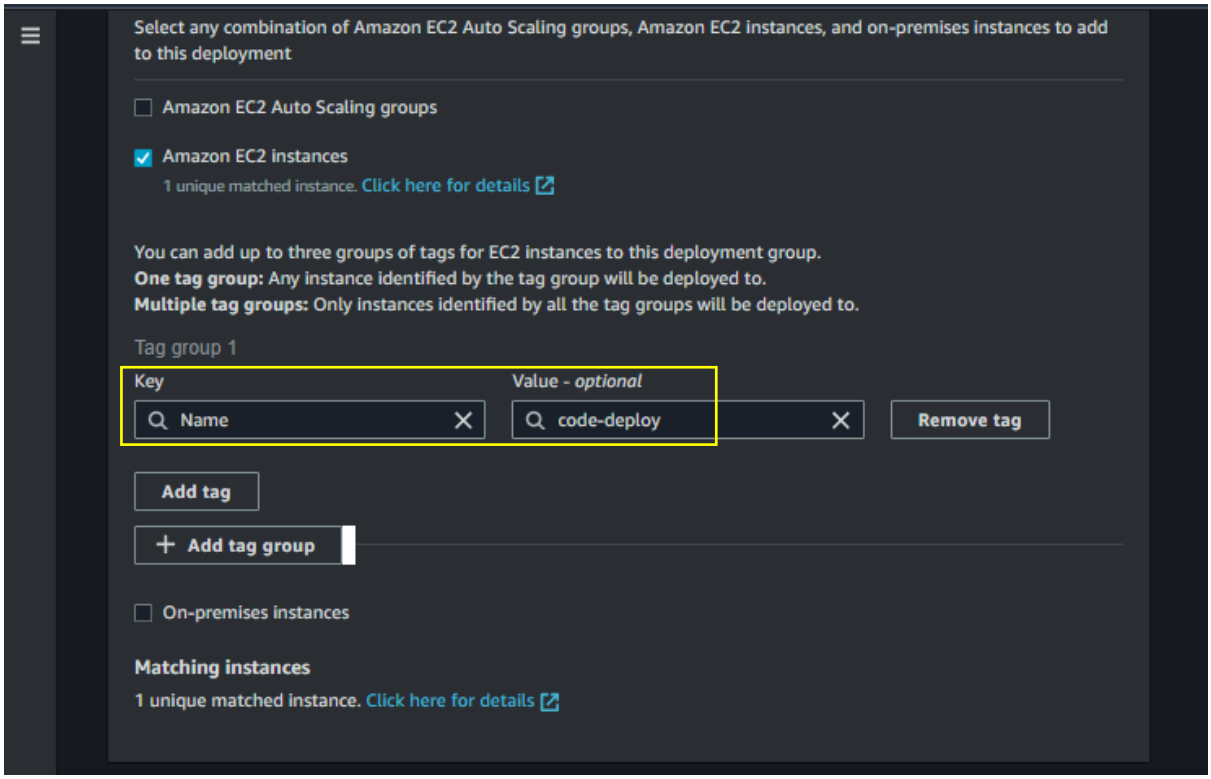
Come Again to CodeDeploy, Create Deployment Group

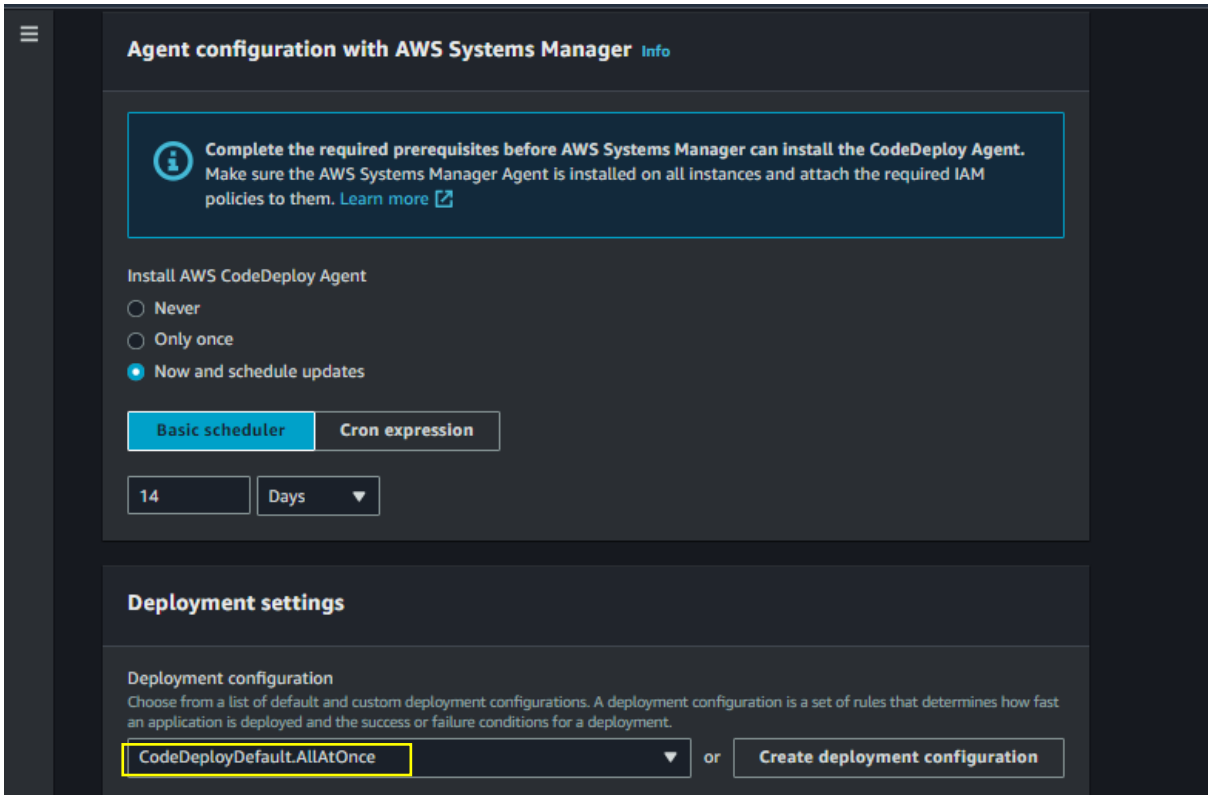


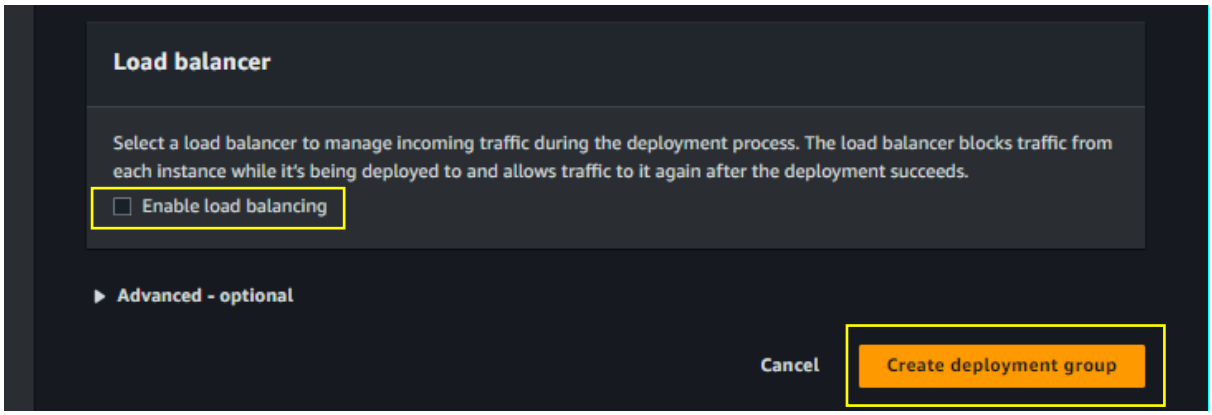


Select Role Which we are Created Previouly

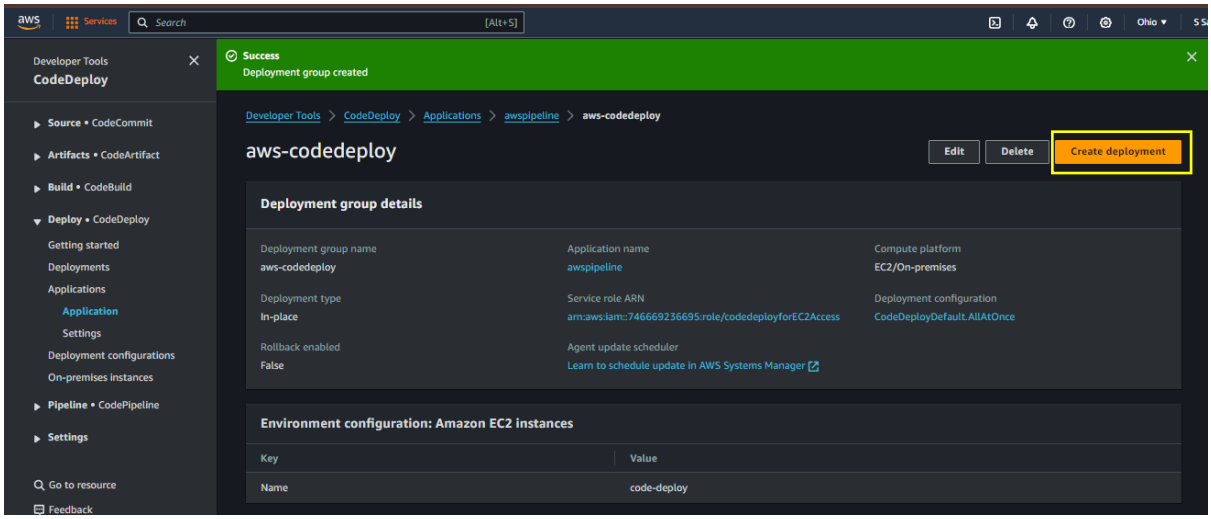




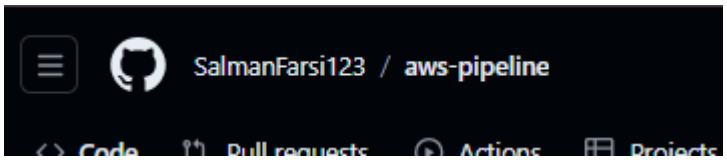




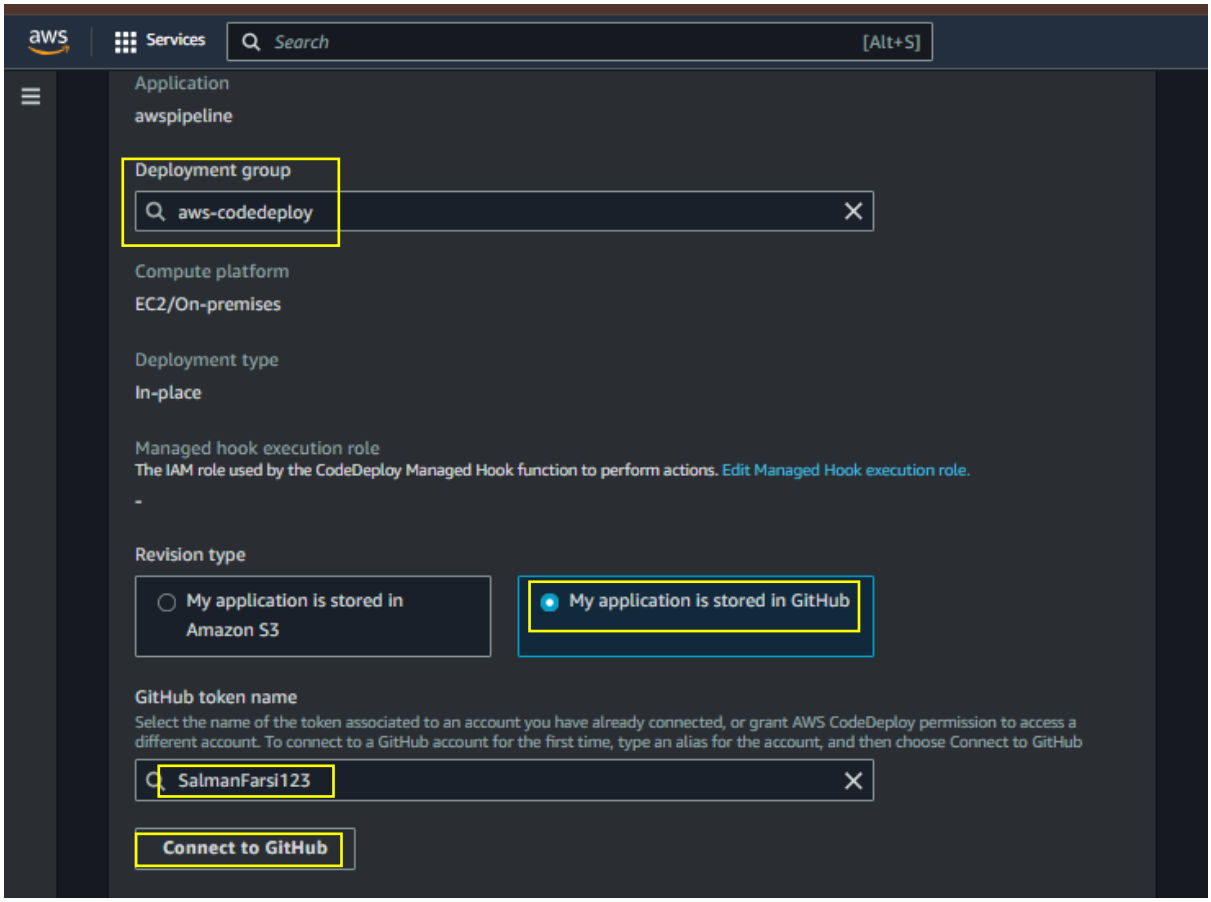
Now Create Deployment

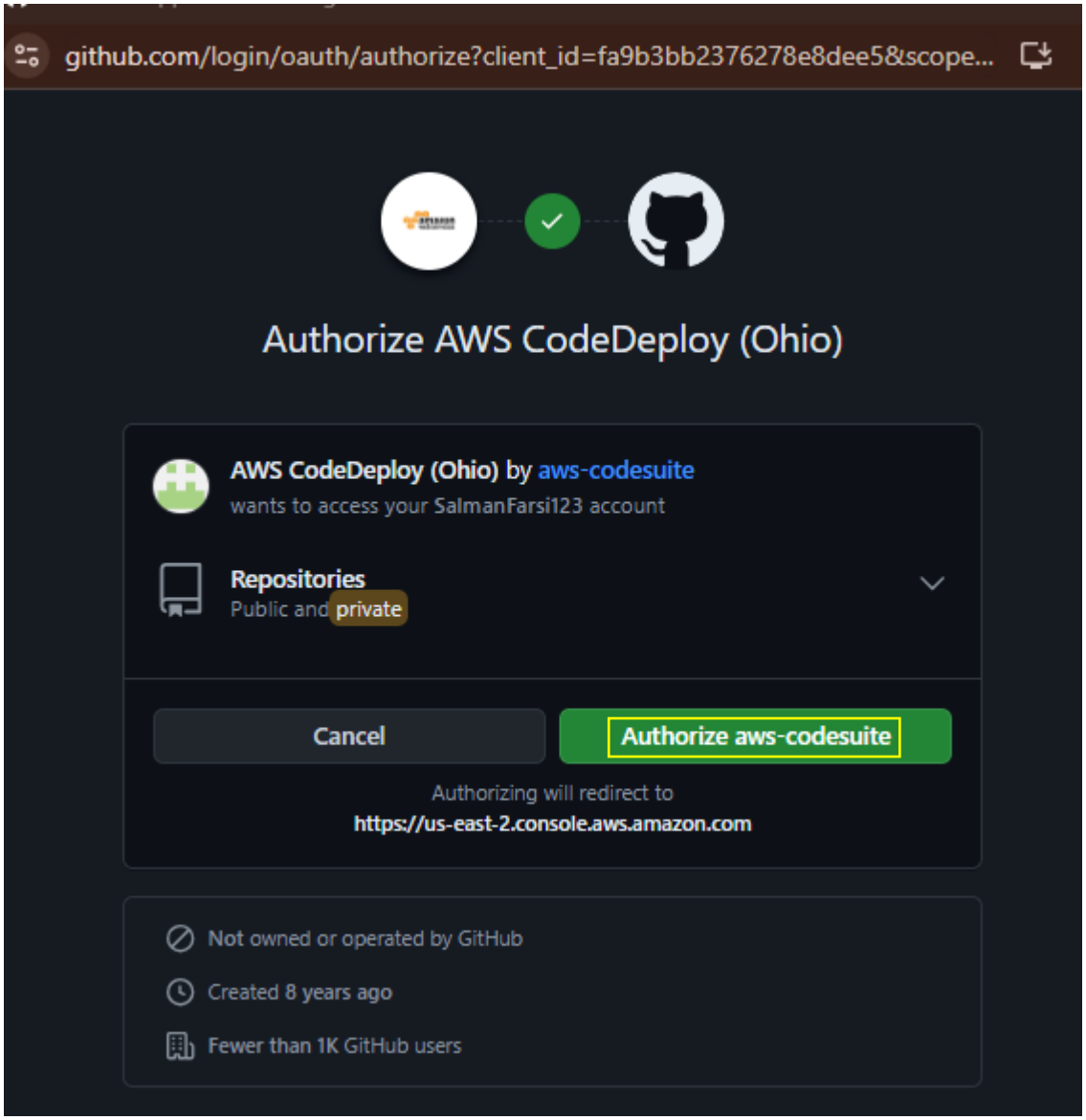


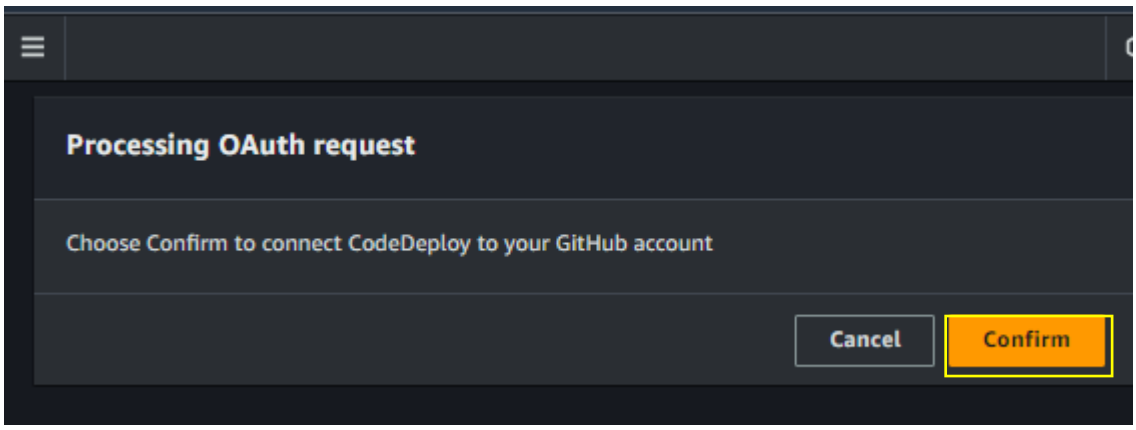
This is My Github User Name and it will use in further

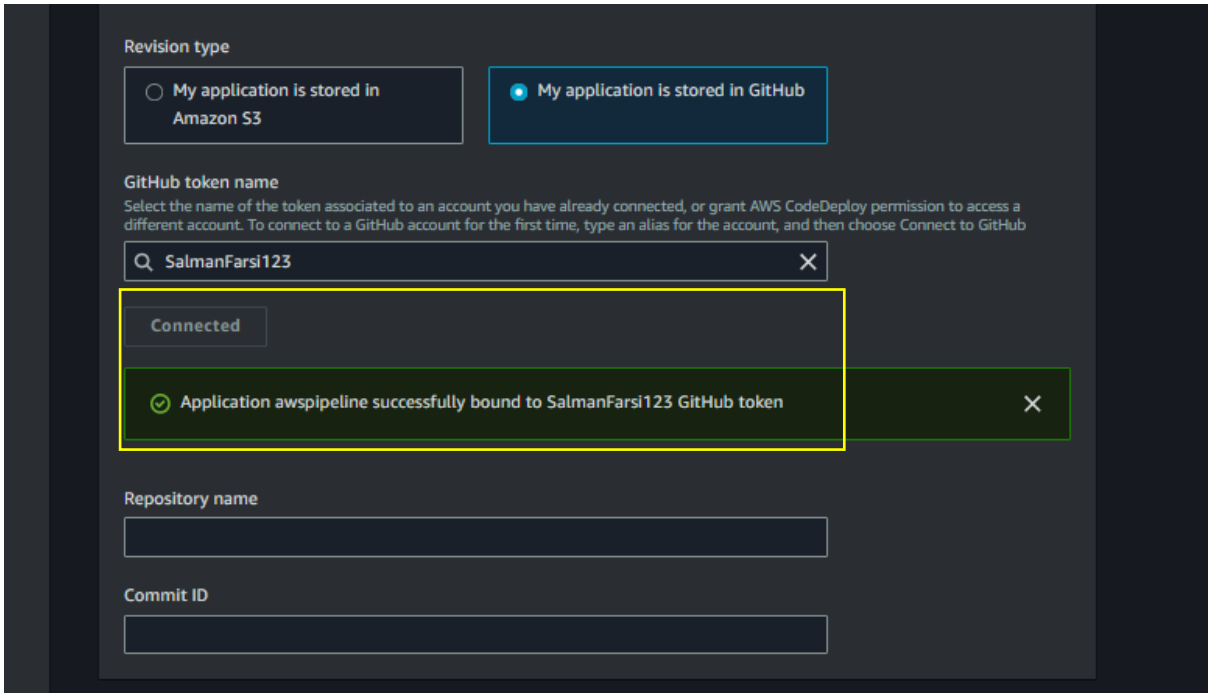


Select your Deployment Group, Click to Connect

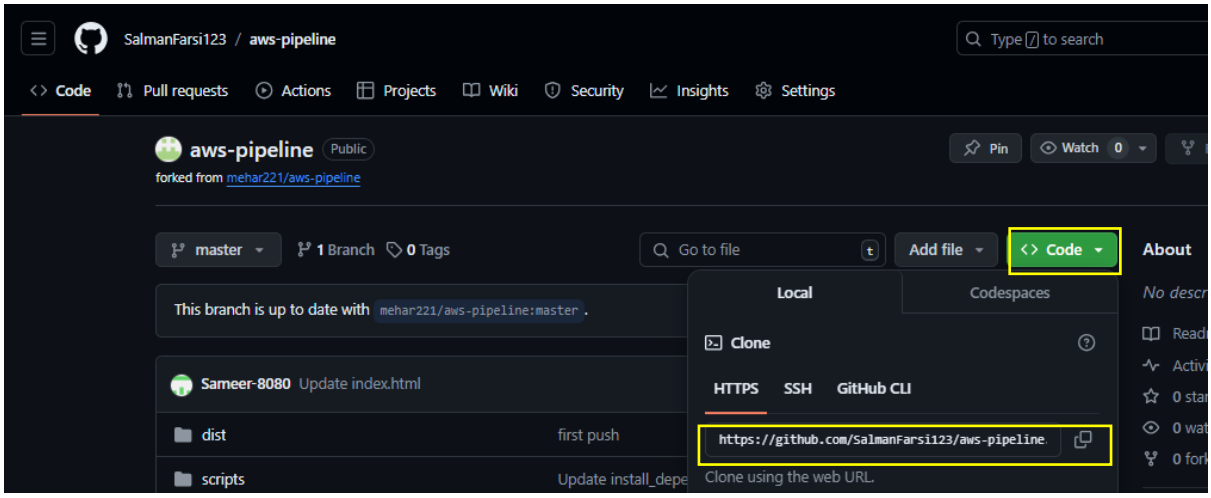



github.com/login/oauth/authorize?client_id=fa9b3bb2376278e8dee5&scope...
Authorize AWS CodeDeploy (Ohio)
AWS CodeDeploy (Ohio) by aws-codesuite
wants to access your SalmanFarsi123 account
Repositories
Public and private
Cancel
Authorize aws-codesuite
Authorizing will redirect to
https://us-east-2.console.aws.amazon.com
Not owned or operated by GitHub
Created 8 years ago
Fewer than 1K GitHub users

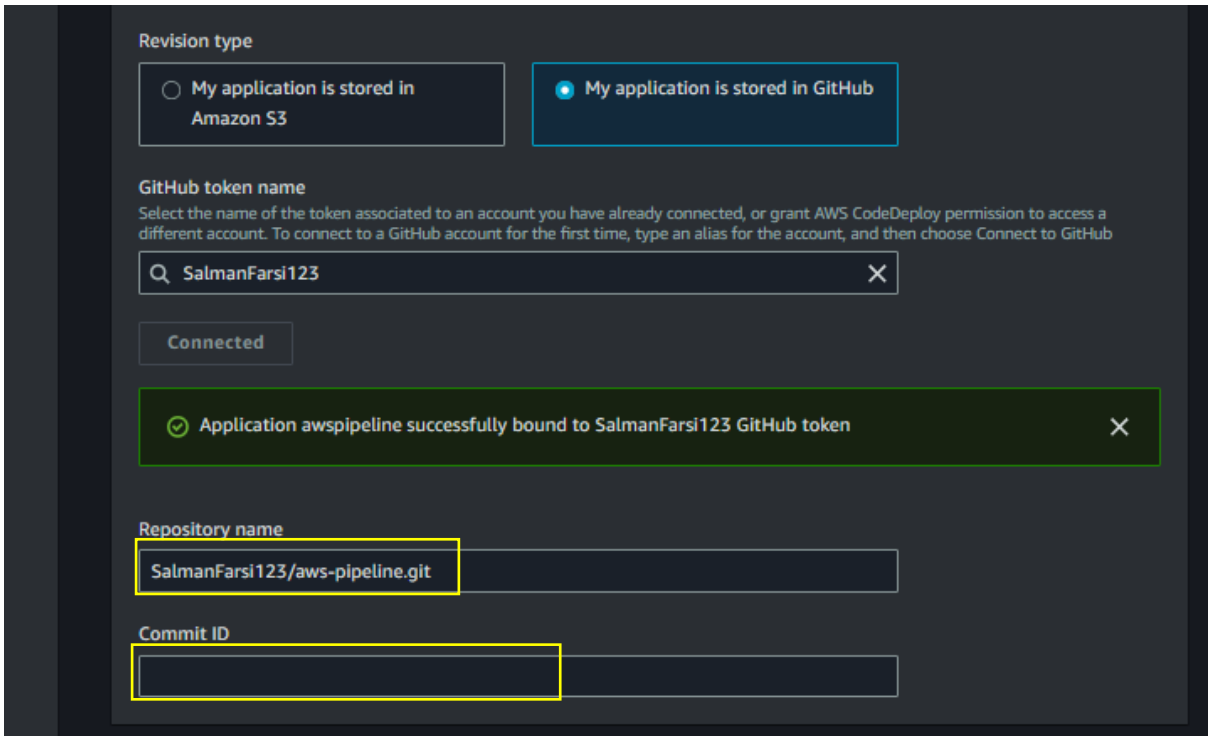

Processing OAuth request
Choose Confirm to connect CodeDeploy to your GitHub account
Cancel
Confirm



Come Here And Copy your Repo Name

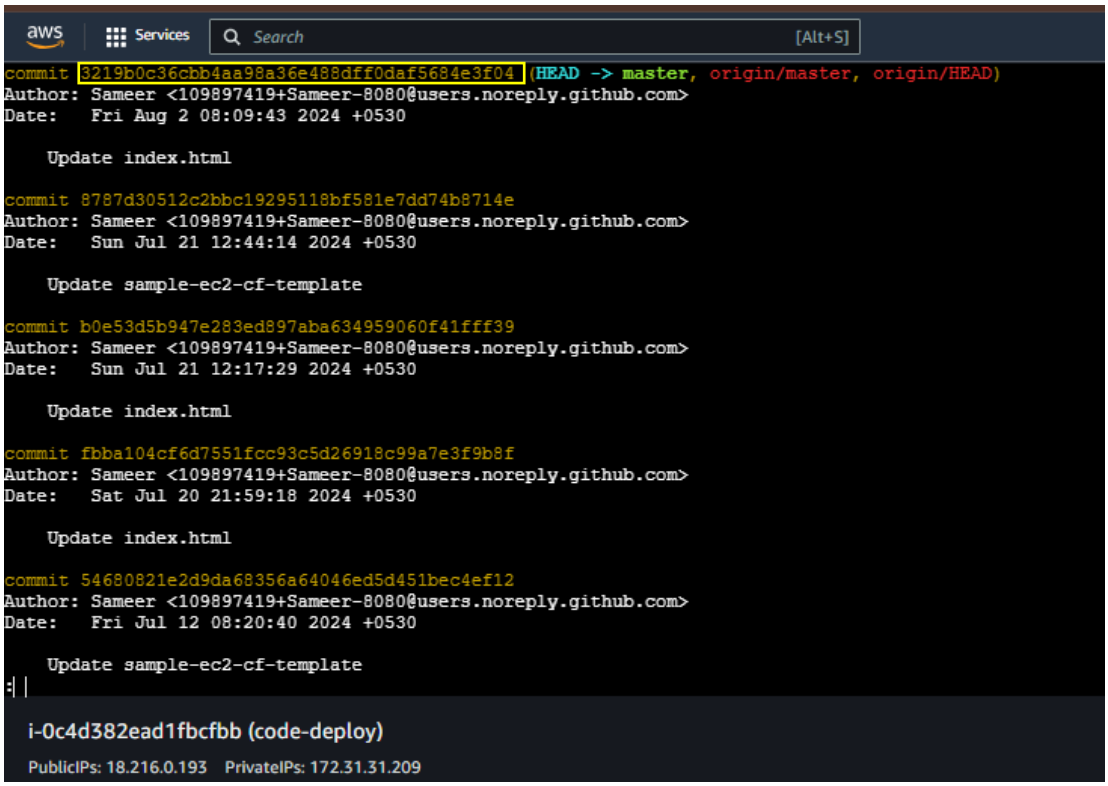


U need to Remove .git, otherwise u will get an error and Now we have to Know latest commit id

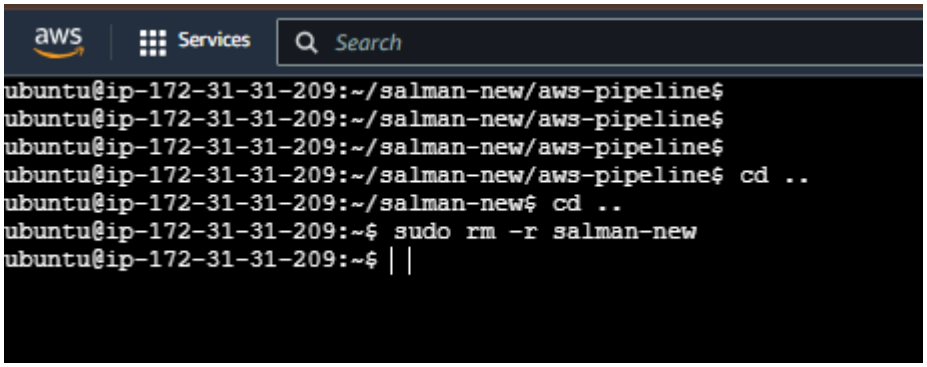


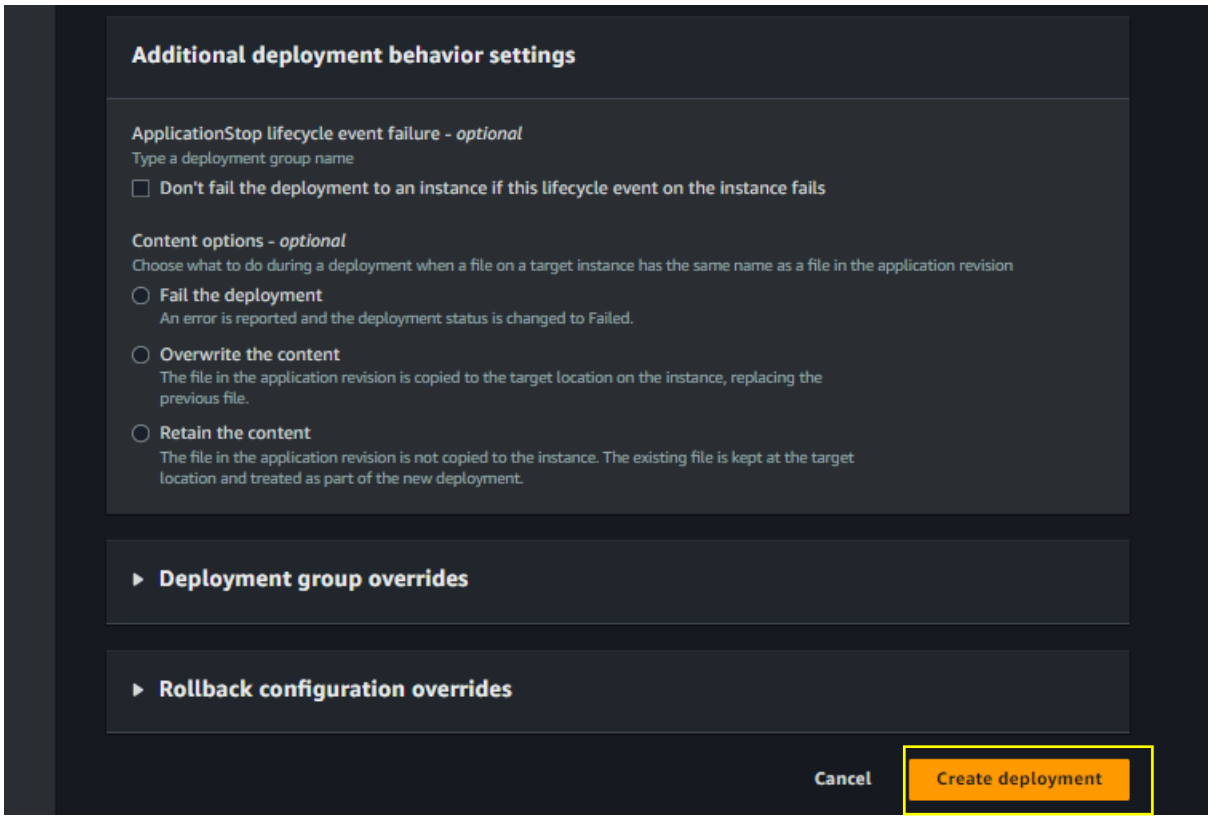
So, Here I am Creating a New Directory

Run git log and Copy latest one, this is the latest changes of project and paste it on creating on deployment, I didn't attached that screenshot

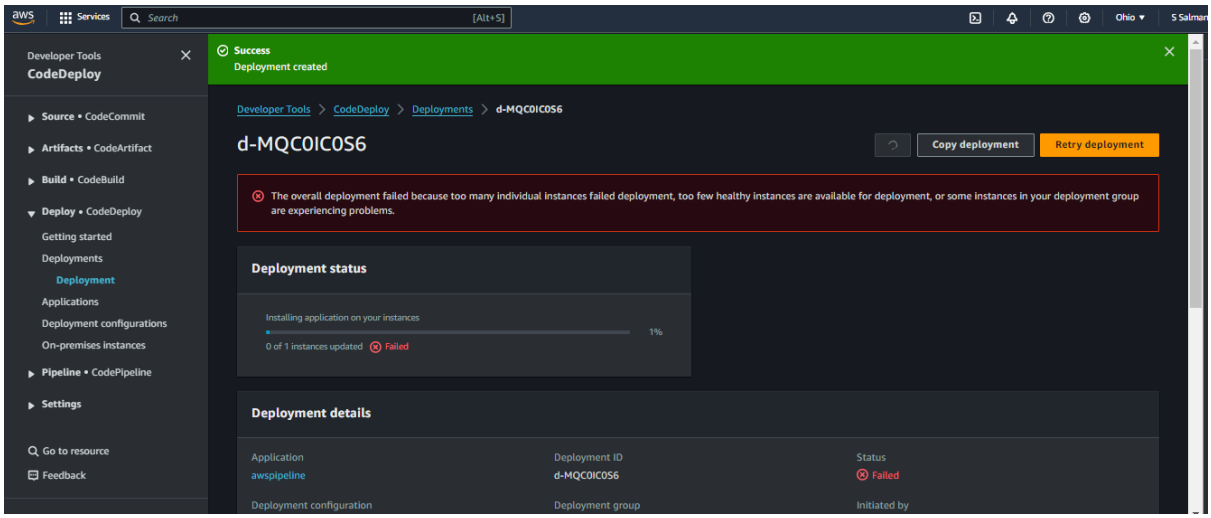


I removed that created directory, u creates only to know latest commit id

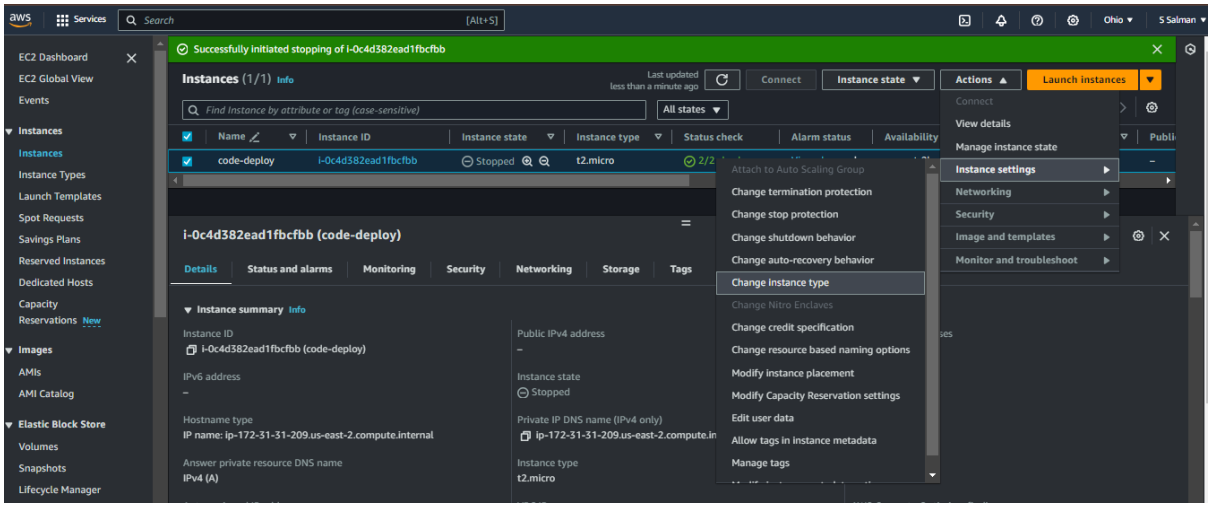


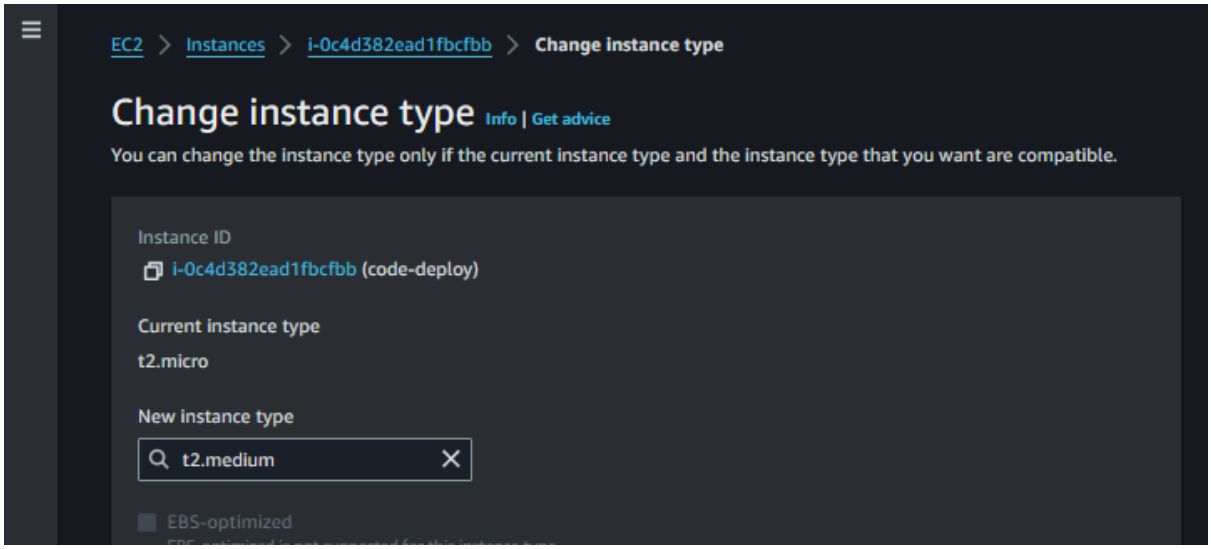


Got an error because of t2.medium and .git extension I have gave in repo

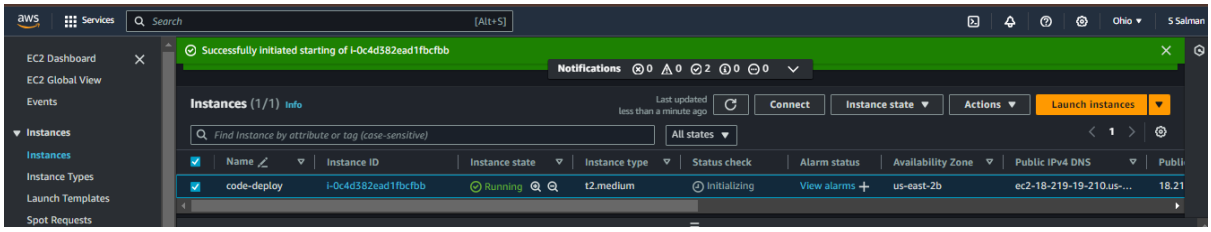


Changing instances type t2. Medium , I notified in above

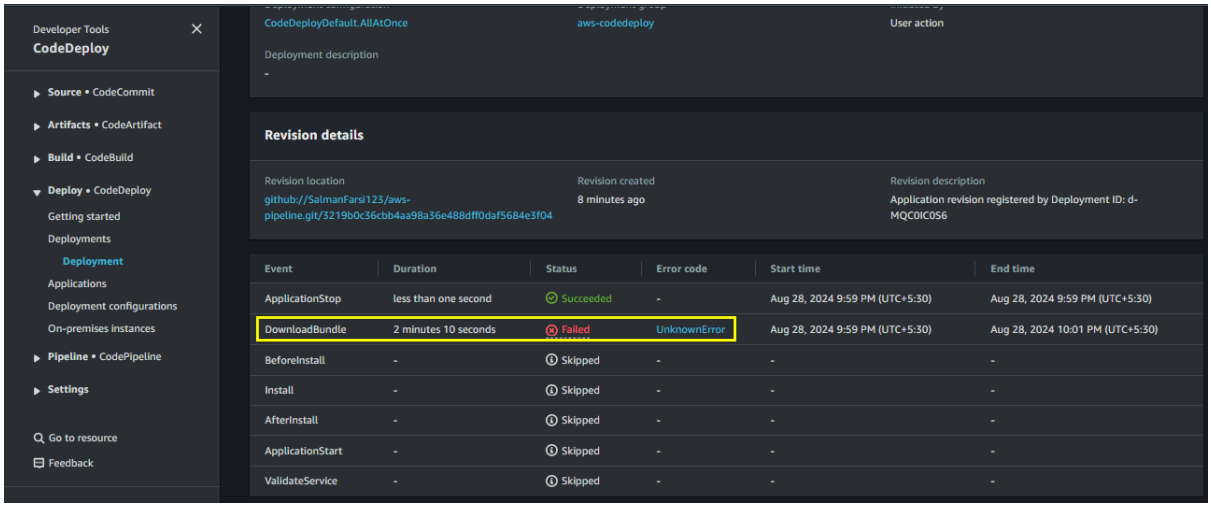




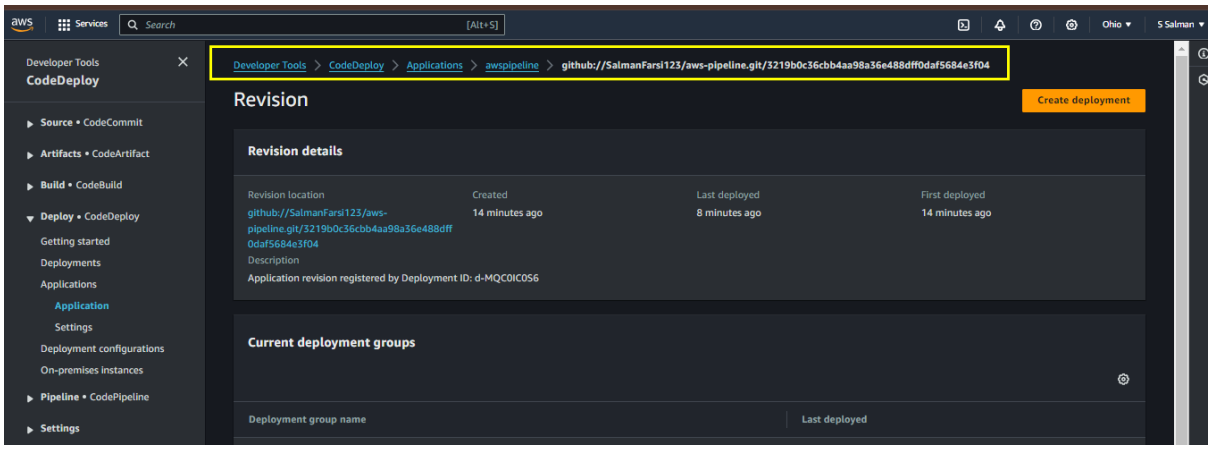
Changed successfully



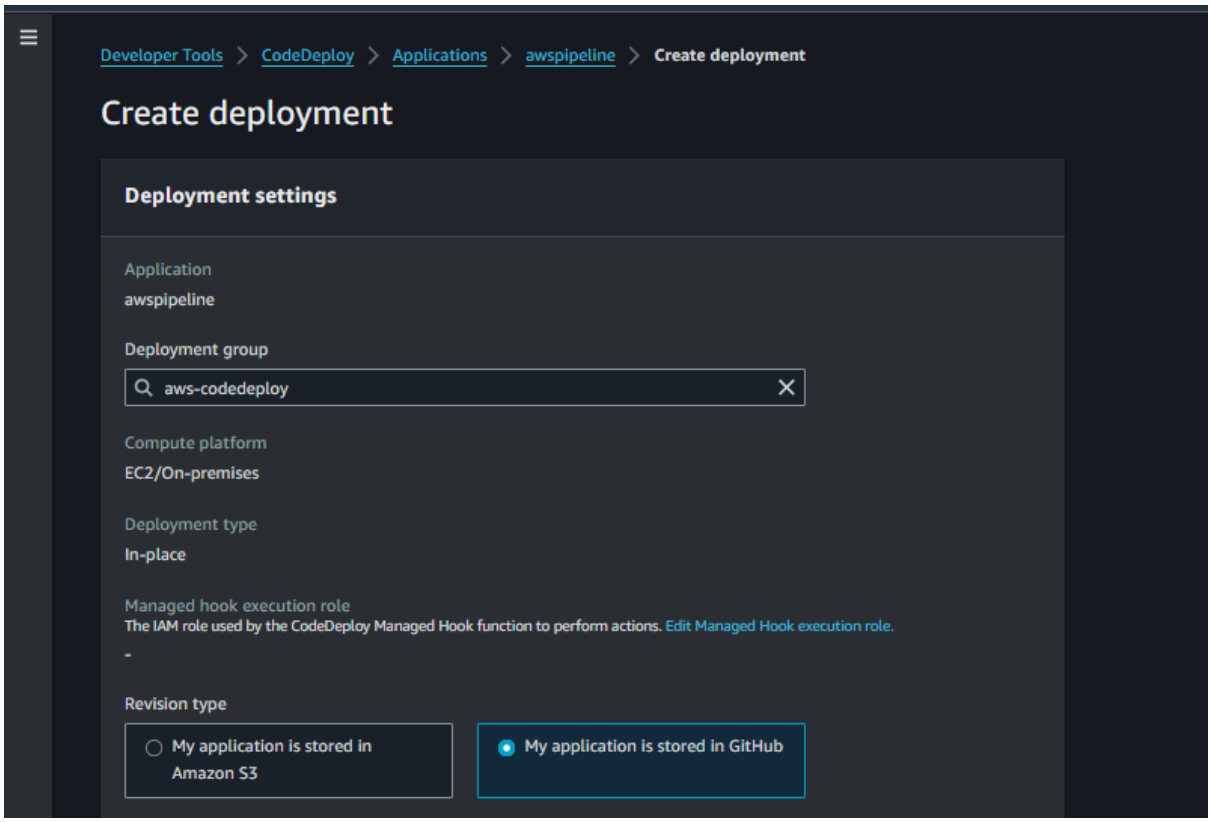
Got an another got an error to download bundle



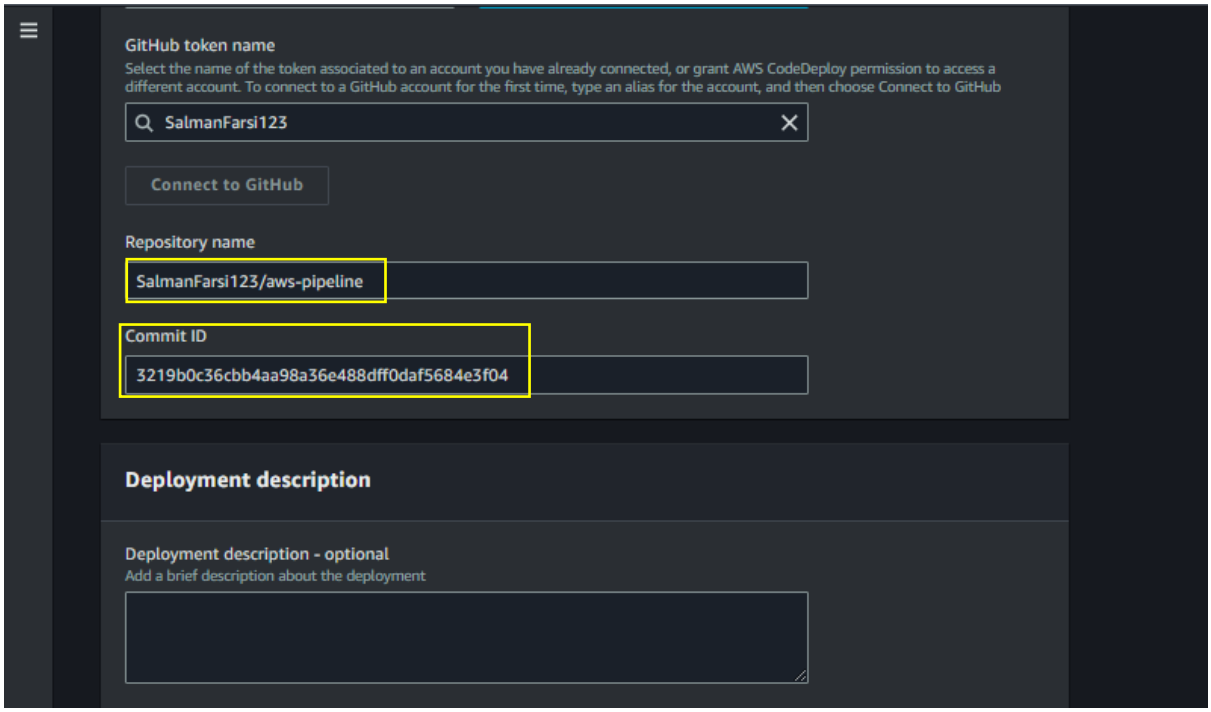
Come Here Click on to create deployment



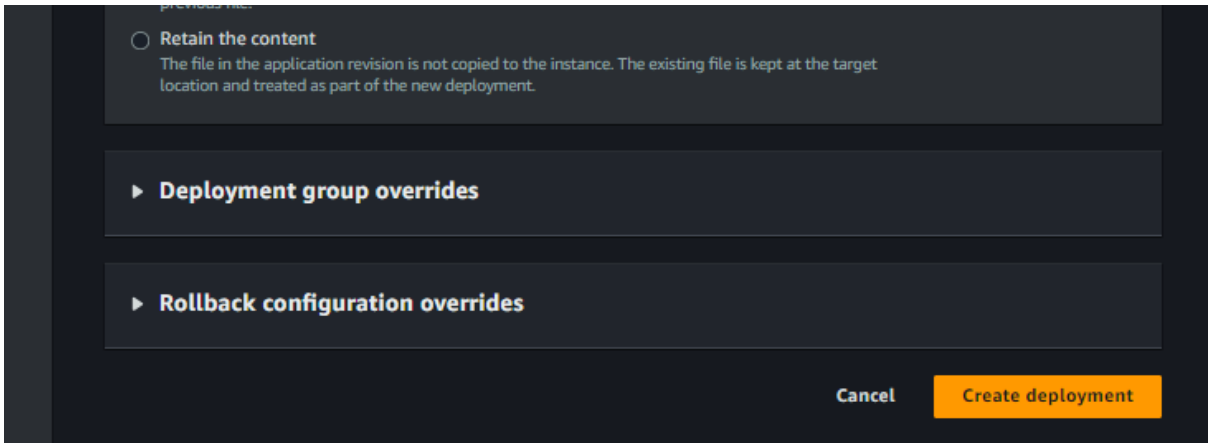
Everything same



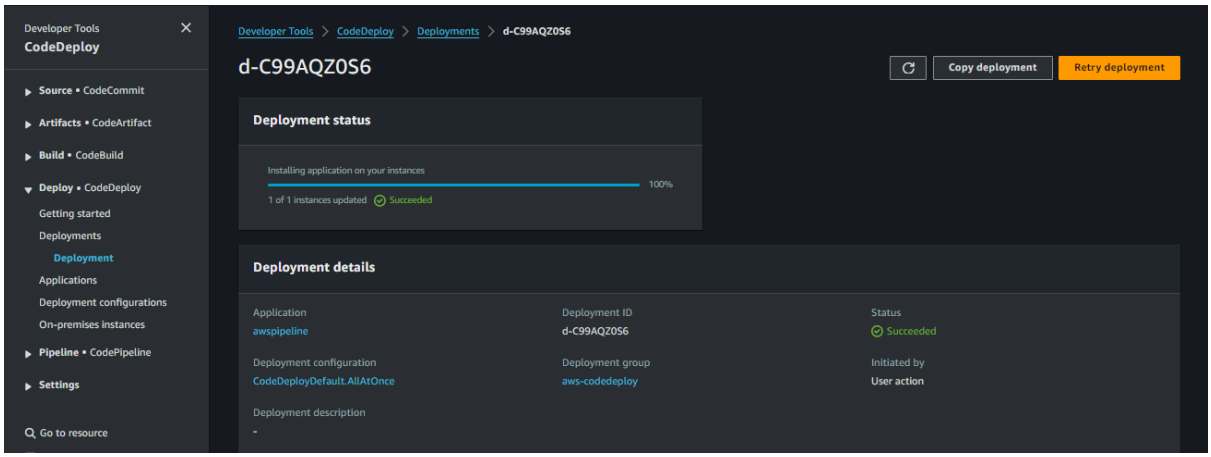
Here need to be remove .git



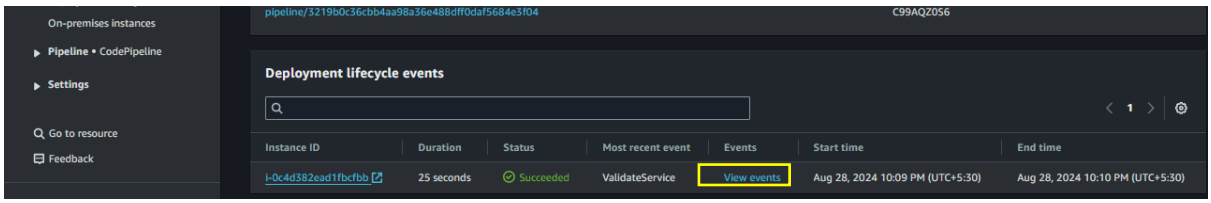
Create Deployment



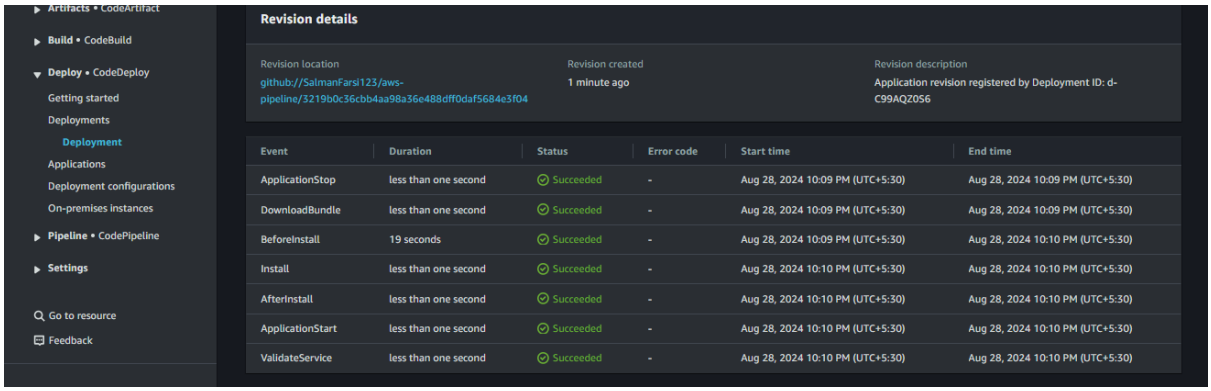
Now its successfully Created



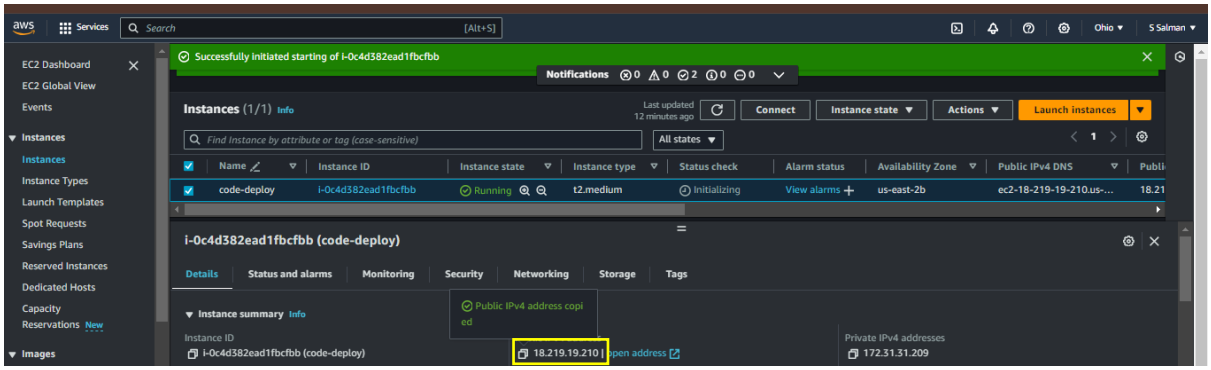
Scroll below, u will be found this Click on View Events



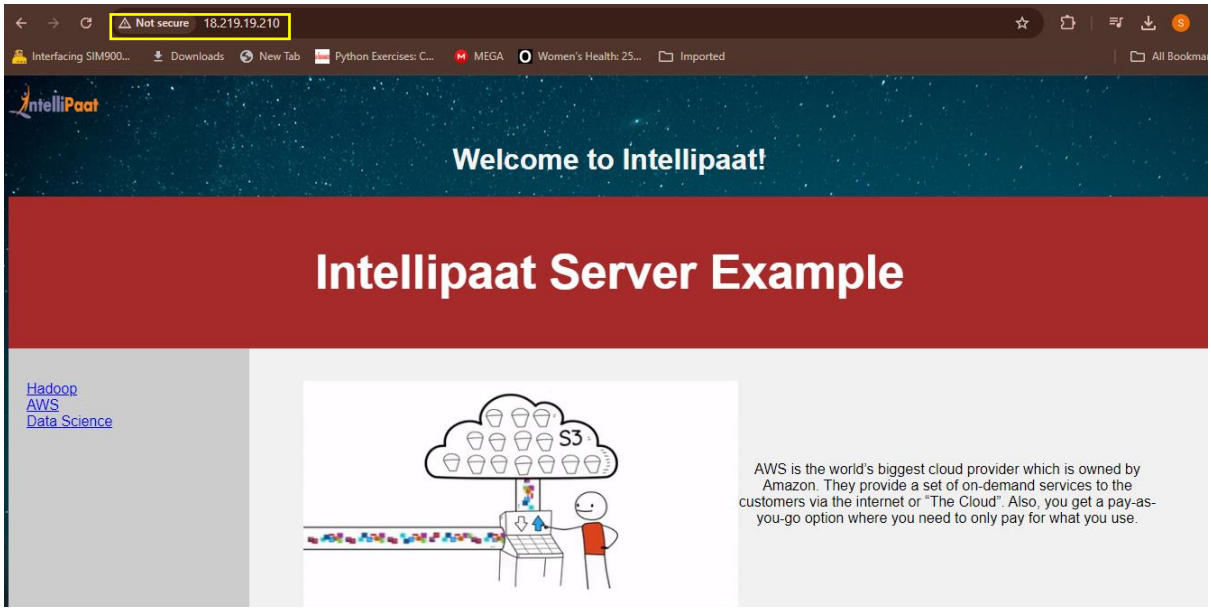
All the Events Succeeded



Now copy the Public IP

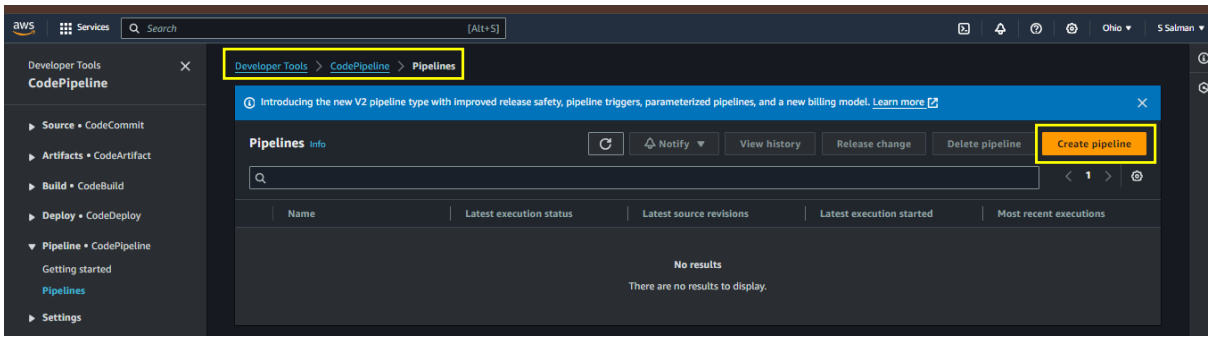


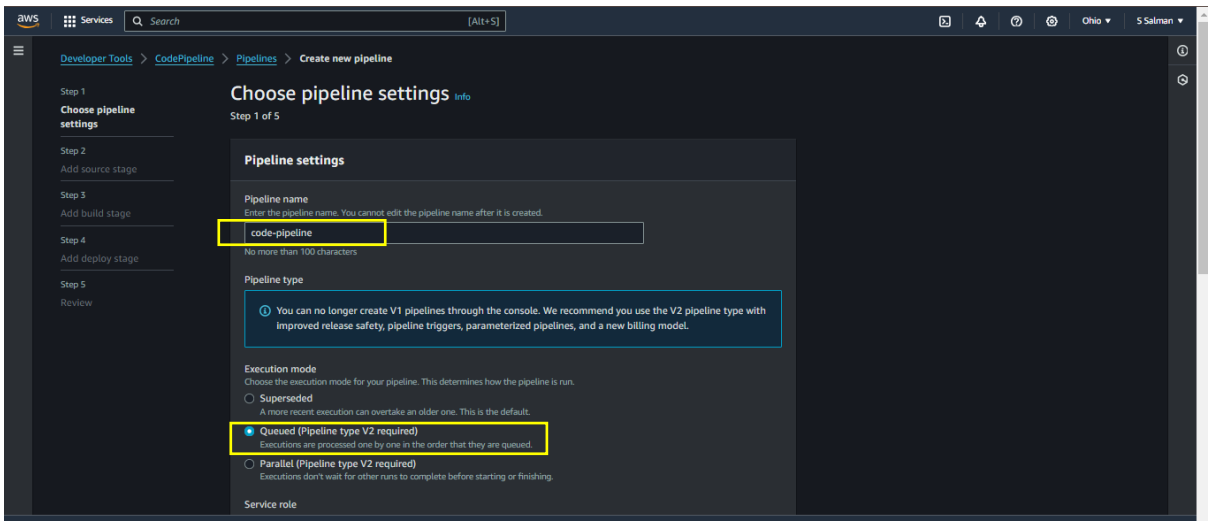
Application Deployed Successfully



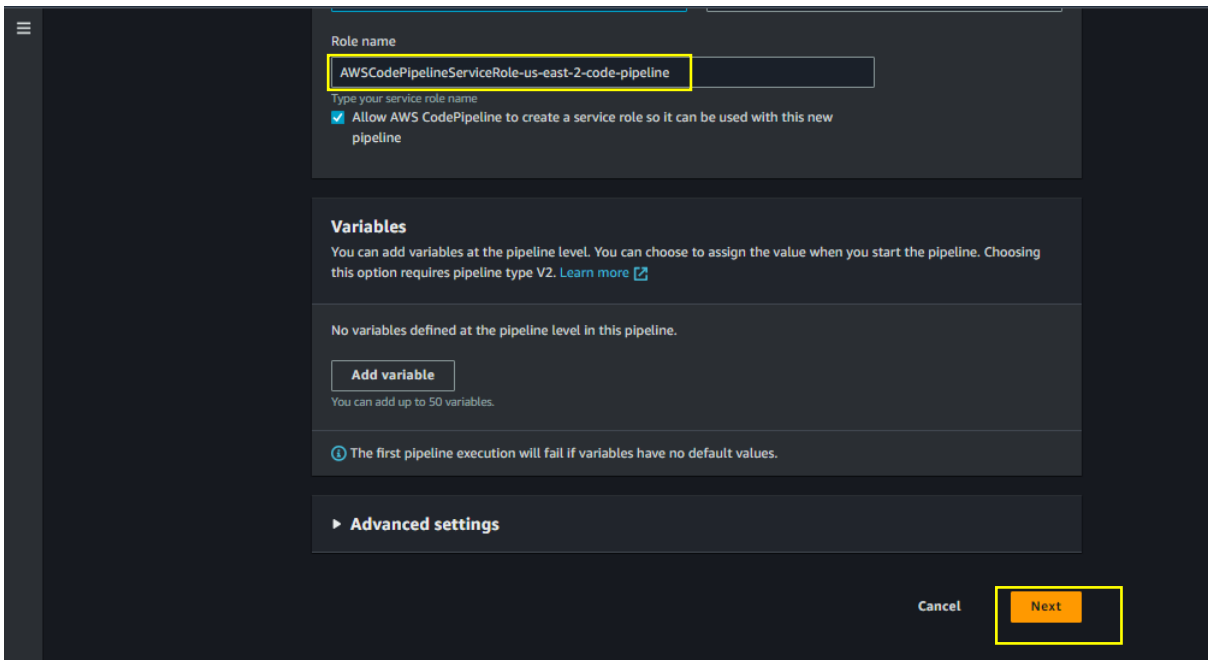
CODE PIPELINE Demo:

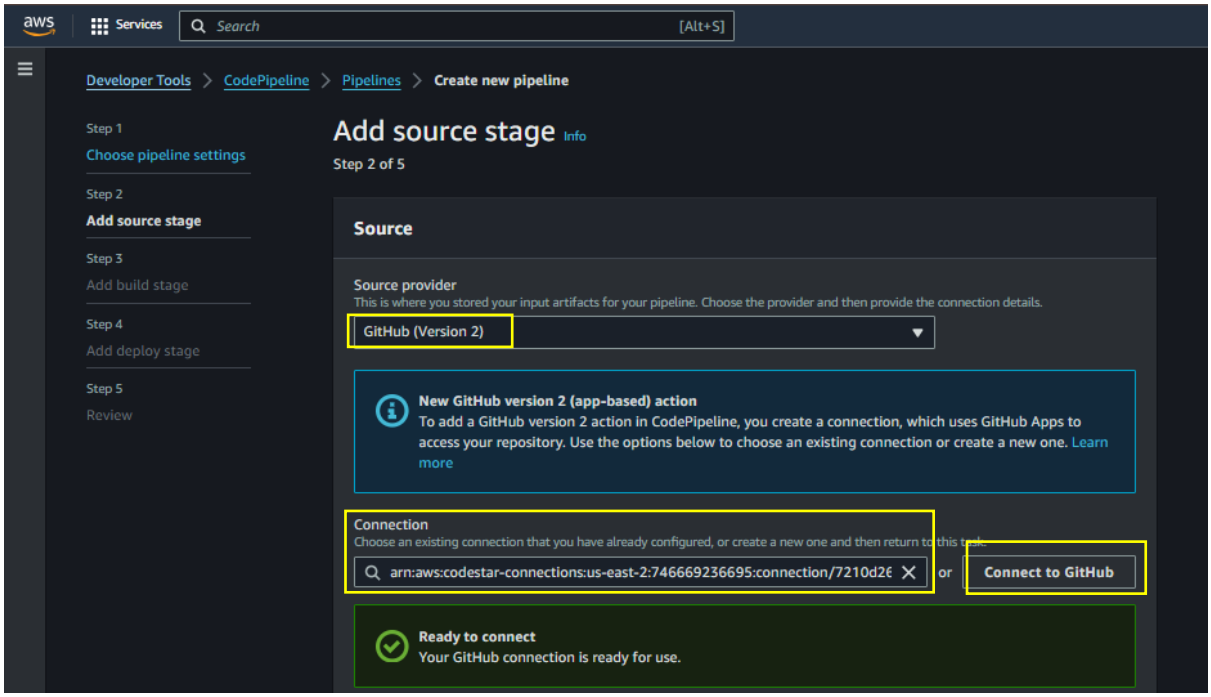
Search Codepipeline

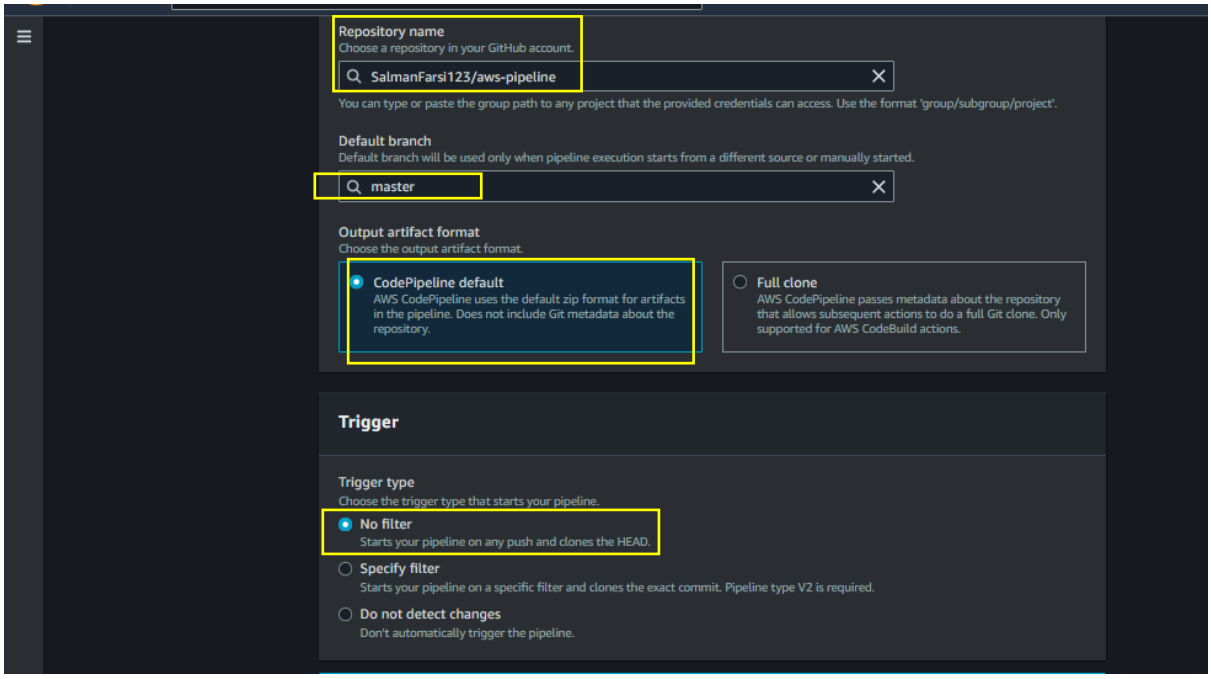


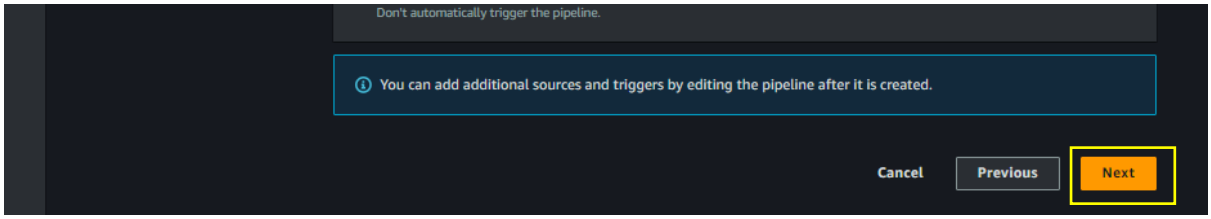


We created Codedeployment and codepipeline role as a 1 role

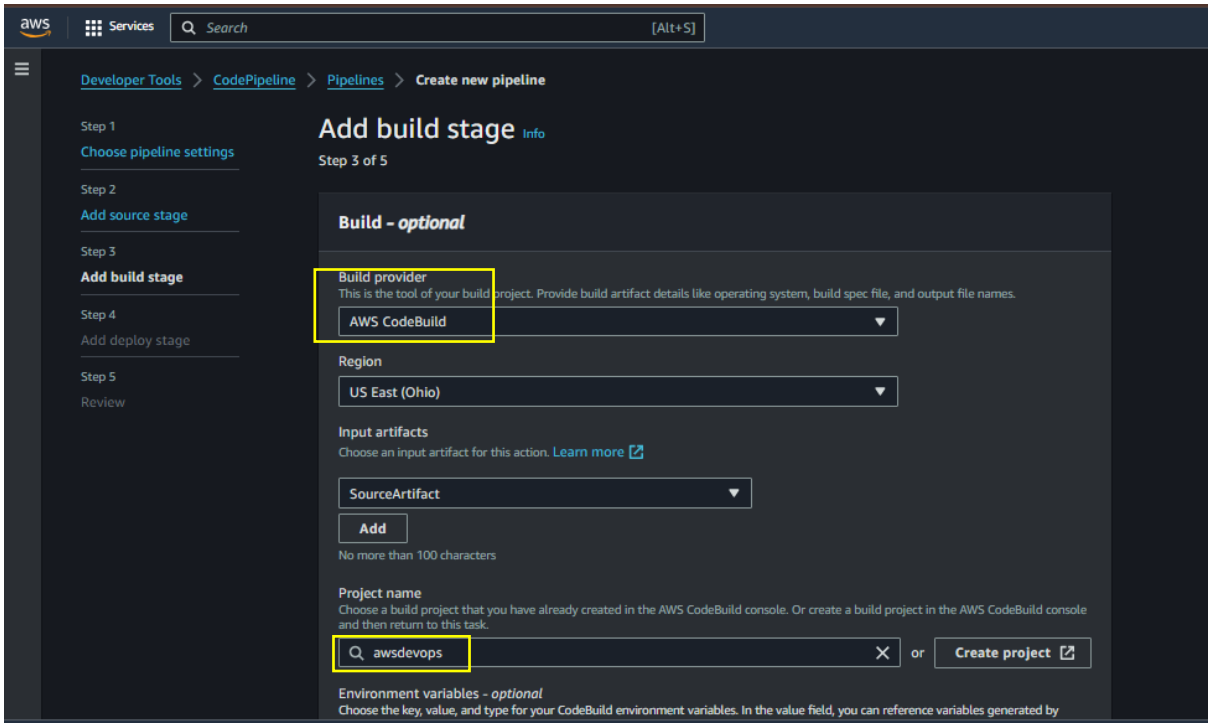


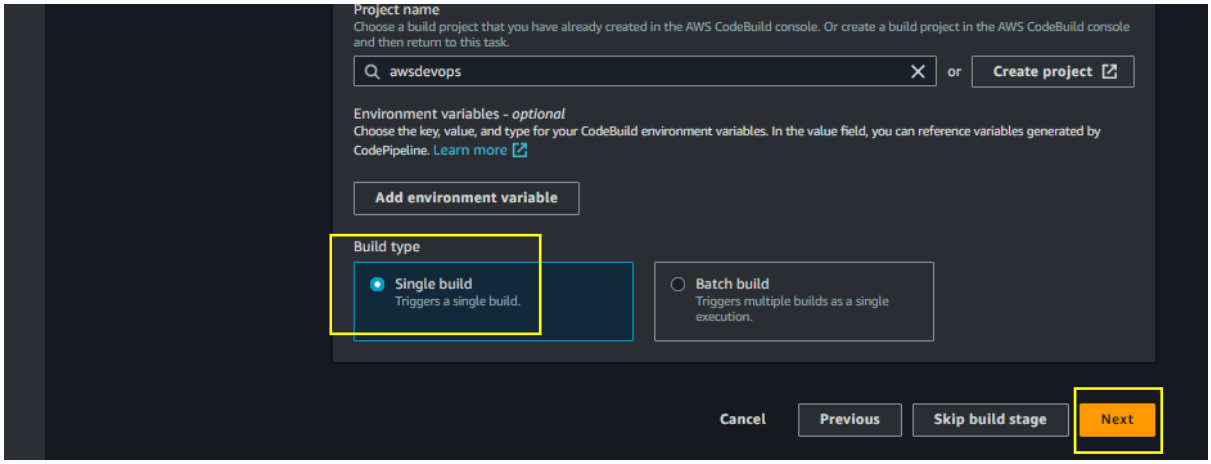

aws
Services
Search
[Alt+S]
Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add source stage Info
Step 2 of 5
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 2)
New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more
Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.
arn:aws:codestar-connections:us-east-2:746669236695:connection/7210d2e
or
Connect to GitHub
Ready to connect
Your GitHub connection is ready for use.


Repository name
Choose a repository in your GitHub account.
SalmanFarsi123/aws-pipeline
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.
Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.
master
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.
Trigger
Trigger type
Choose the trigger type that starts your pipeline.
No filter
Starts your pipeline on any push and clones the HEAD.
Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
Do not detect changes
Don't automatically trigger the pipeline.

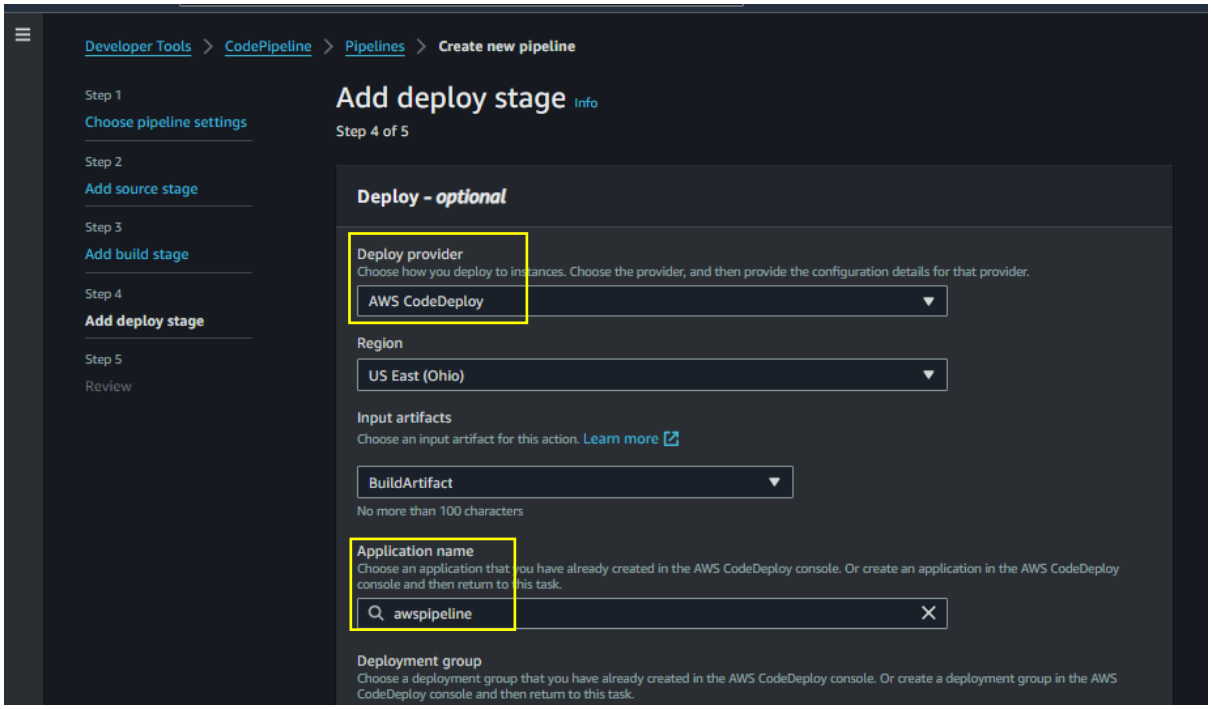

Don't automatically trigger the pipeline.
You can add additional sources and triggers by editing the pipeline after it is created.
Cancel
Previous
Next

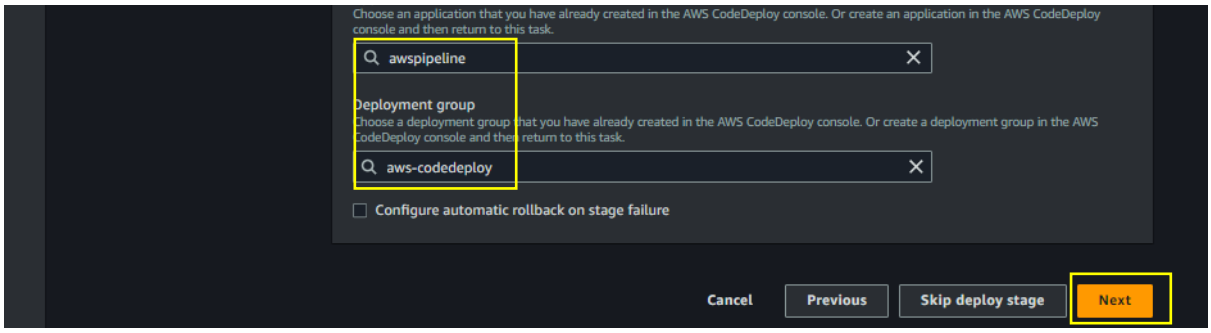
awsdevops is codebuild project name, we need to attach codebuild also

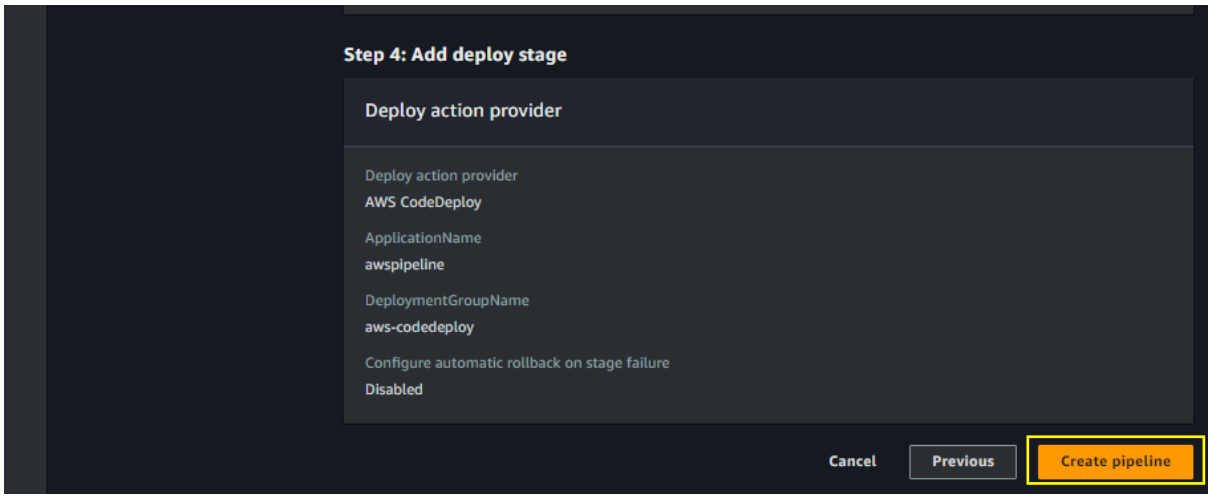




Now here we need to select that codedeploy

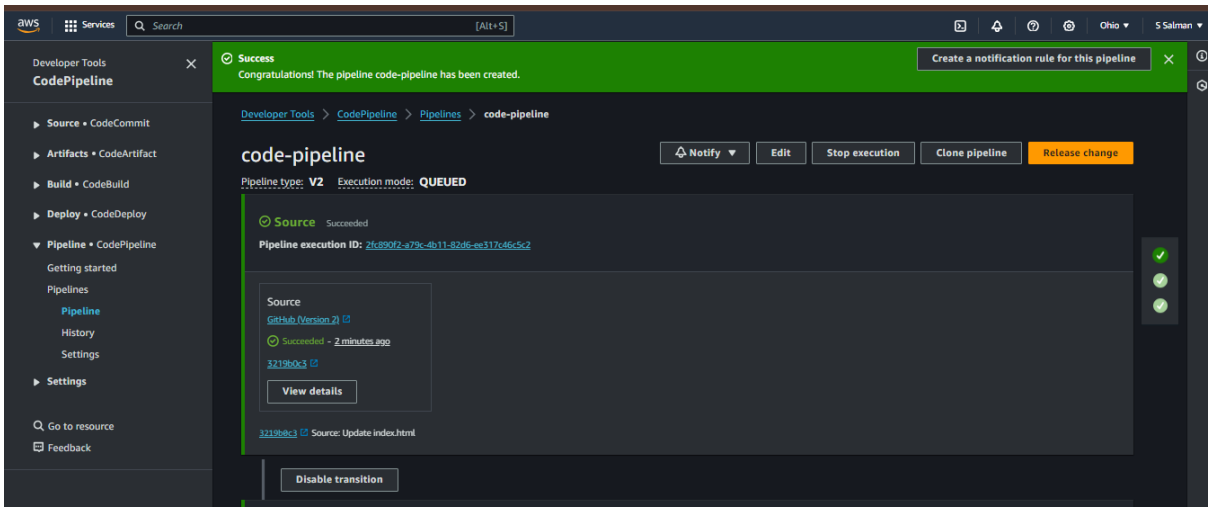


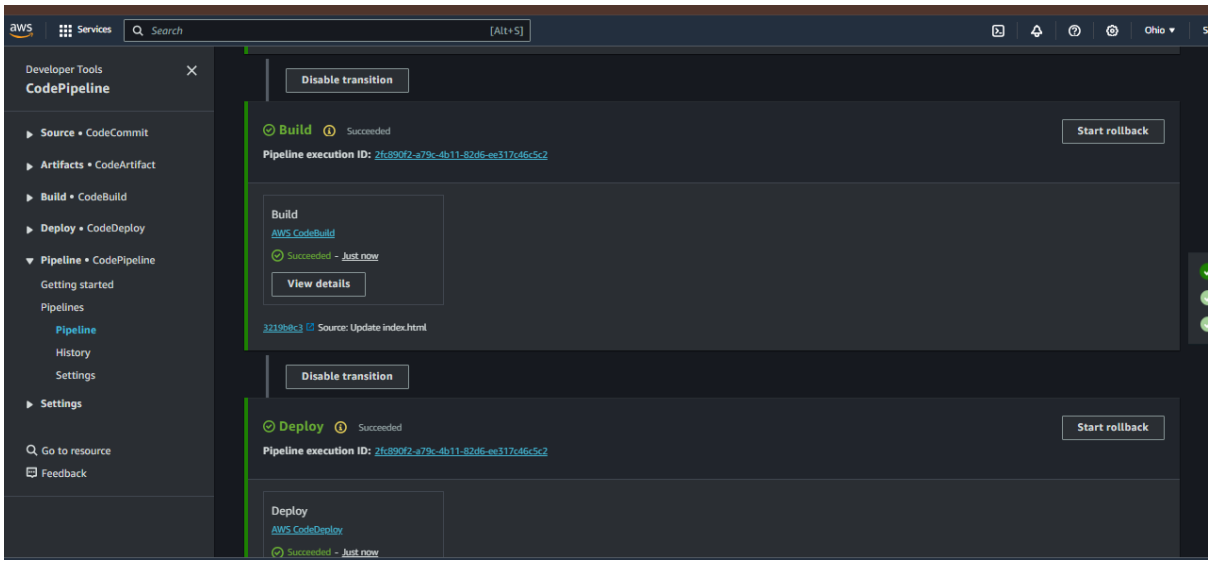


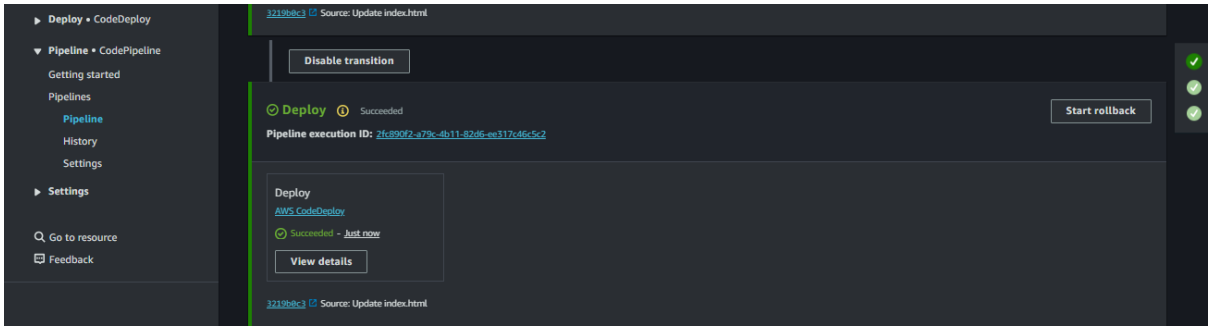


Now this is the CI/CD Pipeline Structure

Source From GitHub>And then CodeBuild> CodeDeploy> CodePipeline, Pipeline has been created successfully, according to setup push or pull will be happen and our to application will be change and this is known as Continuous Integration and Continuous Deployment







AWS CodeCommit :

is a fully managed source control service that hosts secure Git repositories, allowing you to store and manage your code in the cloud. It integrates with other AWS services and provides features like pull requests, branching, and code reviews, making it suitable for collaborative software development.

Terminology in Azure DevOps :

AWS CodeCommit is comparable to Azure Repos in Azure DevOps.

AWS CodeBuild :

is a fully managed service that automates the process of compiling your source code, running tests, and producing build artifacts. It helps streamline continuous integration by automatically building and testing your code every time changes are pushed to your repository.

Terminology in Azure DevOps :

AWS CodeBuild is comparable to Azure Pipelines (Build pipelines)in Azure DevOps.

AWS CodeDeploy :

is a service that automates the deployment of applications to various environments, such as Amazon EC2 instances, on-premises servers, and AWS Lambda. It helps ensure that your application is deployed consistently and with minimal downtime.

Terminology in Azure DevOps :

AWS CodeDeploy is comparable to Azure Pipelines (Release pipelines) in Azure DevOps.

AWS CodePipeline:

is a service that automates the entire release process, allowing you to build, test, and deploy your code every time there's a change. It helps streamline continuous delivery by connecting all the stages of your CI/CD pipeline into a single workflow.

Terminology in Azure DevOps :

AWS CodePipeline is comparable to Azure Pipelines in Azure DevOps.