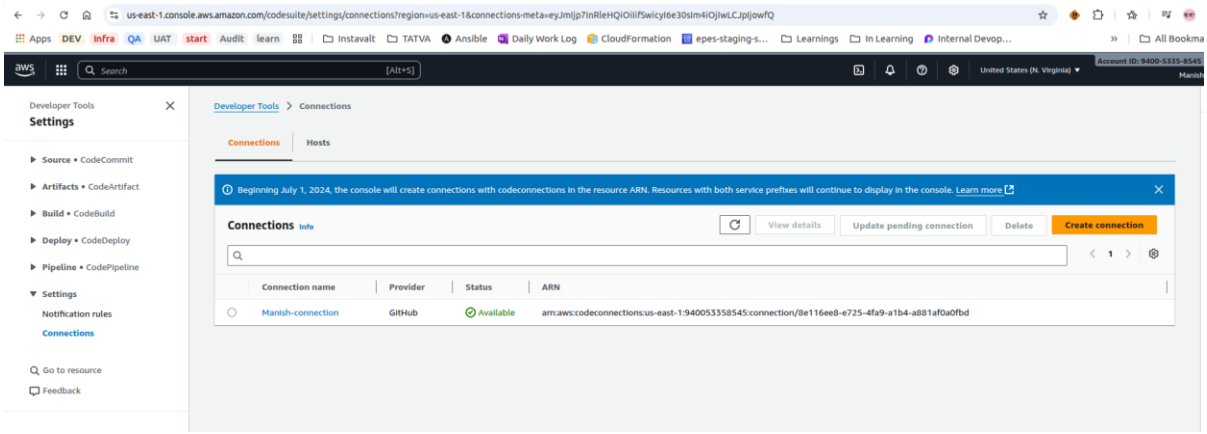
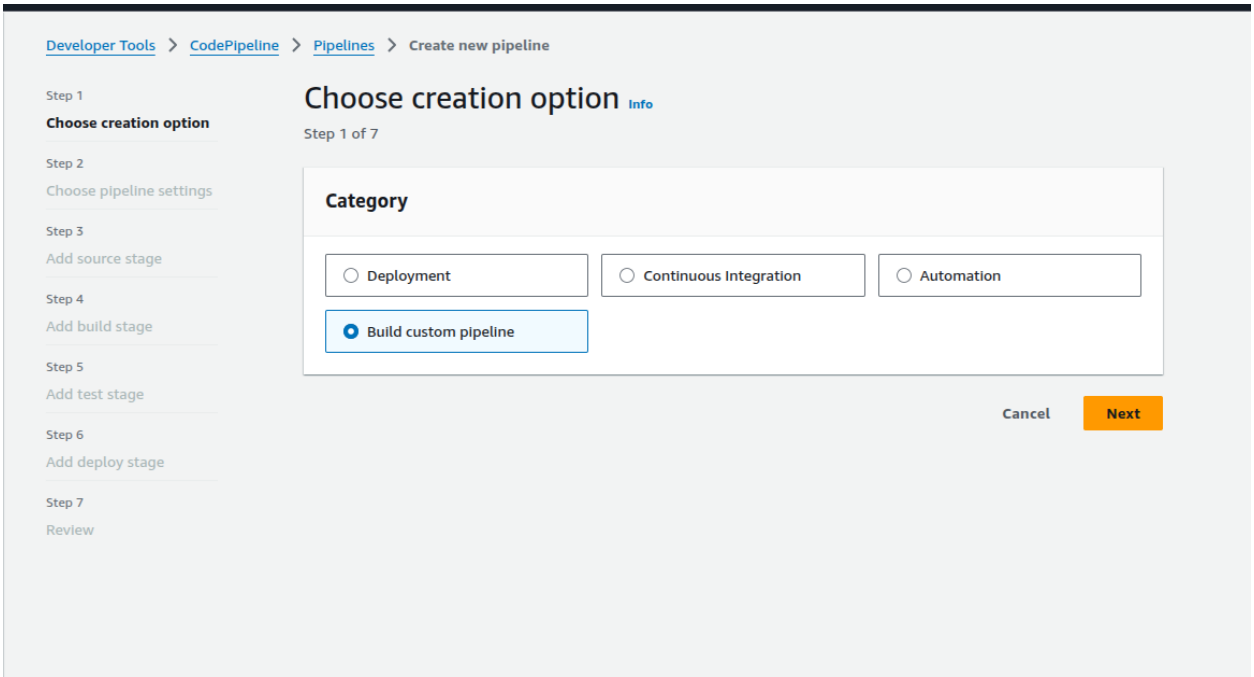
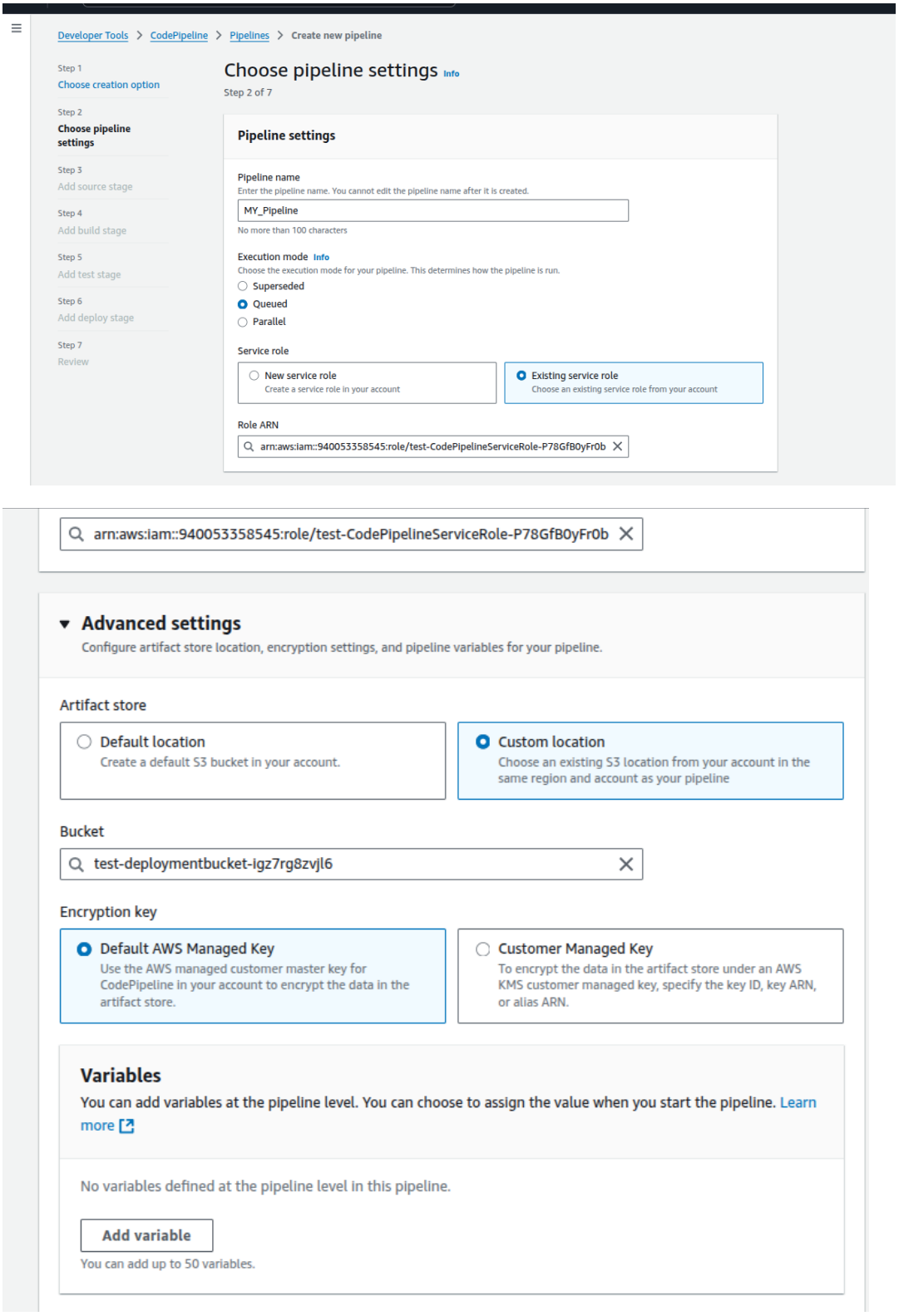
1. Create CodeStarConnection.

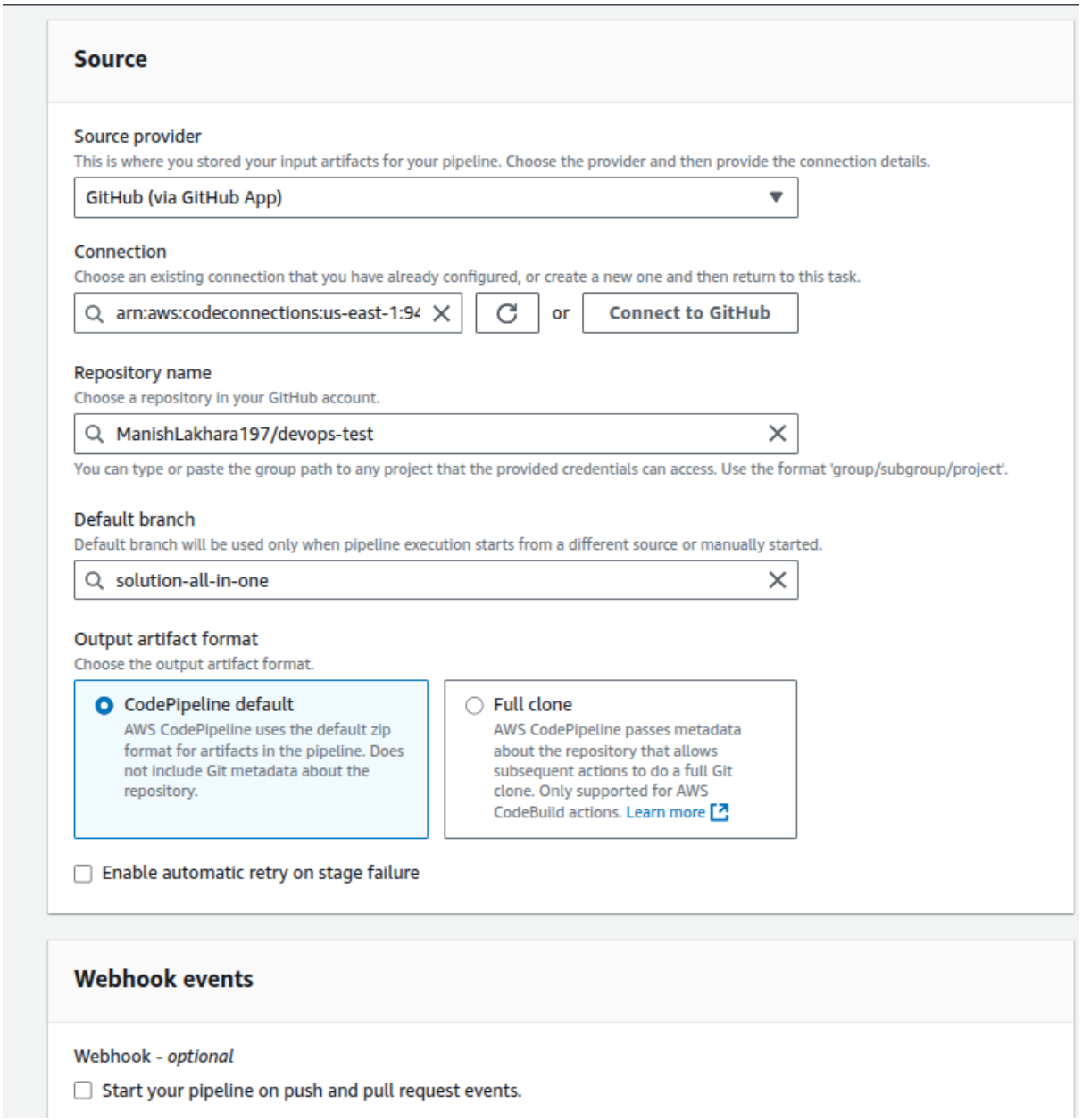


2. Create Pipeline

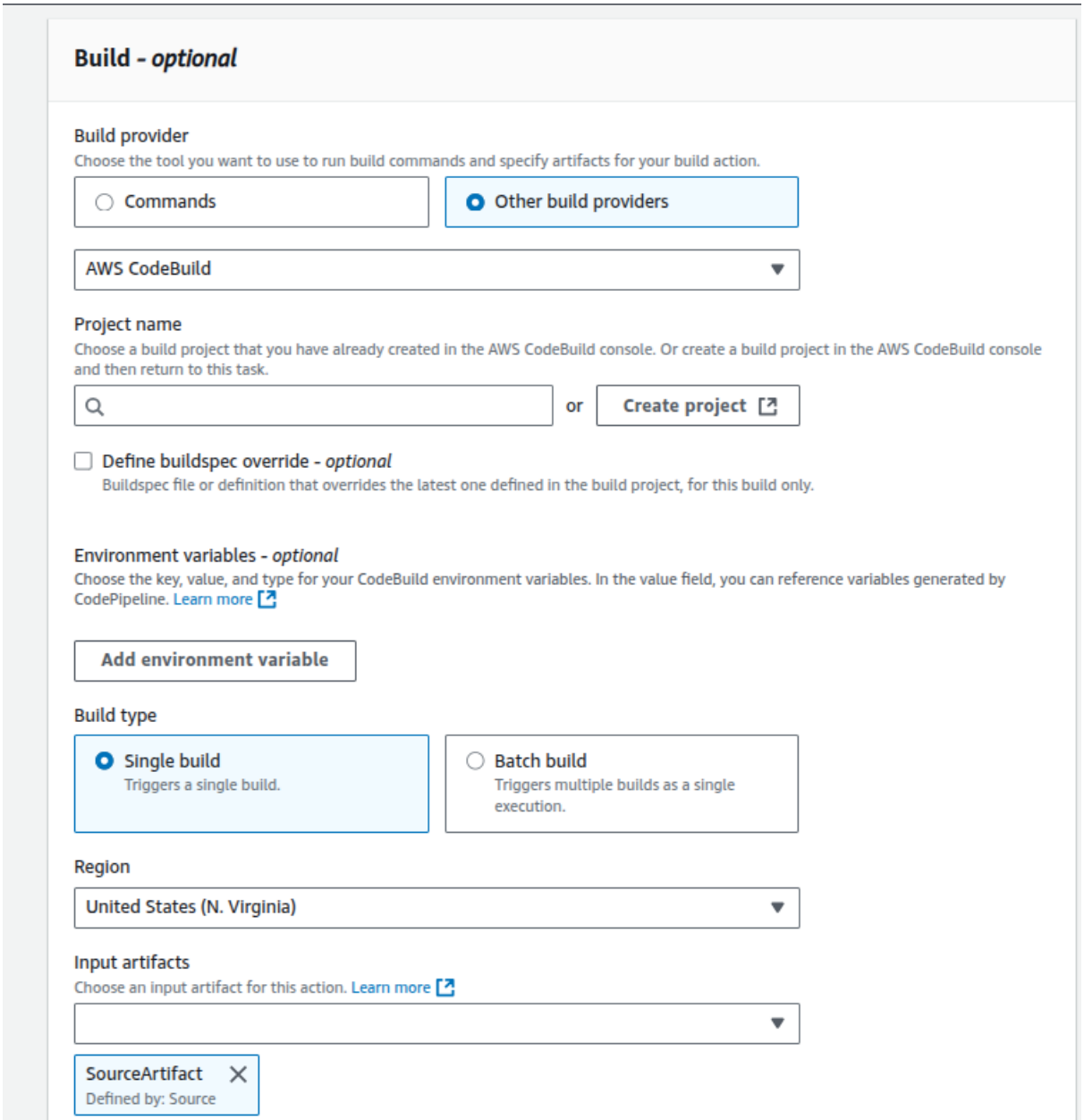




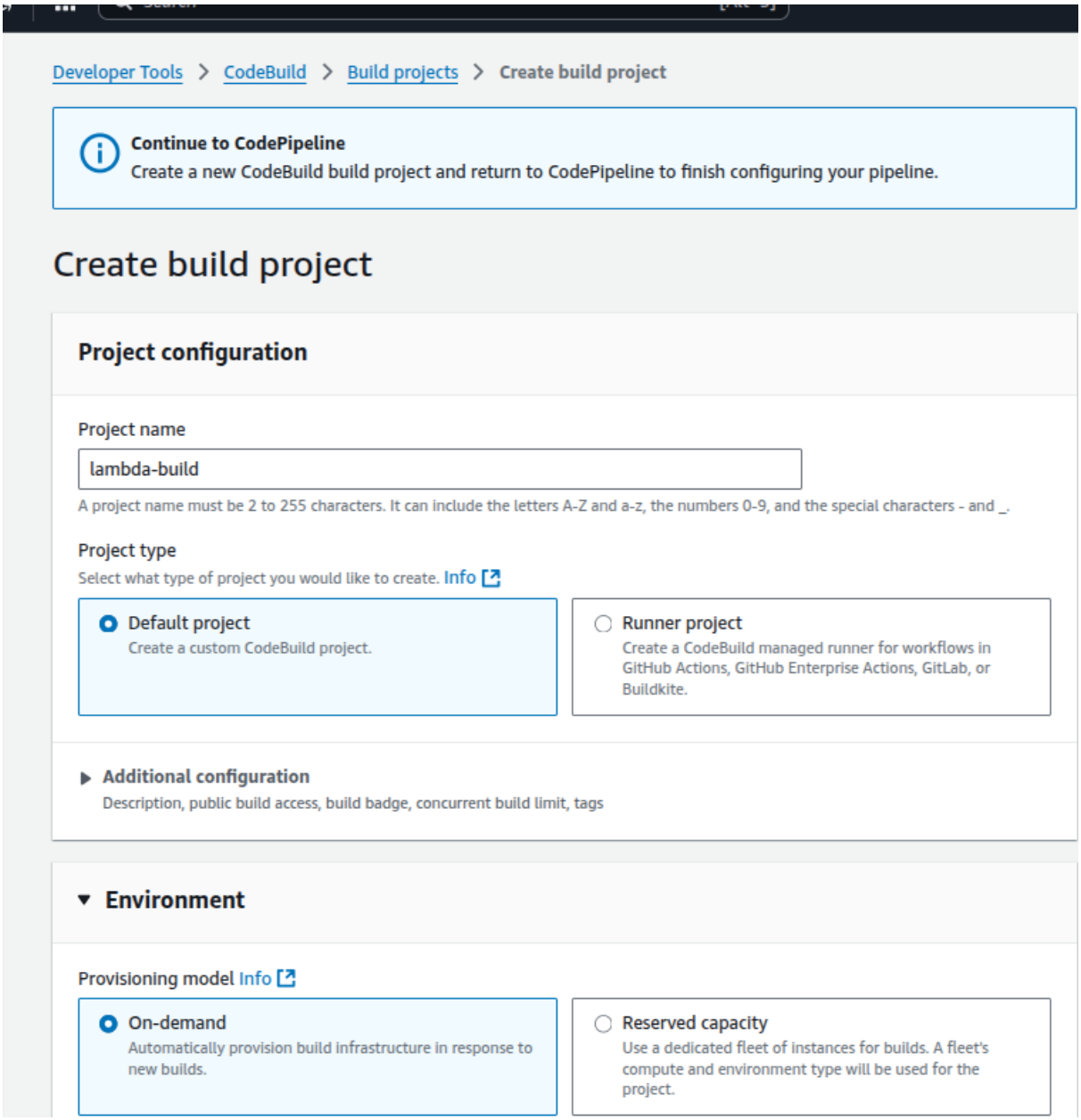
Make sure bucket and pipeline role is same as we created from infrastructure



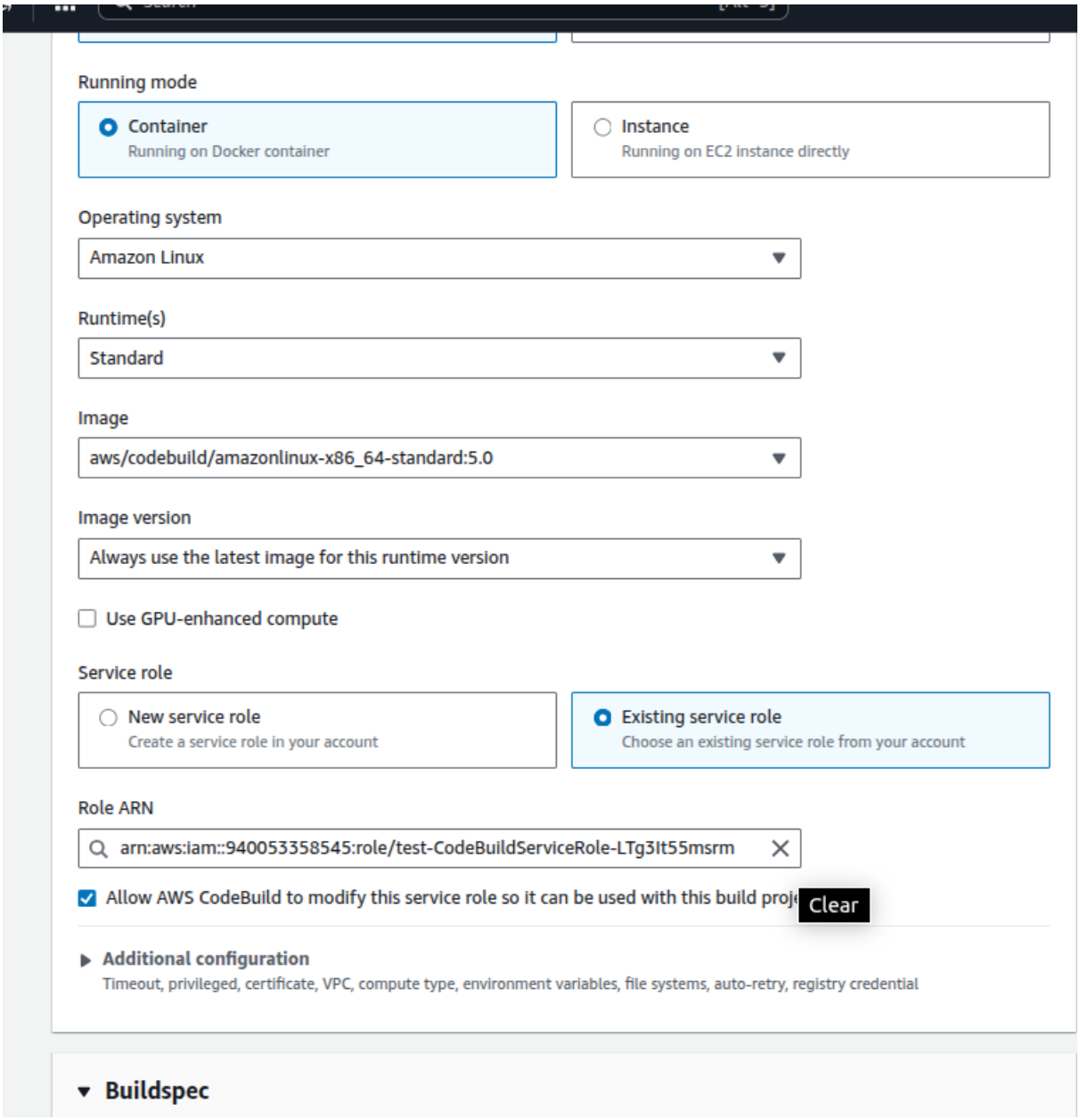
Here we use codestarconnection.

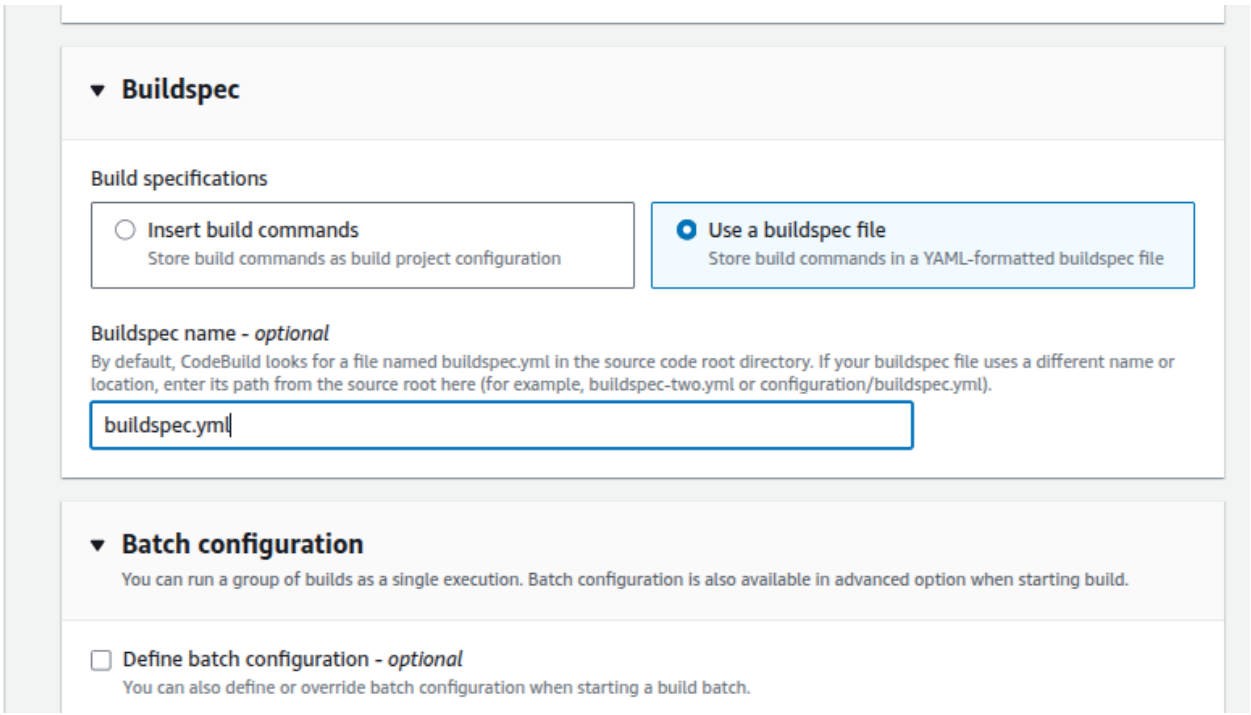


Here click on Create Project


Developer Tools > CodeBuild > Build projects > Create build project
Continue to CodePipeline
Create a new CodeBuild build project and return to CodePipeline to finish configuring your pipeline.
Create build project
Project configuration
Project name
lambda-build
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Project type
Select what type of project you would like to create. Info
Default project
Create a custom CodeBuild project.
Runner project
Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.
Additional configuration
Description, public build access, build badge, concurrent build limit, tags
Environment
Provisioning model Info
On-demand
Automatically provision build infrastructure in response to new builds.
Reserved capacity
Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

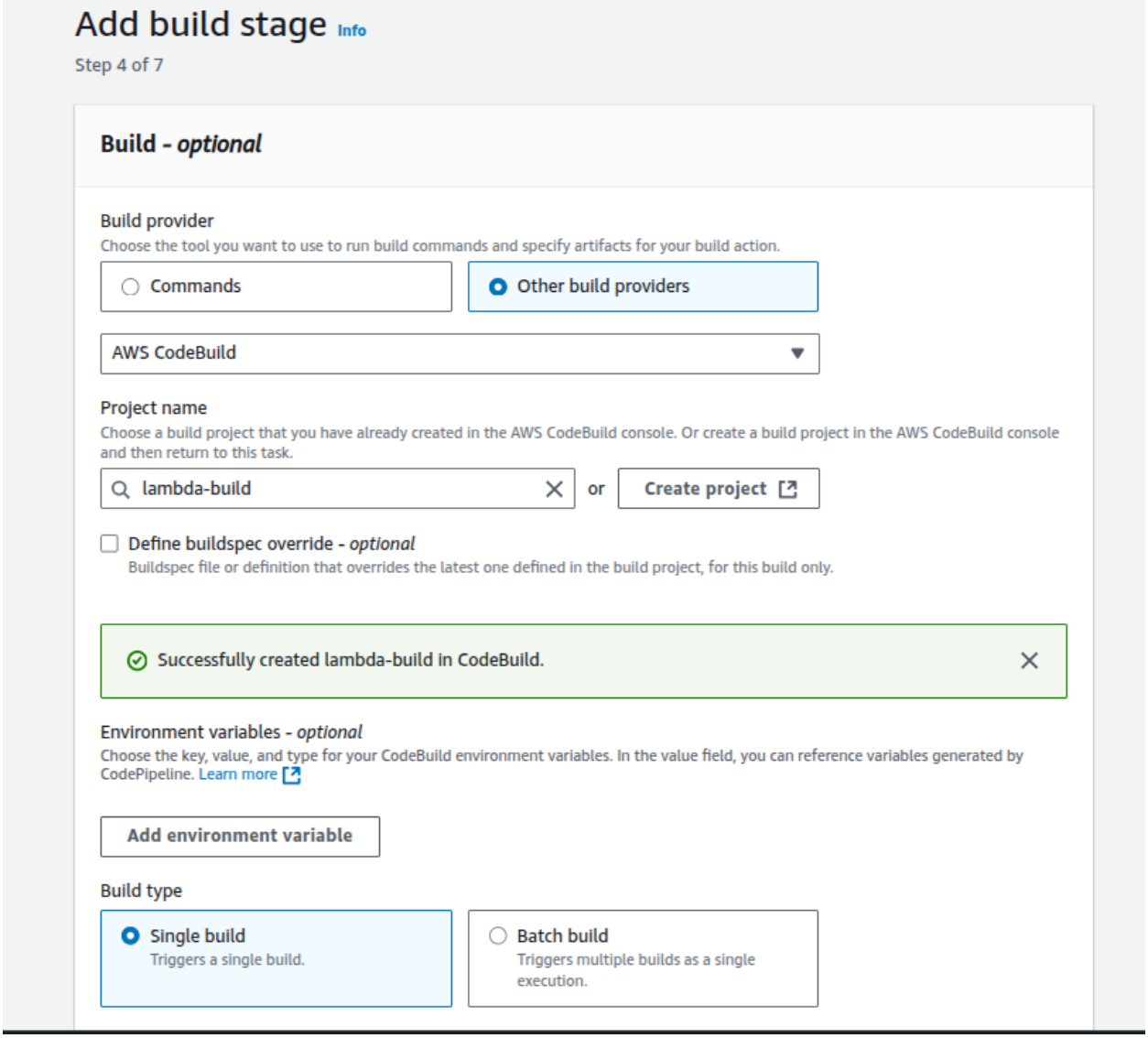
Then under enviroment section select Existing role





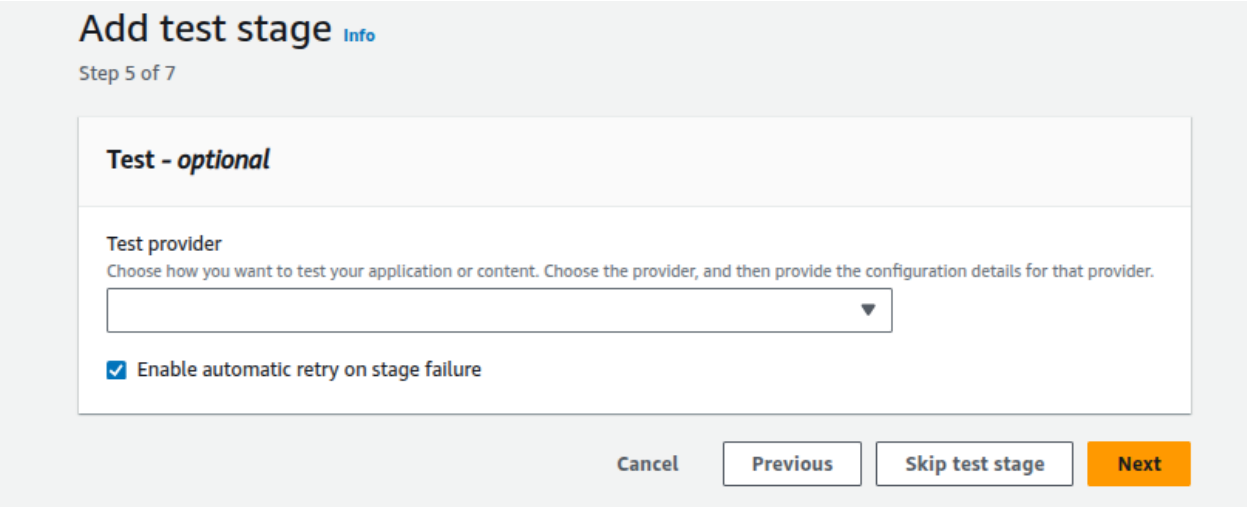
Then click on “Continue ..”

Back onto pipeline console >>

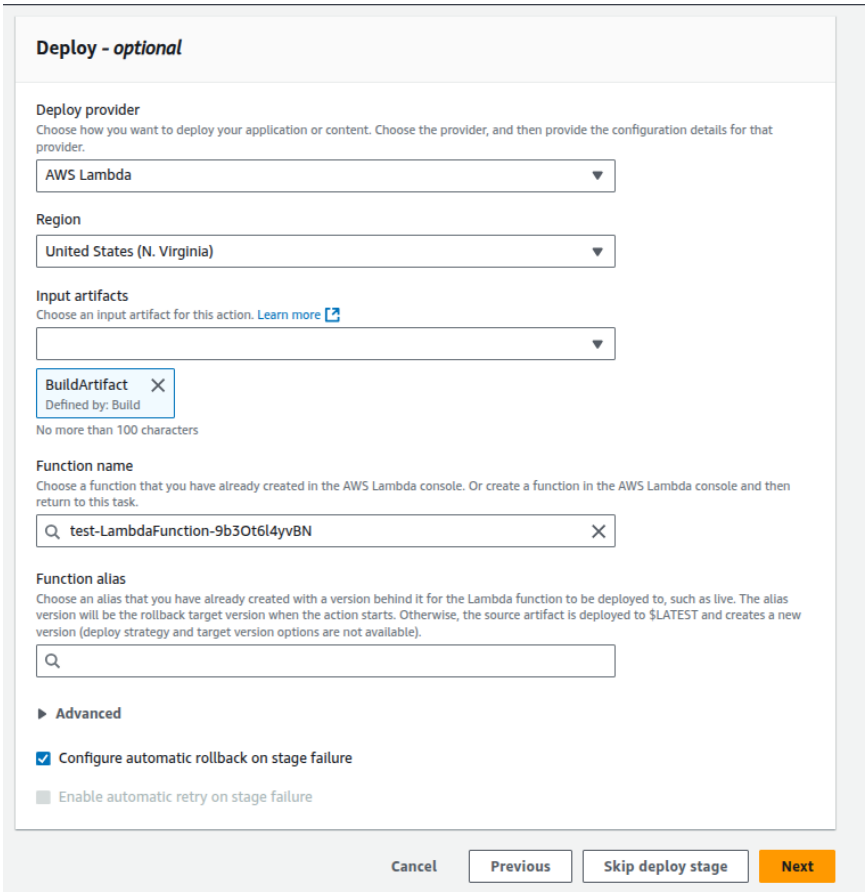


Then click on Next>>

Skip test >>

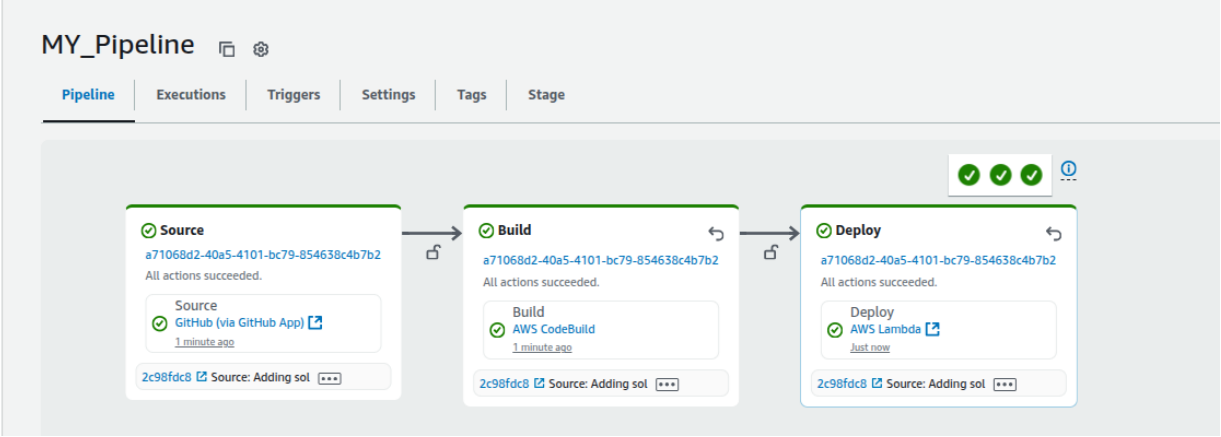


Then select lambda function to deploy

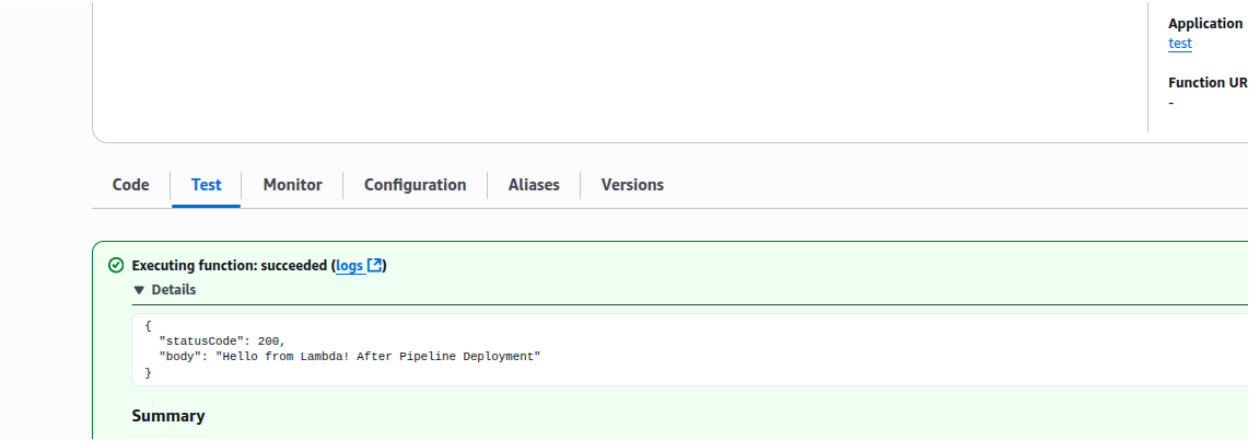


Then create Pipeline.

Get it deploy -->



Test lambda again -->



This complete Lambda Function deployment.

I have also added appspec.yml code that could be used to create CodeDeploy. This step is used to deploy application code to ec2.

You can skip creating that stage as AWS doesn't support codeDeploy in FreeTier. I have Update code just in case you need to view how its done.