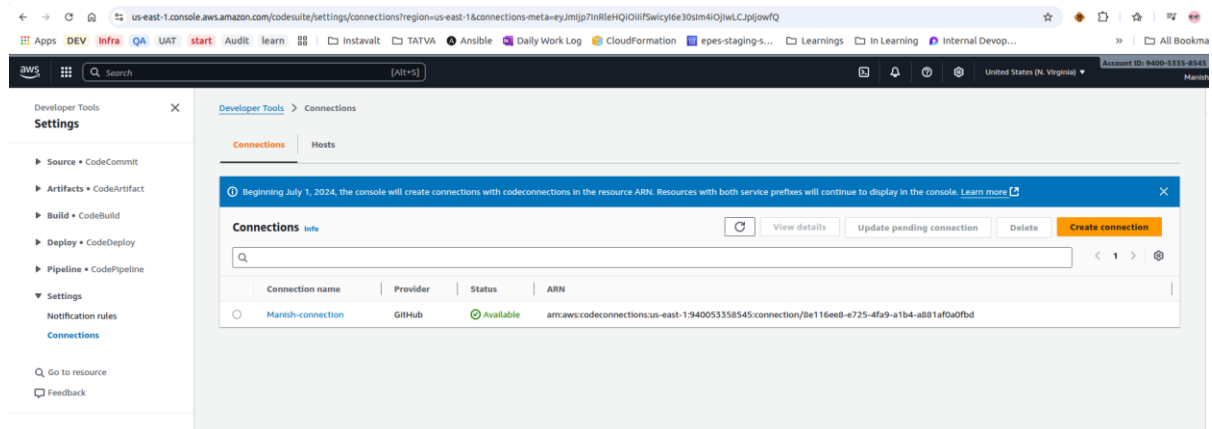
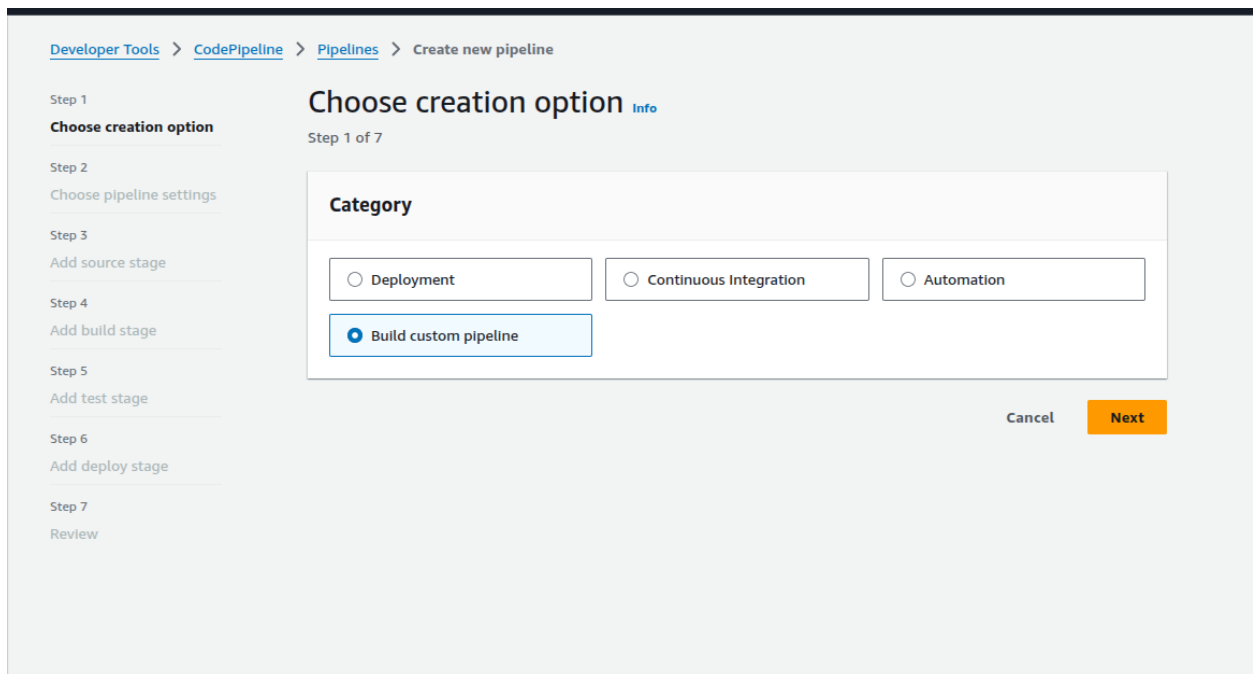


## 1. Create CodeStarConnection.



## 2. Create Pipeline



Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1  
Choose creation option

Step 2  
Choose pipeline settings

Step 3  
Add source stage

Step 4  
Add build stage

Step 5  
Add test stage

Step 6  
Add deploy stage

Step 7  
Review

## Choose pipeline settings [info](#)

Step 2 of 7

### Pipeline settings

**Pipeline name**  
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

**Execution mode** [info](#)  
Choose the execution mode for your pipeline. This determines how the pipeline is run.

☐ Superseded

☒ Queued

☐ Parallel

**Service role**

☐ New service role  
Create a service role in your account

☒ Existing service role  
Choose an existing service role from your account

**Role ARN**

 [X](#)

[X](#)

▼ **Advanced settings**

Configure artifact store location, encryption settings, and pipeline variables for your pipeline.

**Artifact store**

☐ Default location  
Create a default S3 bucket in your account.

☒ Custom location  
Choose an existing S3 location from your account in the same region and account as your pipeline

**Bucket**

 [X](#)

**Encryption key**

☒ Default AWS Managed Key  
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

☐ Customer Managed Key  
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

**Variables**

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. [Learn more](#) [🔗](#)

No variables defined at the pipeline level in this pipeline.

Add variable

You can add up to 50 variables.

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

## Source

### Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via GitHub App) ▼

### Connection

Choose an existing connection that you have already configured, or create a new one and then return to this task.

🔍  ✕



or

[Connect to GitHub](#)

### Repository name

Choose a repository in your GitHub account.

🔍  ✕

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.

🔍  ✕

### 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. [Learn more](#)

☐ Enable automatic retry on stage failure

## Webhook events

### Webhook - optional

☐ Start your pipeline on push and pull request events.

Here we use codestarconnection.

## Build - *optional*

### Build provider

Choose the tool you want to use to run build commands and specify artifacts for your build action.

☐ Commands

☒ Other build providers

AWS CodeBuild ▼

### Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.



or

Create project

### ☐ Define buildspec override - *optional*

Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

### Environment variables - *optional*

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

### Build type

☒ Single build

Triggers a single build.

☐ Batch build

Triggers multiple builds as a single execution.

### Region

United States (N. Virginia) ▼

### Input artifacts

Choose an input artifact for this action. [Learn more](#)

▼

SourceArtifact ✕

Defined by: Source

Here click on Create 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

Running mode

☒ Container  
Running on Docker container

☐ Instance  
Running on EC2 instance directly

Operating system

Amazon Linux ▼

Runtime(s)

Standard ▼

Image

aws/codebuild/amazonlinux-x86\_64-standard:5.0 ▼

Image version

Always use the latest image for this runtime version ▼

☐ Use GPU-enhanced compute

Service role

☐ New service role  
Create a service role in your account

☒ Existing service role  
Choose an existing service role from your account

Role ARN

🔍 arn:aws:iam::940053358545:role/test-CodeBuildServiceRole-LTg3It55msrm ✕

☒ Allow AWS CodeBuild to modify this service role so it can be used with this build project 

Clear

▶ Additional configuration

Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential

▼ Buildspec

## ▼ Buildspec

### Build specifications

☐ Insert build commands

Store build commands as build project configuration

☒ Use a buildspec file

Store build commands in a YAML-formatted buildspec file

### Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

## ▼ Batch configuration

You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.

☐ Define batch configuration - *optional*

You can also define or override batch configuration when starting a build batch.

Then click on “Continue ..”

Back onto pipeline console >>

## Add build stage [Info](#)

Step 4 of 7

### Build - optional

**Build provider**  
Choose the tool you want to use to run build commands and specify artifacts for your build action.

☐ Commands

☒ Other build providers

AWS CodeBuild

**Project name**  
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

Q lambda-build X

 or 

Create project

☐ **Define buildspec override - optional**  
Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

✔ Successfully created lambda-build in CodeBuild. X

**Environment variables - optional**  
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

**Build type**

☒ **Single build**  
Triggers a single build.

☐ **Batch build**  
Triggers multiple builds as a single execution.

Then click on Next>>



Skip test >>

## Add test stage [Info](#)

Step 5 of 7

### Test - optional

**Test provider**  
Choose how you want to test your application or content. Choose the provider, and then provide the configuration details for that provider.

☒ Enable automatic retry on stage failure

[Cancel](#) [Previous](#) [Skip test stage](#) [Next](#)

Then select lambda function to deploy

### Deploy - optional

**Deploy provider**  
Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for that provider.

**Region**

**Input artifacts**  
Choose an input artifact for this action. [Learn more](#)

**BuildArtifact** ✕  
Defined by: Build  
No more than 100 characters

**Function name**  
Choose a function that you have already created in the AWS Lambda console. Or create a function in the AWS Lambda console and then return to this task.

**Function alias**  
Choose an alias that you have already created with a version behind it for the Lambda function to be deployed to, such as live. The alias version will be the rollback target version when the action starts. Otherwise, the source artifact is deployed to \$LATEST and creates a new version (deploy strategy and target version options are not available).

**Advanced**

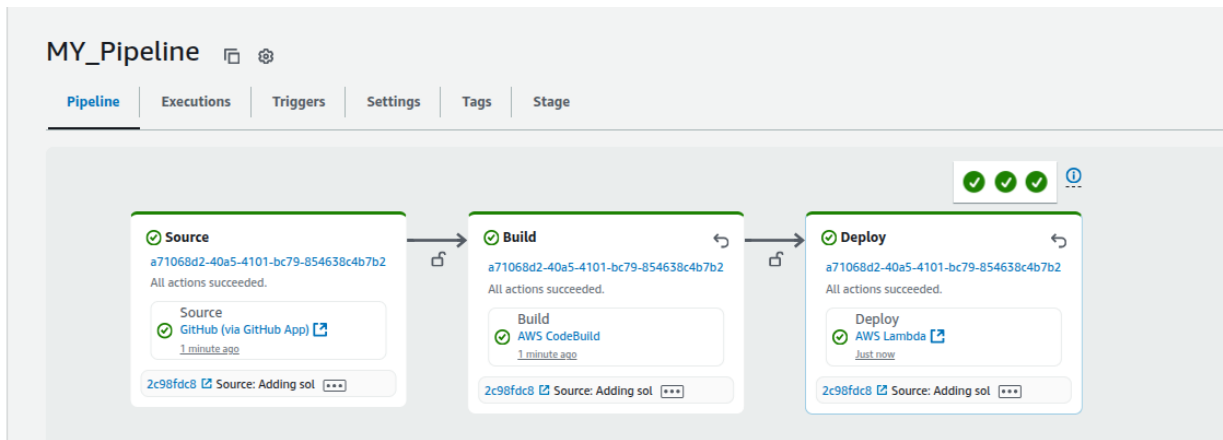
☒ Configure automatic rollback on stage failure

☐ Enable automatic retry on stage failure

[Cancel](#) [Previous](#) [Skip deploy stage](#) [Next](#)

Then create Pipeline.

Get it deploy -->



Test lambda again -->

The screenshot shows the AWS Lambda console for a function. The 'Test' tab is selected, showing a successful test execution. The details section displays the following JSON response:

```
{
  "statusCode": 200,
  "body": "Hello from Lambda! After Pipeline Deployment"
}
```

The summary section is also visible.

On the right side of the console, the 'Application' is listed as 'test' and the 'Function UR' is shown as a redacted string.

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.