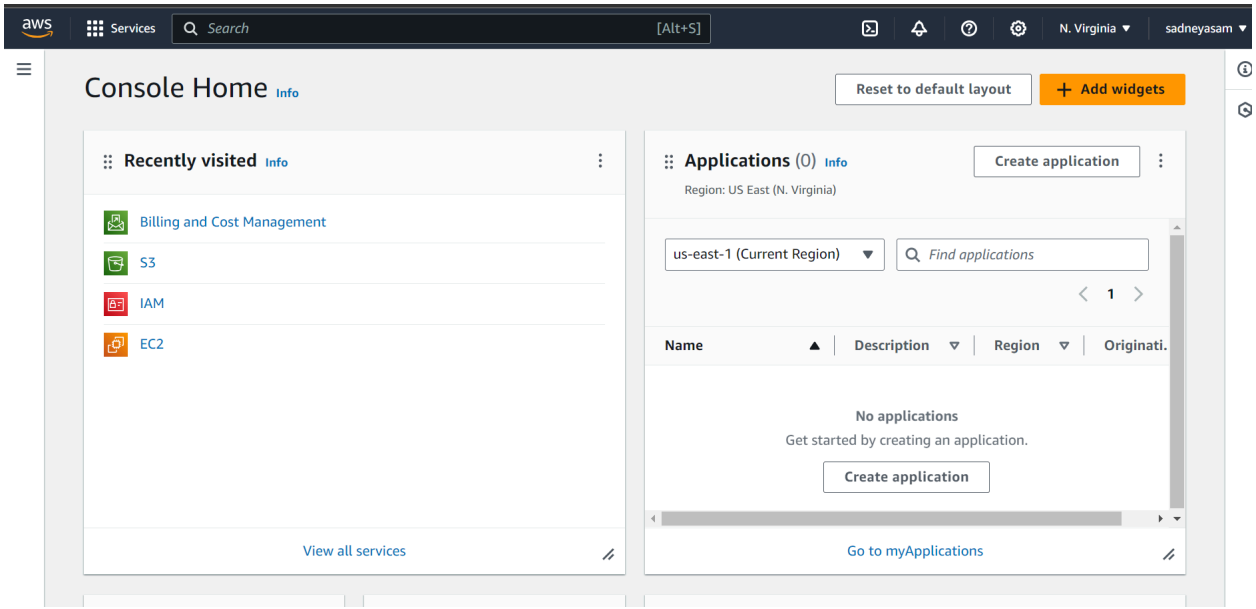
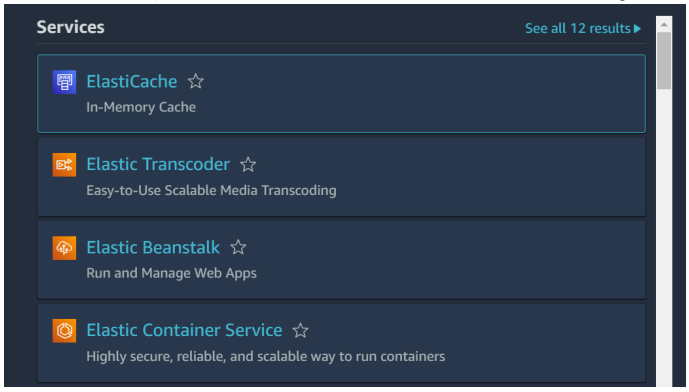
Experiment 2

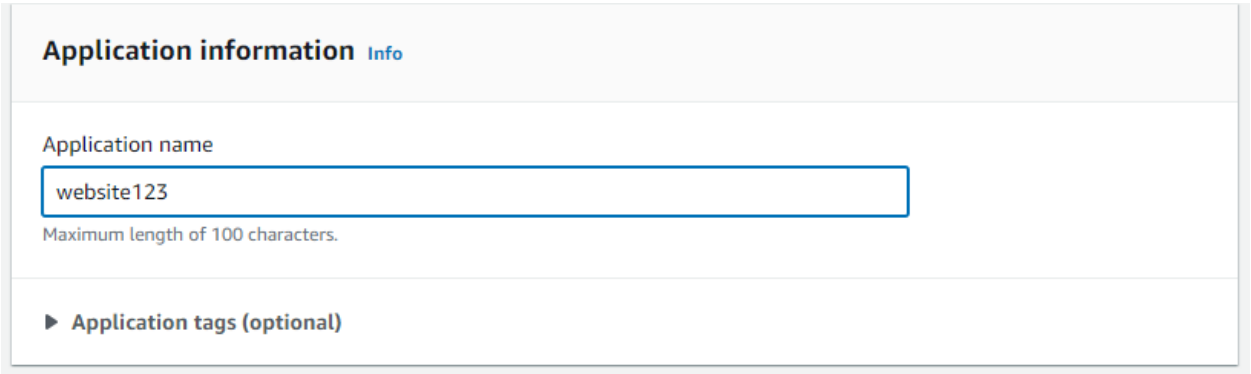
1. Go to the amazon console page

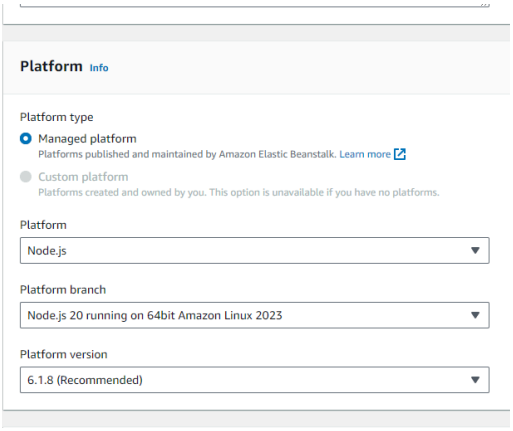


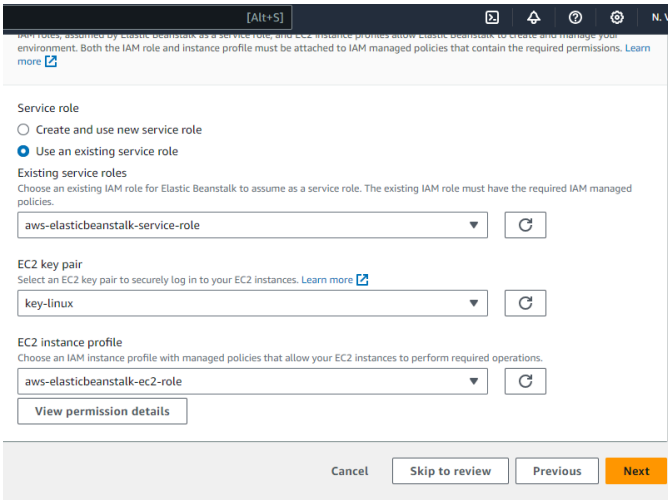
2. Open up Elastic Beanstalk and name your web app.



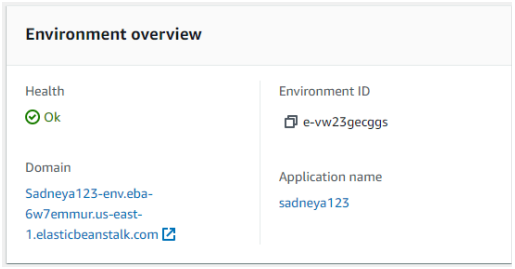
3. Enter your website and its information



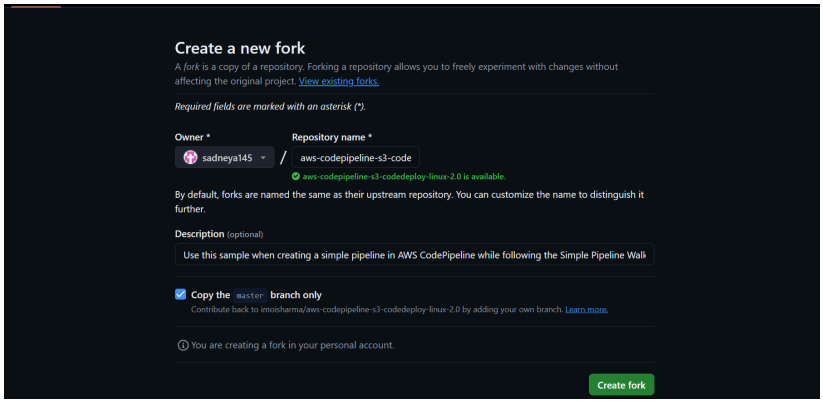




4. Your environment is created



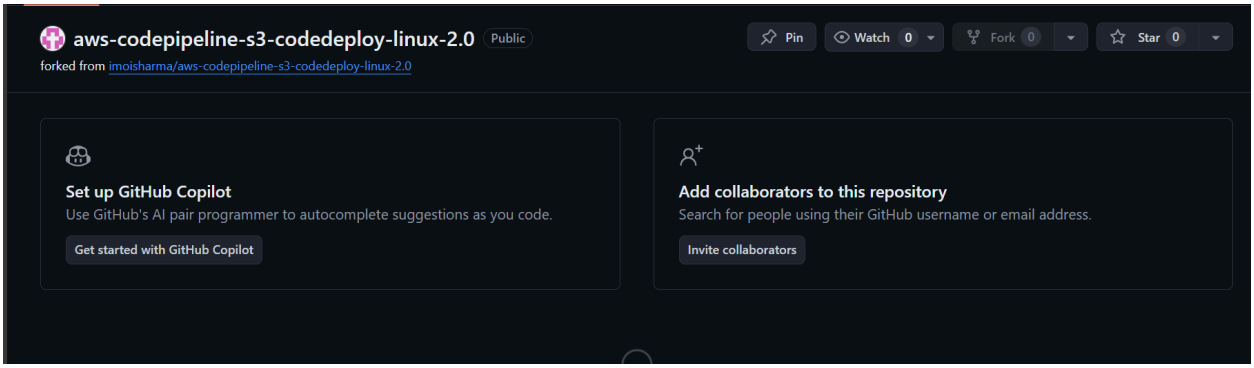
Step 2: Get a copy of your sample code



In this step, we will get the sample code from this GitHub Repository to host it later. The pipeline takes code from the source and then performs actions on it.

We will use this forked GitHub repository as a source for this experiment. We can alternatively also use Amazon S3 and AWS CodeCommit.

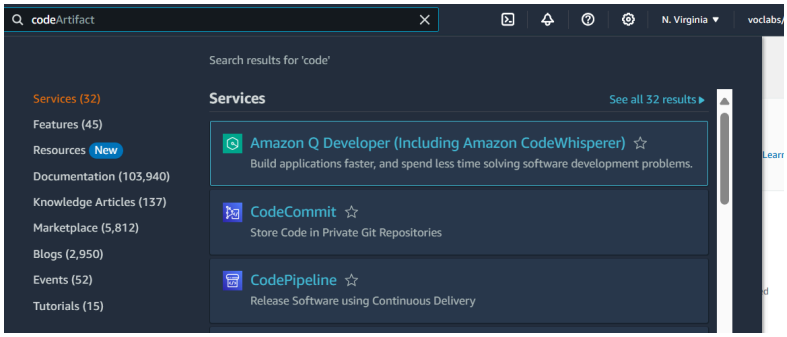
Go to the repository shared above and simply fork it.

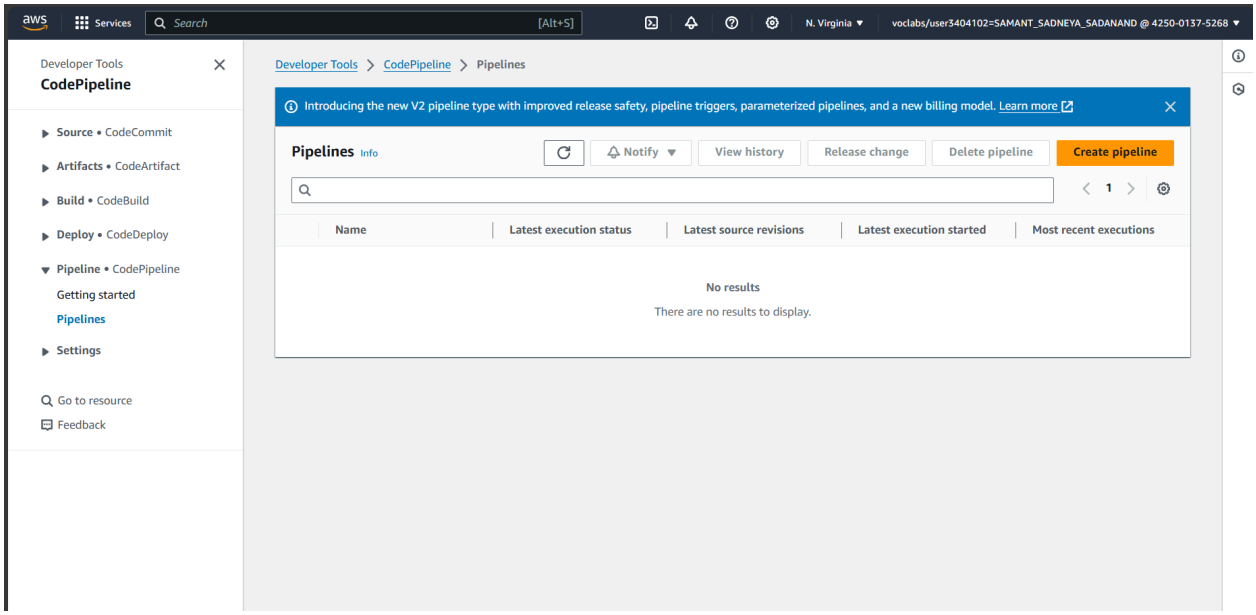


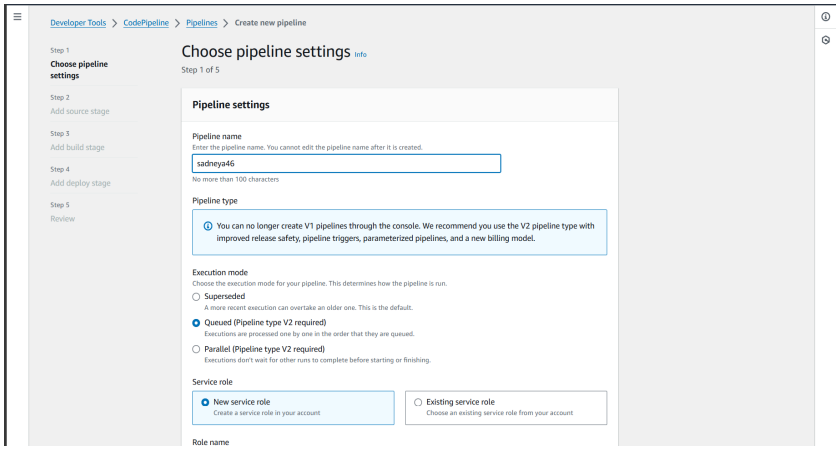
Step 3: Creating a CodePipeline

In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

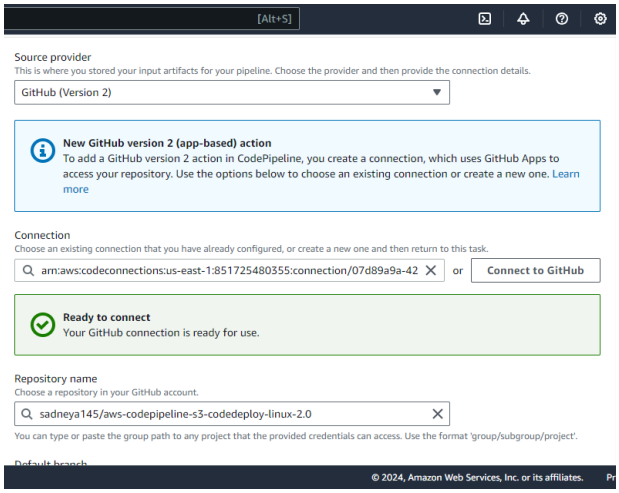
1. Go to AWS Developer Tools -> CodePipeline and create a new Pipeline. Fill in the initial settings first.







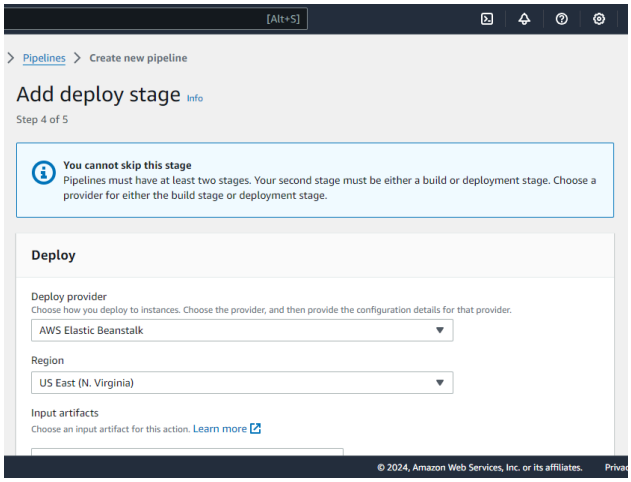
2. In the source stage, choose GitHub v1 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

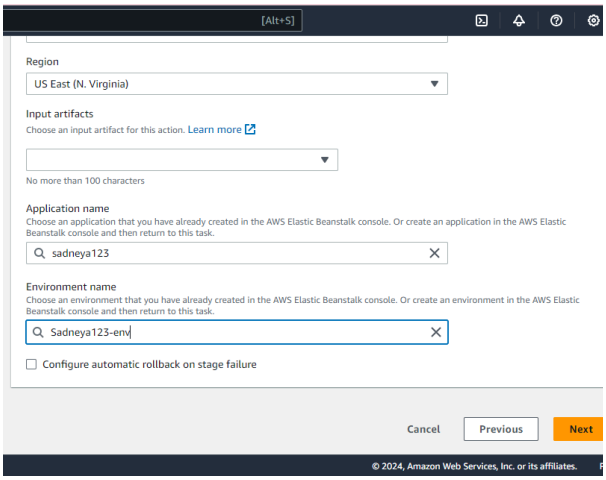


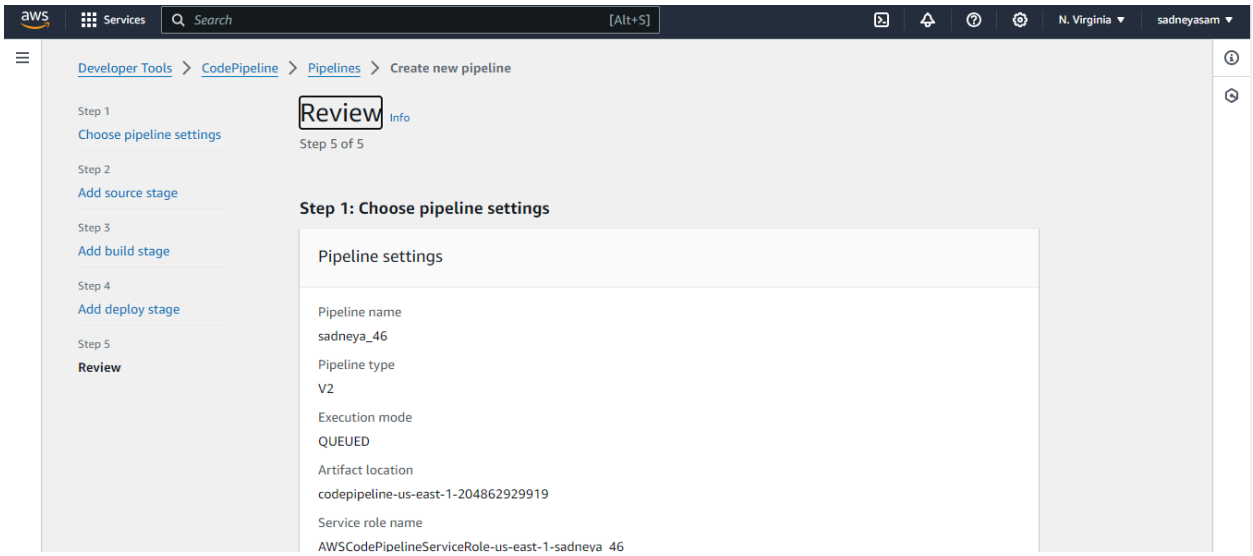
3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

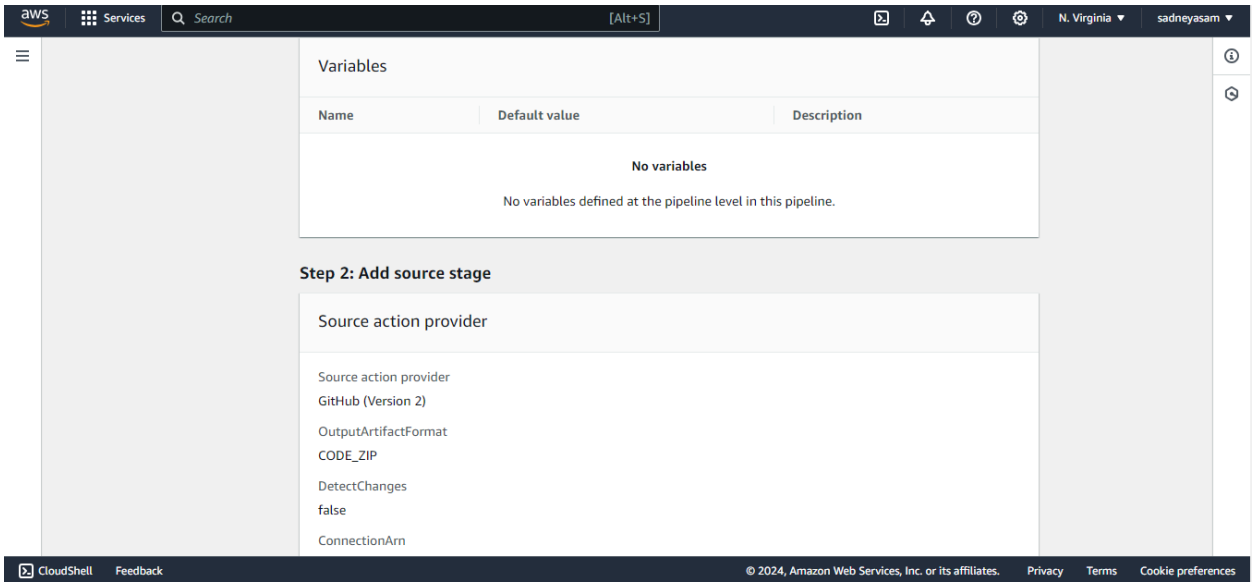
Step 4: Deployment

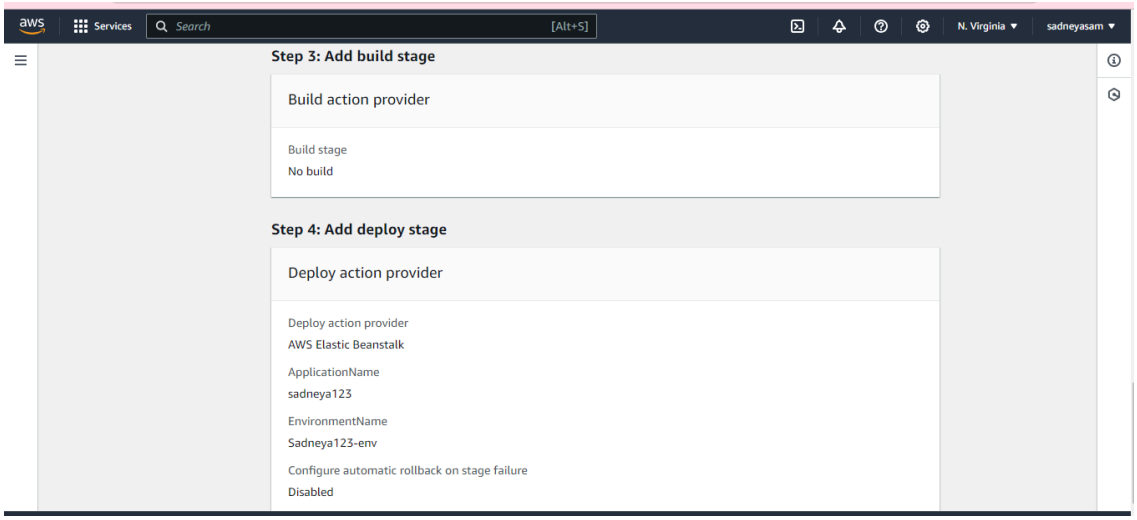
1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.

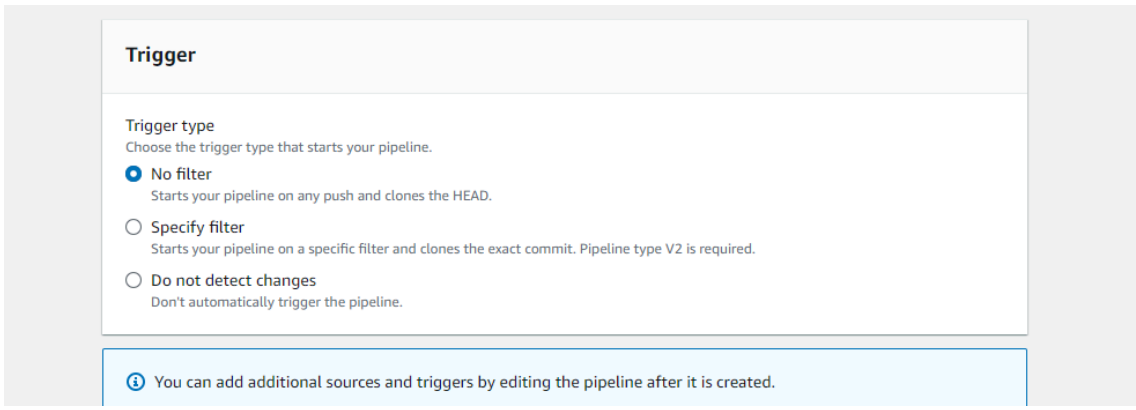

[Alt+S]
Pipelines > Create new pipeline
Add deploy stage Info
Step 4 of 5
You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.
Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk
Region
US East (N. Virginia)
Input artifacts
Choose an input artifact for this action. Learn more
© 2024, Amazon Web Services, Inc. or its affiliates.

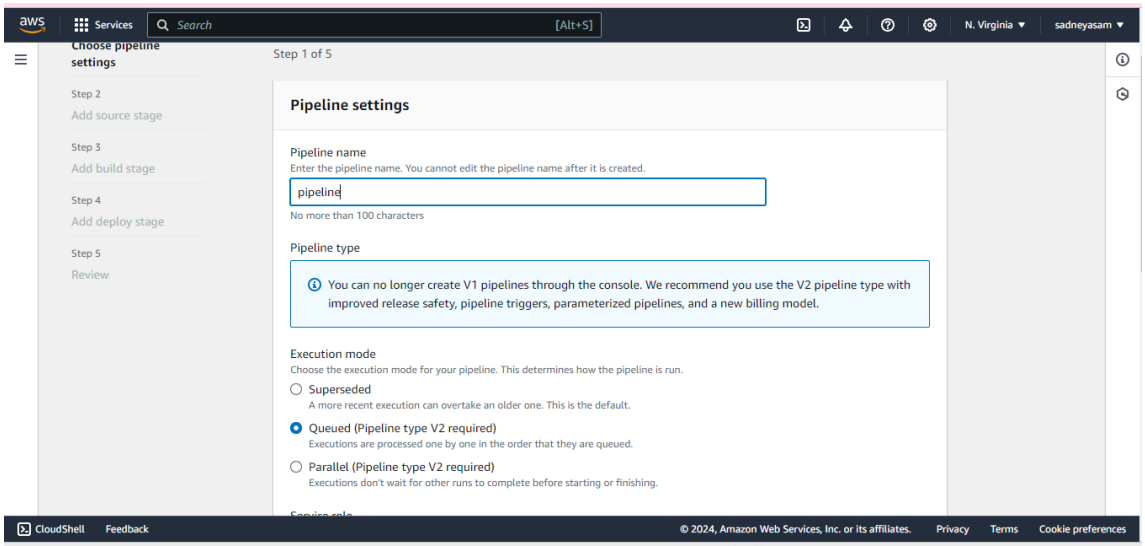

[Alt+S]
Region
US East (N. Virginia)
Input artifacts
Choose an input artifact for this action. Learn more
No more than 100 characters
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
sadneya123
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
Sadneya123-env
Configure automatic rollback on stage failure
Cancel
Previous
Next
© 2024, Amazon Web Services, Inc. or its affiliates.


aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
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
Review Info
Step 5 of 5
Step 1: Choose pipeline settings
Pipeline settings
Pipeline name
sadneya_46
Pipeline type
V2
Execution mode
QUEUED
Artifact location
codepipeline-us-east-1-204862929919
Service role name
AWSCodePipelineServiceRole-us-east-1-sadneya_46

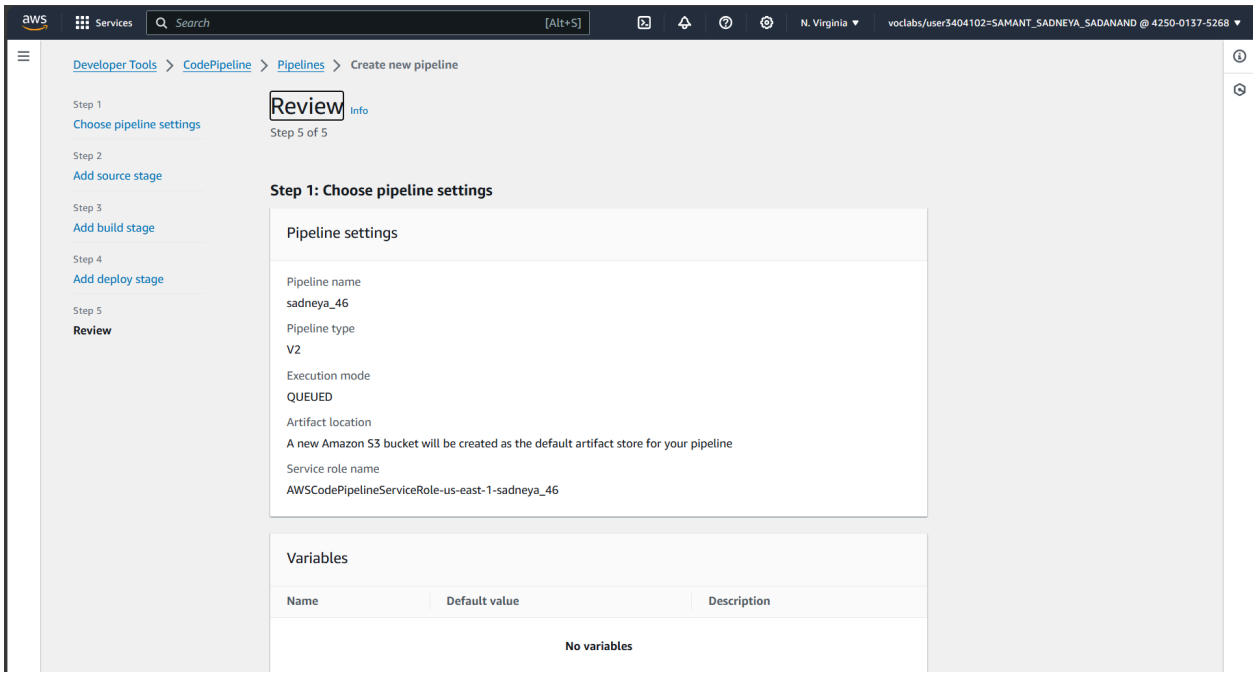

aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Variables
Name
Default value
Description
No variables
No variables defined at the pipeline level in this pipeline.
Step 2: Add source stage
Source action provider
Source action provider
GitHub (Version 2)
OutputArtifactFormat
CODE_ZIP
DetectChanges
false
ConnectionArn
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

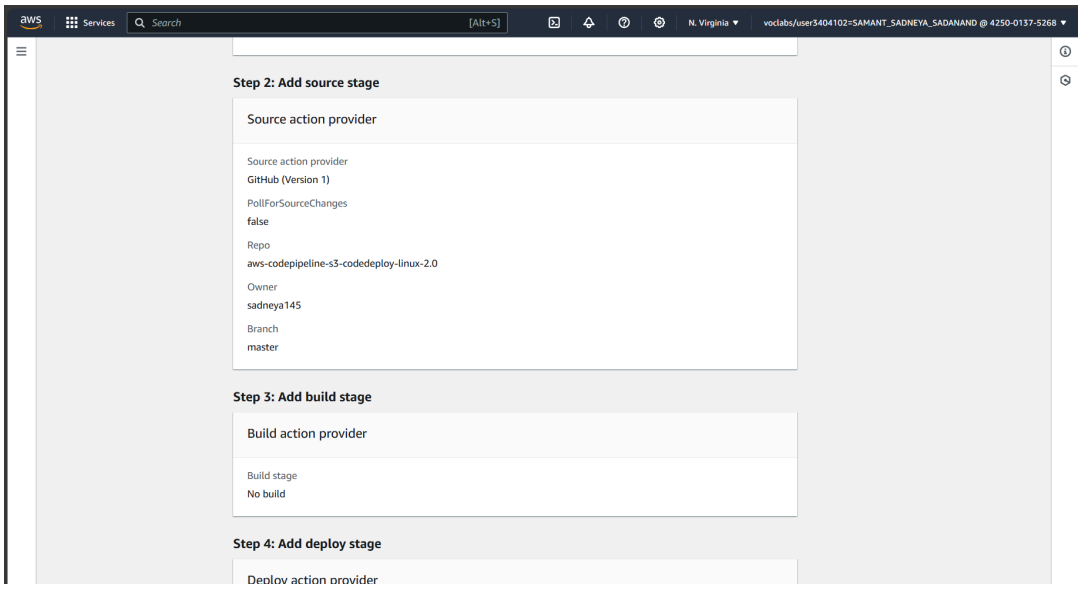

aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Step 3: Add build stage
Build action provider
Build stage
No build
Step 4: Add deploy stage
Deploy action provider
Deploy action provider
AWS Elastic Beanstalk
ApplicationName
sadneya123
EnvironmentName
Sadneya123-env
Configure automatic rollback on stage failure
Disabled

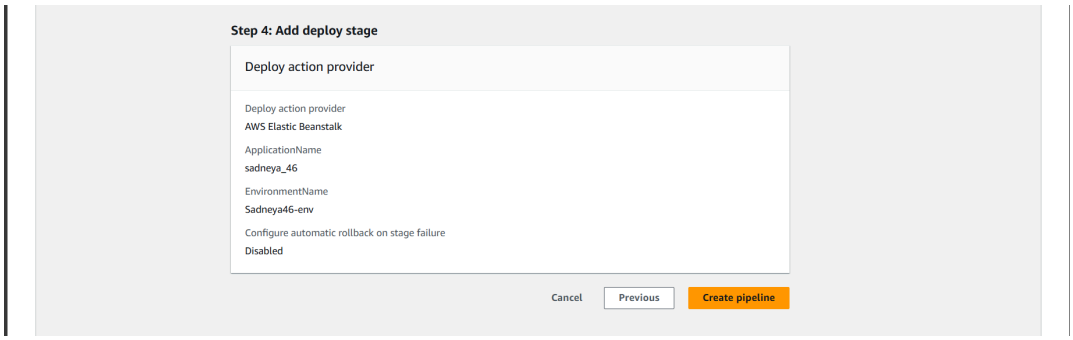

Trigger
Trigger type
Choose the trigger type that starts your pipeline.
No filter
Starts your pipeline on any push and clones the HEAD.
Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.
Do not detect changes
Don't automatically trigger the pipeline.
You can add additional sources and triggers by editing the pipeline after it is created.


aws
Services
Search
[Alt+S]
N. Virginia
sadneyasam
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Step 1 of 5
Pipeline settings
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
pipeline
No more than 100 characters
Pipeline type
You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.
Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.
Superseded
A more recent execution can overtake an older one. This is the default.
Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.
Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

2. Review all the settings and click on create pipeline







3.in the end, you can see that the pipeline has been deployed successfully