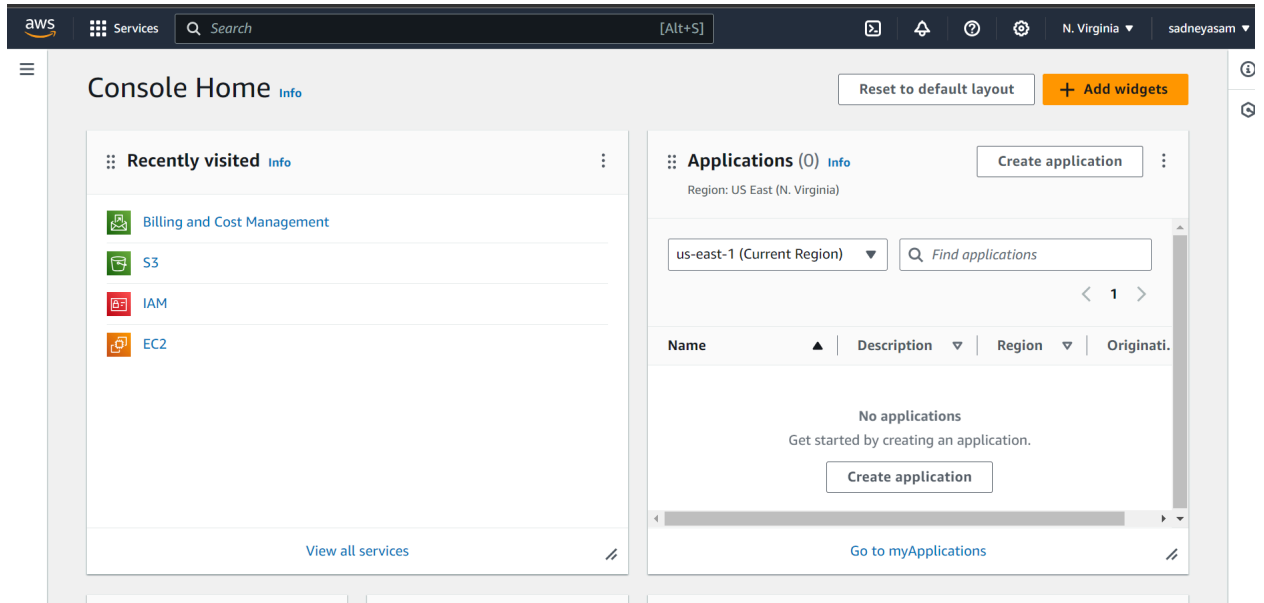
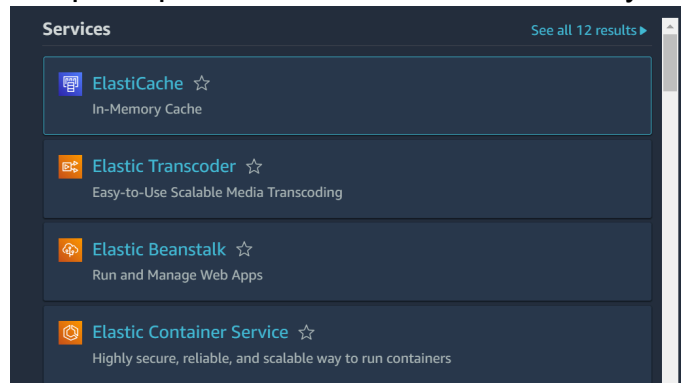


## Experiment 2

### 1. Go to the amazon console page



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



### 3. Enter your website and its information

#### Application information [Info](#)

Application name

Maximum length of 100 characters.

► Application tags (optional)

Platform

Info

Platform type

☒ Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ Custom platform

Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Node.js

Platform branch

Node.js 20 running on 64bit Amazon Linux 2023

Platform version

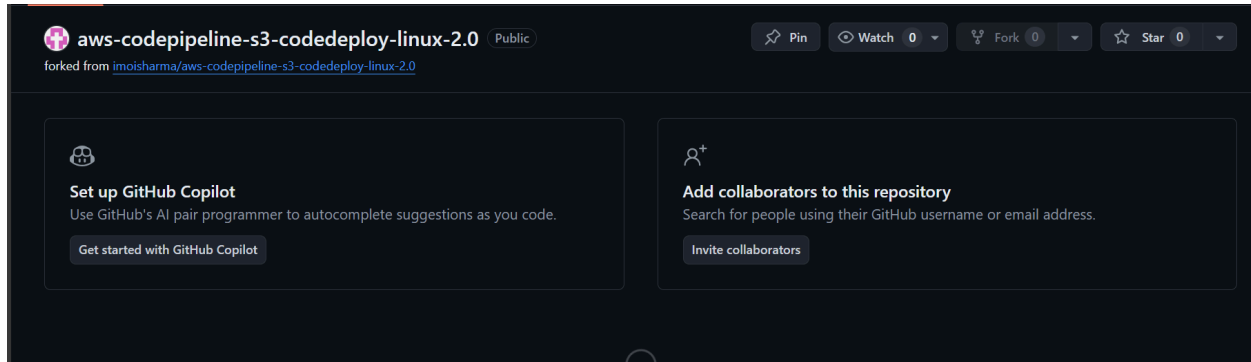
6.1.8 (Recommended)

#### 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.

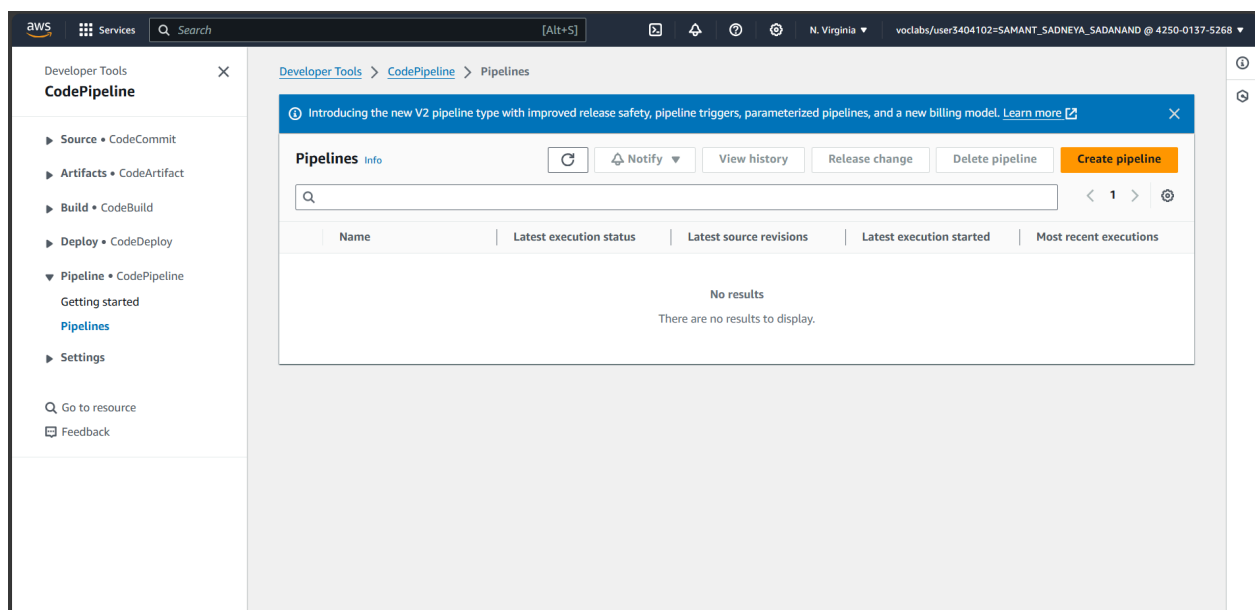
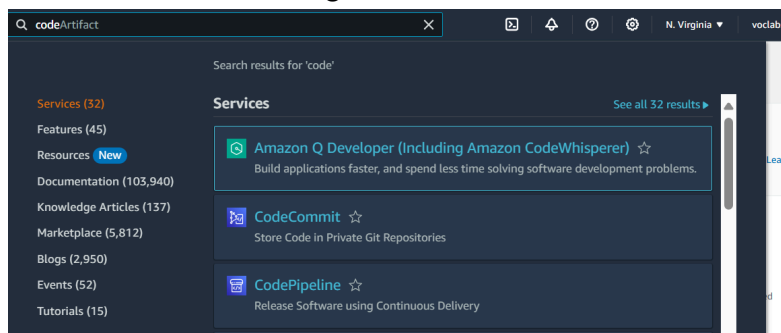
Go to the repository shared above and simply fork it.



### 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.



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

### Choose pipeline settings

Step 1 of 5

**Pipeline settings**

Pipeline name  
Enter the pipeline name. You cannot edit the pipeline name after it is created.  
sadneya46  
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.  
Executions are processed one by one in the order that they are queued.

☒ Queued (Pipeline type V2 required)  
Executions don't wait for other runs to complete before starting or finishing.

☐ Parallel (Pipeline type V2 required)

**Service role**

☒ New service role  
Create a service role in your account

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

Role name

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.

[Alt+S]

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.

am:aws:codeconnections:us-east-1:851725480355:connection/07d89a9a-42 X or [Connect to GitHub](#)

**Ready to connect**  
Your GitHub connection is ready for use.

**Repository name**  
Choose a repository in your GitHub account.

sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0 X

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default browser © 2024, Amazon Web Services, Inc. or its affiliates. In

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](#)

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.

Privacy

[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.

Privacy

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

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]

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

awsServicesSearch[Alt+S]

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.

awsServicesSearch[Alt+S]

Choose pipeline settings

Step 1 of 5

Step 2

Add source stage

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

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.

CloudShellFeedback

© 2024, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

## 2. Review all the settings and click on create pipeline

The screenshot shows the AWS CodePipeline console with the 'Review' step selected. The left sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main content area is titled 'Step 1: Choose pipeline settings' and contains two sections: 'Pipeline settings' and 'Variables'.

**Pipeline settings**

- Pipeline name: sadneya\_46
- Pipeline type: V2
- Execution mode: QUEUED
- Artifact location: A new Amazon S3 bucket will be created as the default artifact store for your pipeline
- Service role name: AWSCodePipelineServiceRole-us-east-1-sadneya\_46

**Variables**

Name	Default value	Description
No variables		

The screenshot shows the AWS CodePipeline console with the 'Add source stage', 'Add build stage', and 'Add deploy stage' steps. The left sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), and Step 4 (Add deploy stage). The main content area is titled 'Step 2: Add source stage', 'Step 3: Add build stage', and 'Step 4: Add deploy stage'.

**Step 2: Add source stage**

Source action provider

Source action provider: GitHub (Version 1)

PollForSourceChanges: false

Repo: aws-codepipeline-s3-codedeploy-linux-2.0

Owner: sadneya145

Branch: master

**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: sadneya\_46

EnvironmentName: Sadneya46-env

Configure automatic rollback on stage failure: Disabled


Buttons: Cancel, Previous, Create pipeline


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


 **Source** Succeeded

Pipeline execution ID: [bbb2a722-1bdb-4e0d-a7c7-a9cd0eb6613c](#)


Source

[GitHub \(Version 1\)](#) 

 Succeeded - *Just now*

[8fd5da54](#) 

View details

[8fd5da54](#)  Source: Update README.md