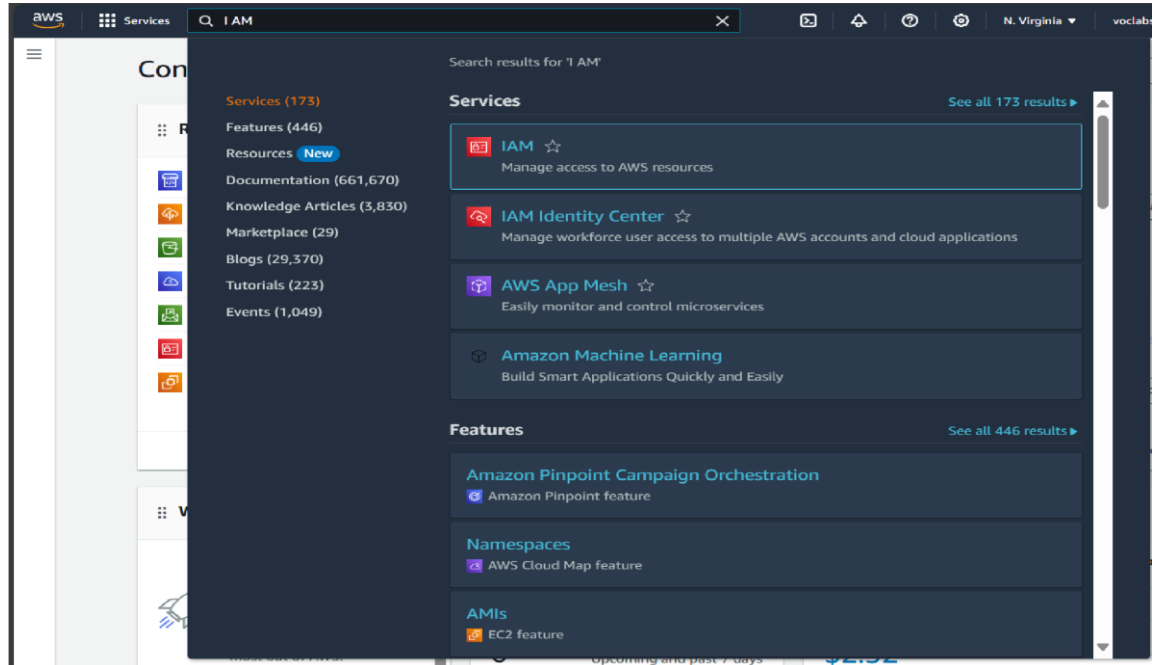


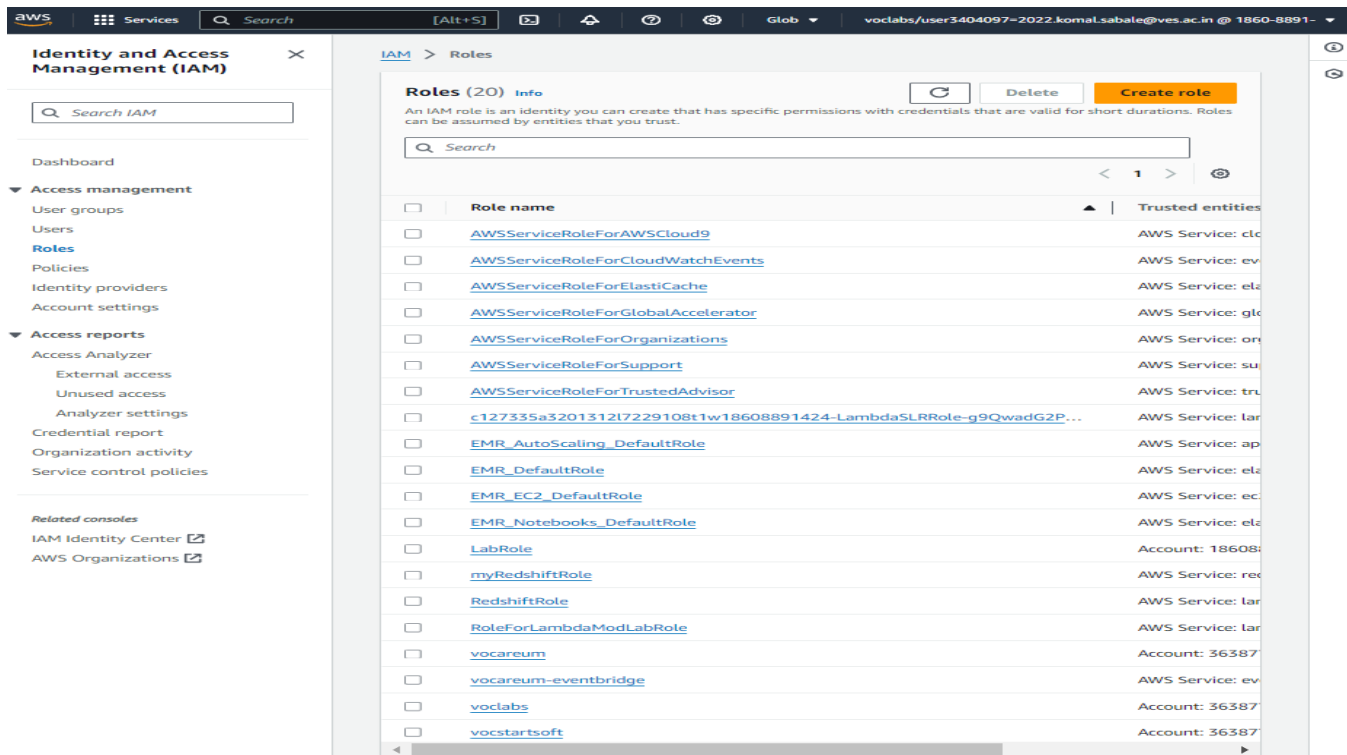
## Experiment No: 2

### Step1:- Creation of role:-

1. Login to your AWS account and search for IAM



2. Then go into the role section and click on create role.



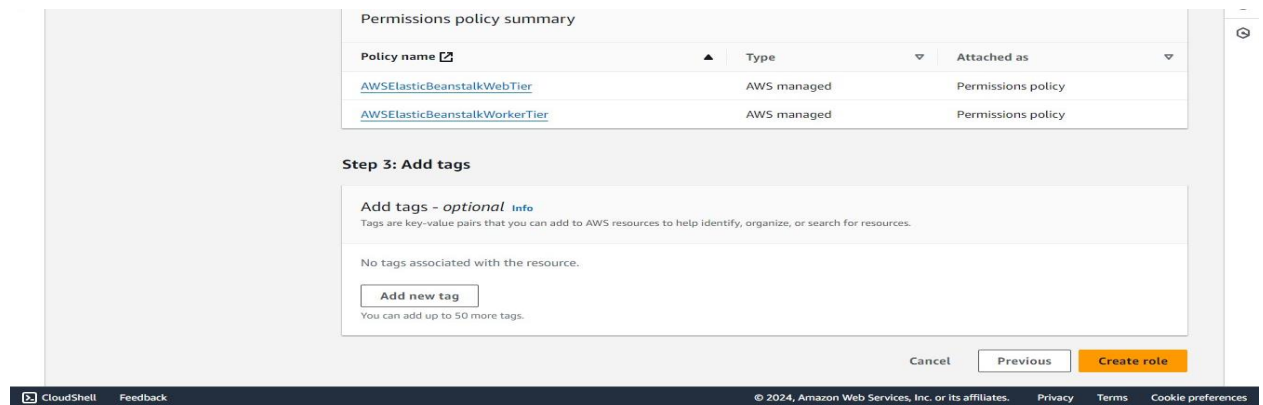
- Then select a trusted entity as AWS service.

The screenshot shows the AWS IAM console interface for creating a role. The breadcrumb navigation is 'IAM > Roles > Create role'. On the left, a sidebar shows the progress: Step 1 'Select trusted entity' (active), Step 2 'Add permissions', and Step 3 'Name, review, and create'. The main content area is titled 'Select trusted entity' with an 'Info' link. Under 'Trusted entity type', there are five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Each option has a brief description. Below this, the 'Use case' section is visible, with a description 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' and a dropdown menu labeled 'Service or use case' with the placeholder text 'Choose a service or use case'. At the bottom right are 'Cancel' and 'Next' buttons.

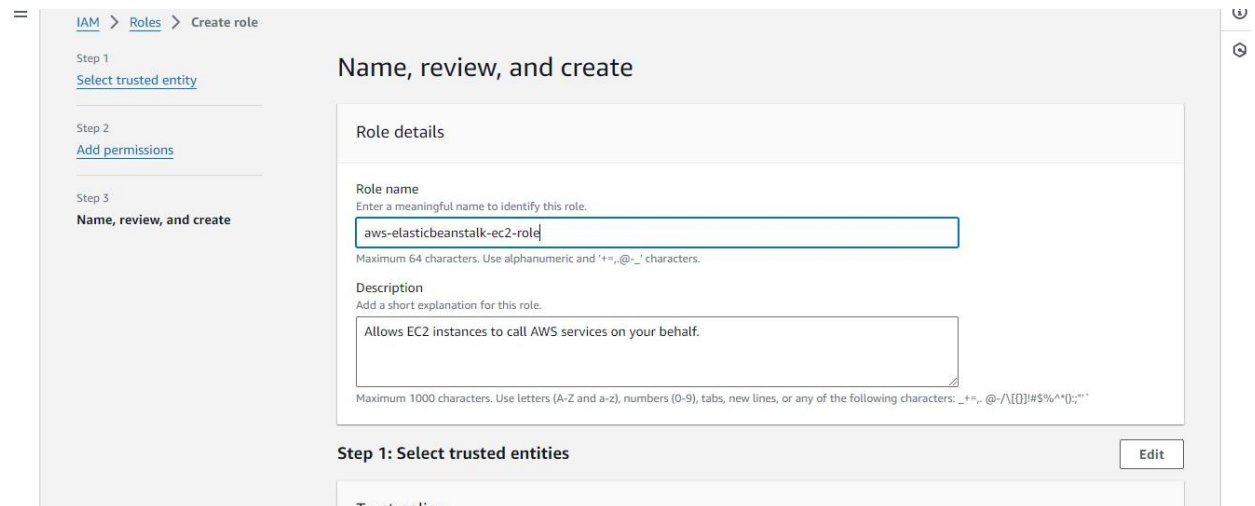
- Select use case as EC2.

This screenshot shows the 'Use case' selection screen within the AWS IAM console. The title is 'Use case' with a description: 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' Below this is a dropdown menu for 'Service or use case' which has 'EC2' selected. The main section is titled 'Choose a use case for the specified service.' and lists several use cases for EC2, each with a radio button and a description: 'EC2' (selected), 'EC2 Role for AWS Systems Manager', 'EC2 Spot Fleet Role', 'EC2 - Spot Fleet Auto Scaling', 'EC2 - Spot Fleet Tagging', 'EC2 - Spot Instances', 'EC2 - Spot Fleet', and 'EC2 - Scheduled Instances'. At the bottom right are 'Cancel' and 'Next' buttons.

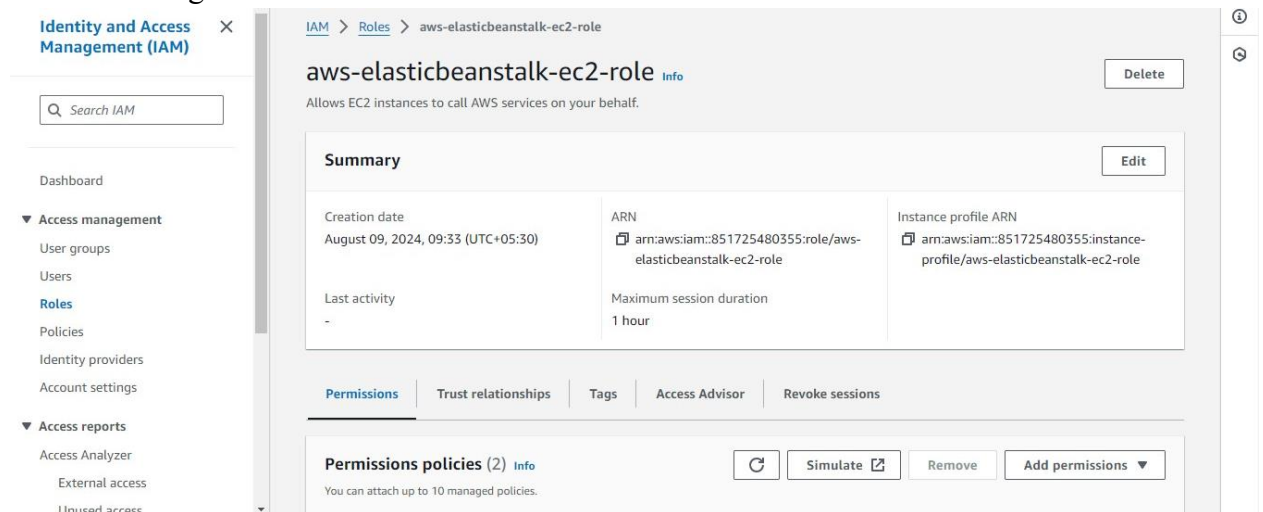
5. Select permissions as AWS Elastic Beanstalk Web Tier and AWS elastic Beanstalk worker tier.



6. Give a name to Role. Here I have given my role name as aws -elasticbeanstalk -ec2 role.



7. Then the role gets created.



Komal Sabale  
D15C-45

## Step2:- Creation Elastic Beanstalk Environment 1

1. Search and Open up Elastic Beanstalk and name your web app. (here I have given name komal)

The screenshot shows the AWS Elastic Beanstalk console. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information. The left sidebar lists the steps of the environment creation process. The main panel is titled 'Configure environment' and contains three sections: 'Environment tier' with 'Web server environment' selected, 'Application information' with 'Application name' set to 'komal', and 'Environment information' with 'Environment name' set to 'Komal-env'.

2. Select platform as PHP.

The screenshot shows the 'Platform' configuration section of the AWS Elastic Beanstalk console. It includes dropdown menus for 'Platform type' (Managed platform), 'Platform' (PHP), 'Platform branch' (PHP 8.3 running on 64bit Amazon Linux 2023), and 'Platform version' (4.3.1 (Recommended)). Below this is the 'Application code' section with 'Sample application' selected, and the 'Presets' section with 'Single instance (free tier eligible)' selected.

- After clicking on next u need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Keep other settings as it is.

## Configure service access [Info](#)

### Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

**Service role**

☒ Create and use new service role  
☐ Use an existing service role

**Service role name**  
Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

View permission details

**EC2 key pair**  
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Komal22

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role

View permission details

Cancel

Skip to review

Previous

Next

Komal Sabale  
D15C-45

Keep Set up networking, database, and tags, Configure instance traffic and scaling, Configure updates, monitoring, and logging all these default.

4. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

The screenshot shows the AWS Elastic Beanstalk console for an environment named 'Komal-env'. At the top, a blue banner states 'Elastic Beanstalk is launching your environment. This will take a few minutes.' Below this, the breadcrumb navigation shows 'Elastic Beanstalk > Environments > Komal-env'. The main content area is divided into two panels. The left panel, titled 'Environment overview', shows the 'Health' status as 'Unknown' and the 'Domain' as '-'. The right panel, titled 'Platform', shows the 'Platform' as 'PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2' and the 'Platform state' as 'Supported'. Below these panels is a horizontal tab bar with 'Events' selected. The 'Events' section shows a list of events with columns for 'Time', 'Type', and 'Details'. Two events are listed: one at 'August 16, 2024 21:01:28 (UTC+5:30)' with type 'INFO' and details 'Using elasticbeanstalk-us-east-1-017820683346 as Amazon S3 storage bucket for environment data.', and another at 'August 16, 2024 21:01:27 (UTC+5:30)' with type 'INFO' and details 'createEnvironment is starting.'.

### Step 3: Get a copy of your sample code

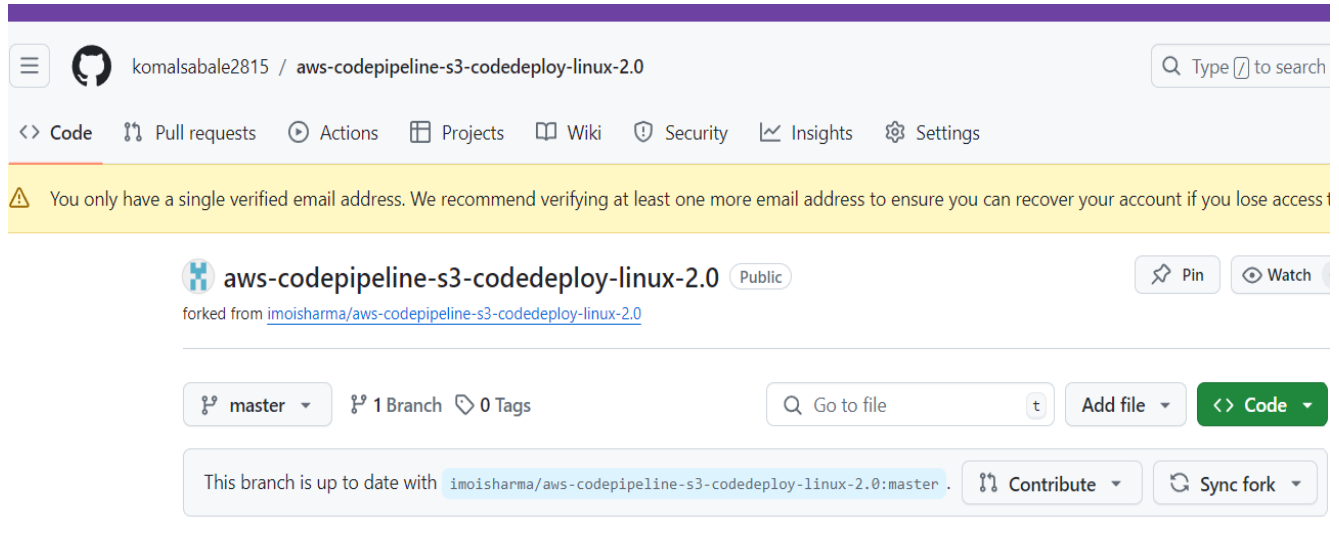
The screenshot shows the GitHub repository page for 'aws-codepipeline-s3-codedeploy-linux-2.0' by user 'imoisharma'. The page is titled 'Create a new fork' and includes a description: 'A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)'. Below this, a note states 'Required fields are marked with an asterisk (\*)'. The form has two main sections: 'Owner \*' and 'Repository name \*'. The 'Owner \*' section shows a dropdown menu with 'sadneya145' selected. The 'Repository name \*' section shows a text input field with 'aws-codepipeline-s3-code' entered. A green checkmark indicates 'aws-codepipeline-s3-codedeploy-linux-2.0 is available.' Below these sections, a note states 'By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.' The 'Description (optional)' section has a text input field with the text 'Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walk'. Below this, a checkbox labeled 'Copy the master branch only' is checked. A note below the checkbox states 'Contribute back to imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0 by adding your own branch. [Learn more.](#)'. At the bottom, a note states 'You are creating a fork in your personal account.' and a green 'Create fork' button is visible.

Komal Sabale  
D15C-45

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

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

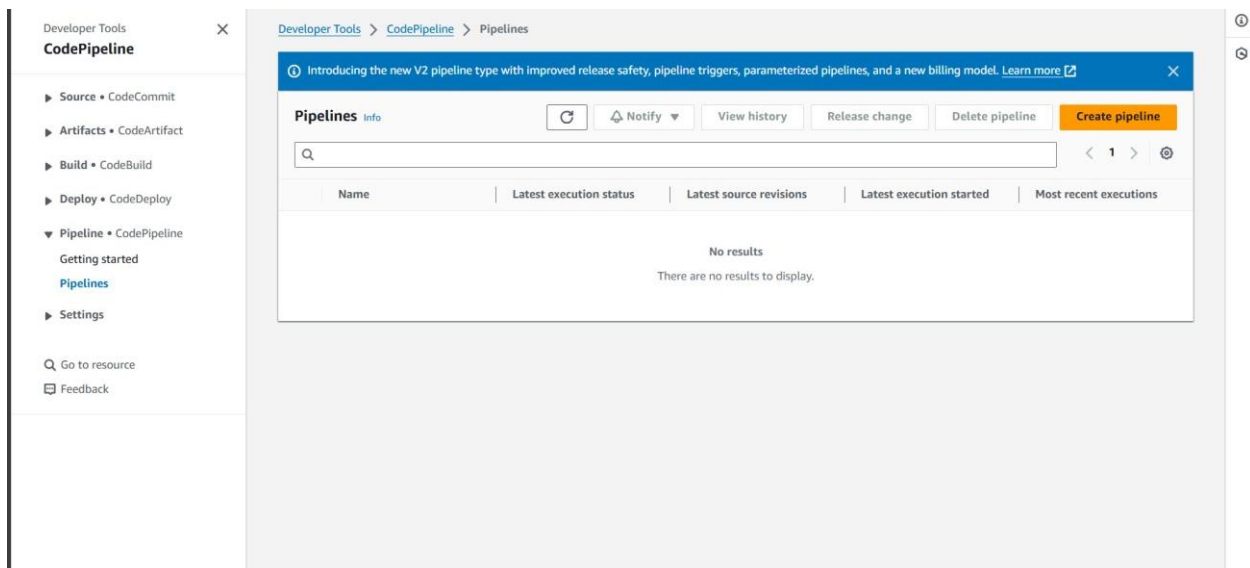
Go to the repository shared above and simply fork it.



#### Step 4: 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. Search CodePipeline in the search bar and click on create a new Pipeline.



Komal Sabale  
D15C-45

Give a name to your pipeline. Here I have given name as pipeline1.

[Developer Tools](#) > [CodePipeline](#) > Pipelines

Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. [Learn more](#)

×

Pipelines Info

↻

🔔 Notify ▼

View history

Release change

Delete pipeline

Create pipeline

🔍 pipeline1

×

0 matches < 1 > ⚙️

| Name   | Latest execution status | Latest source revisions | Latest execution started | Most recent executions |
|--|-------------------------|-------------------------|--------------------------|------------------------|
| Not found  |                         |                         |                          |                        |
| No results found for the following search: pipeline1 |                         |                         |                          |                        |



Komal Sabale  
D15C-45

2. In the source stage, choose GitHub v2 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.

[Pipeline](#) > [Pipelines](#) > [Create new pipeline](#)

## 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 1) ▼

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

[Connect to GitHub](#)



#### The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

#### Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

3.

### 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 1) ▼

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected



You have successfully configured the action with the provider.



#### The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

#### Repository

Q komalsabale2815/demo X

#### Branch

Q main X

#### Change detection options

4. select the forked repository then select the master branch.

**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 1) ▼

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

✓ You have successfully authenticated your account. ✕

**The GitHub (Version 1) action is not recommended**  
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

**Repository**  
komalsabale2815/aws-codepipeline-s3-codedeploy-linux-2.0 ✕

**Branch**  
🔍

**Change detection options**  
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ **GitHub webhooks (recommended)**  
Use webhooks in GitHub to automatically start my pipeline when a change occurs

☐ **AWS CodePipeline**  
Use AWS CodePipeline to check periodically for changes

5. Then select trigger type none.

**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.

After that, click Continue and skip the build stage. Proceed to the Deployment stage.

### Step 5: Deployment


1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

[Alt+S]



# 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](#) 

SourceArtifact ▼

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.

🔍 komal ✕

**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.

🔍 Komal-env ✕

Komal-env

Komal Sabale  
D15C-45

2. Click Next, Review and create the pipeline.

### Step 1: Choose pipeline settings

#### Pipeline settings

Pipeline name

pipeline1

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-pipeline1

#### Variables

| Name   | Default value | Description |
|--|---------------|-------------|
| <b>No variables</b>  |               |             |
| No variables defined at the pipeline level in this pipeline. |               |             |

[Alt+S]

Owner

komalsabale2815

Branch

main

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

komal

EnvironmentName

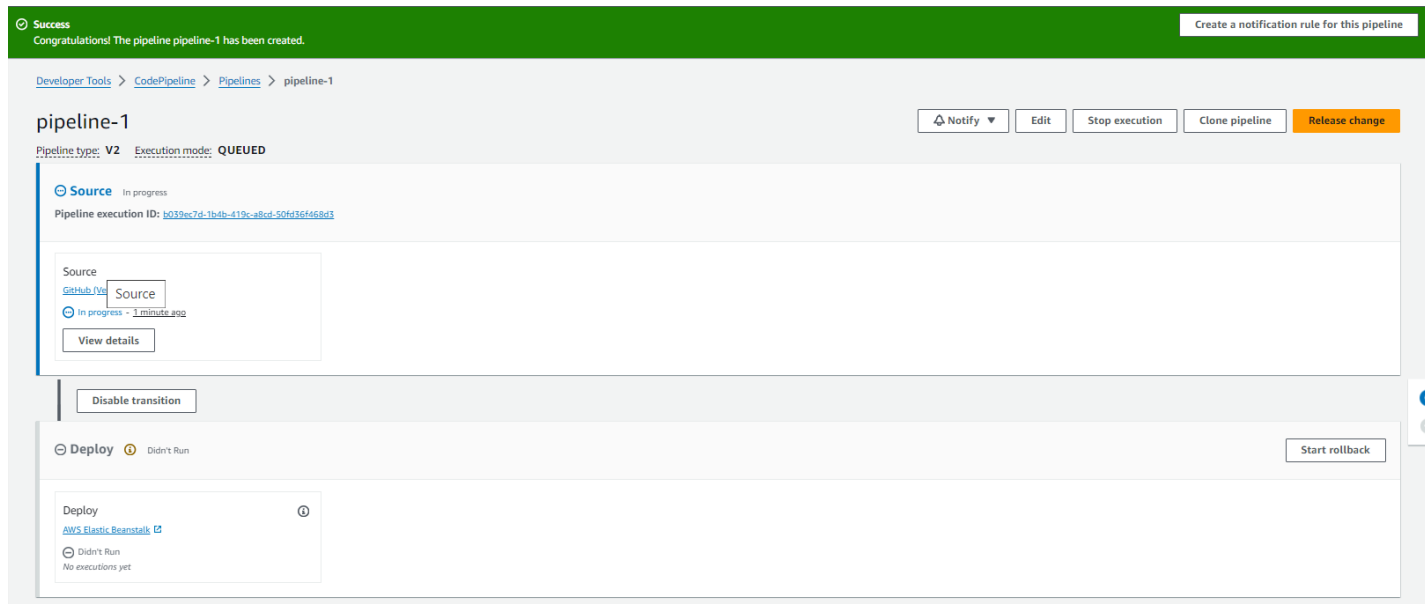
Komal-env

Configure automatic rollback on stage failure

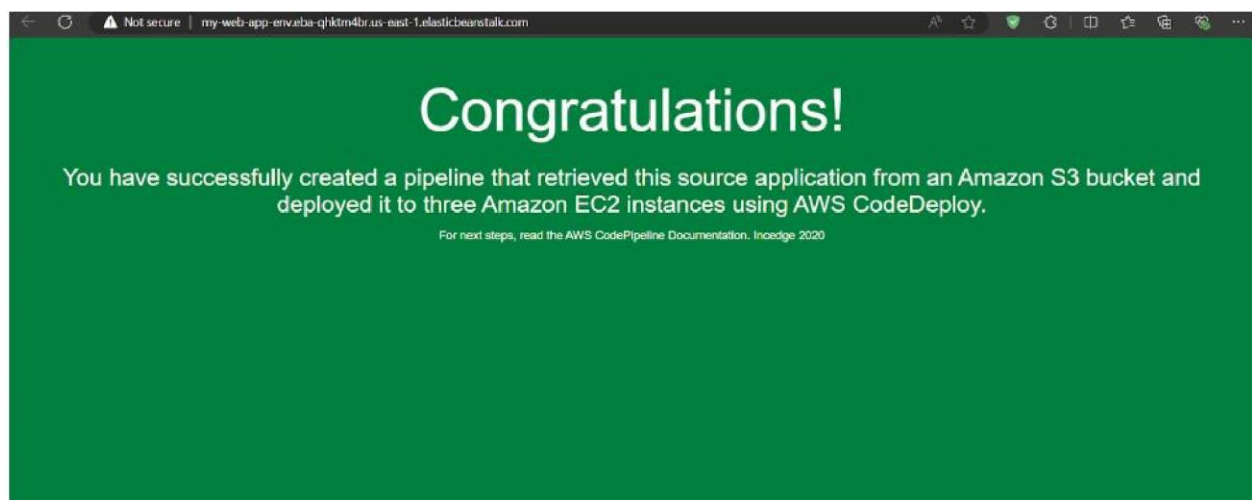
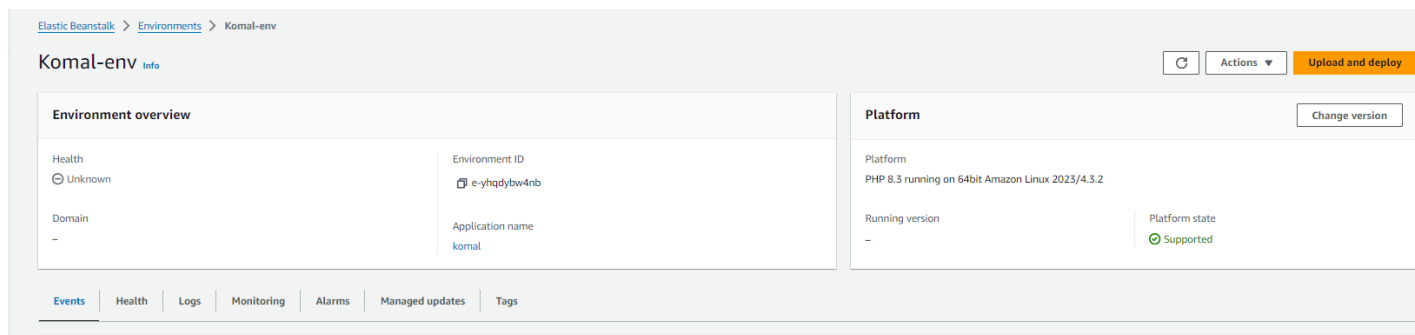
Disabled

Komal Sabale  
D15C-45

3. Then it will give you this result on screen. i.e. deployed successfully.



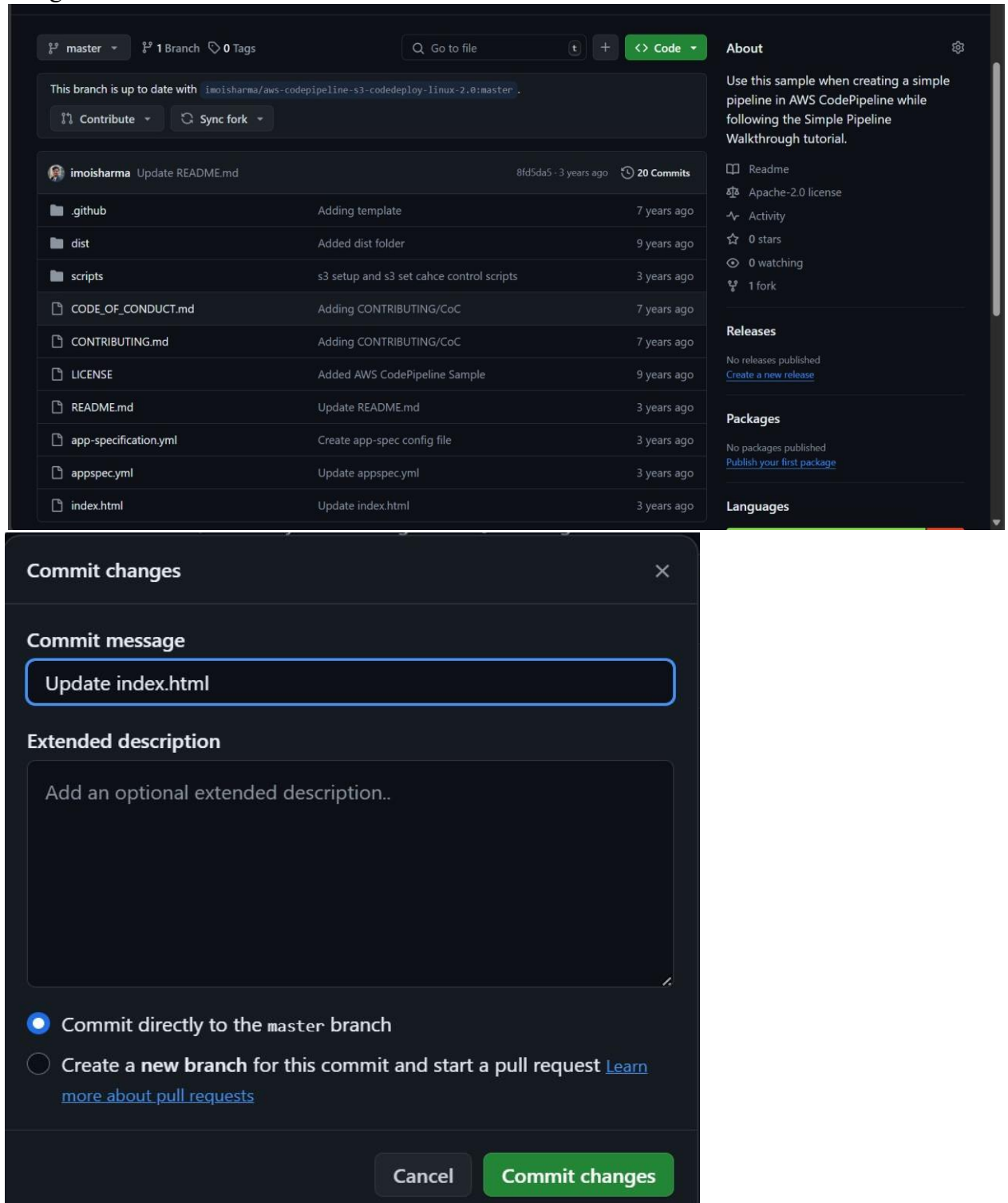
4. In a few minutes the website will get hosted successfully. Then click on the url present over the environment created on Elastic Benstalk.



If you can see this, that means that you successfully created an automated software using CodePipeline.

## Step 6: Committing changes to update app

1. In this we make some changes in the file. Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.



2. Then again start the deployment of the pipeline.

Success

Congratulations! The pipeline pipeline-1 has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > pipeline-1

pipeline-1

Pipeline type: V2 Execution mode: QUEUED

Source

In progress

Pipeline execution ID: b039ec7d-1b4b-419c-a8cd-50f436f658d3

Source

[GitHub / js](#)

In progress - 1 minute ago

View details

Disable transition

Deploy

Didn't Run

Deploy

[AWS Elastic Beanstalk](#)

Didn't Run

No executions yet

Start rollback

3. Check the changes in the website , here I have added a message in h3 tag.