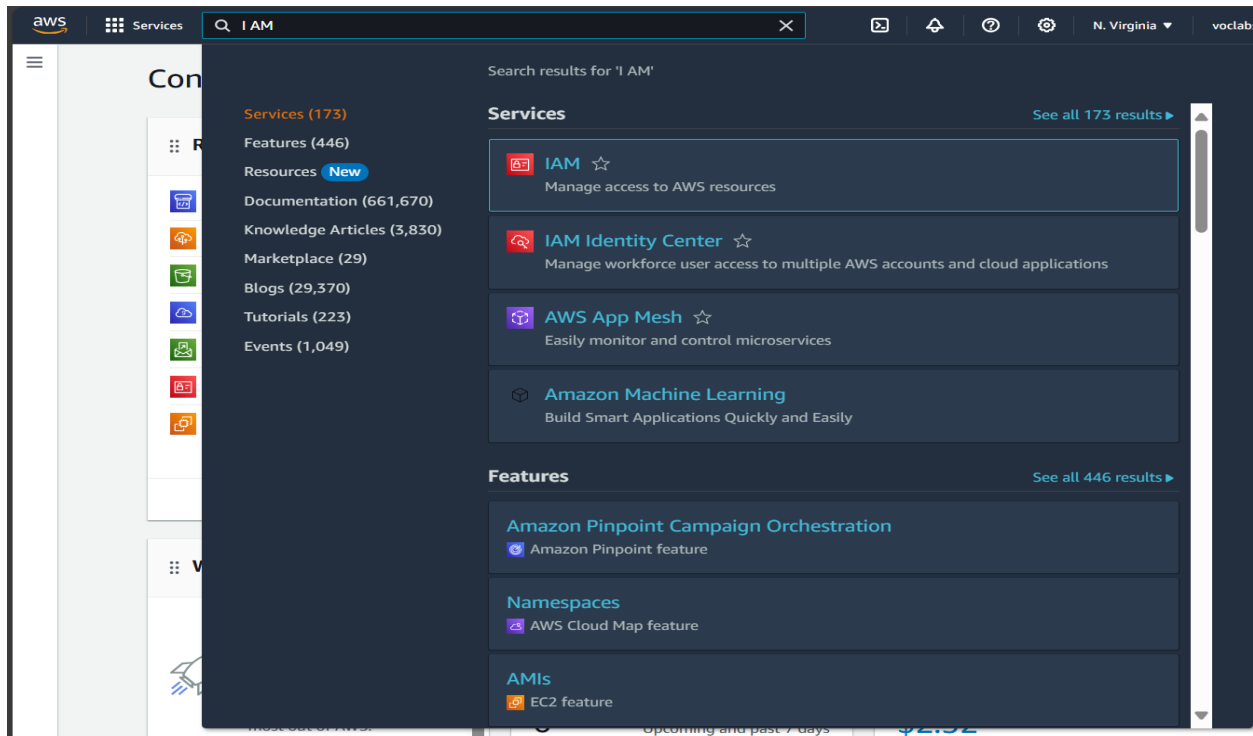


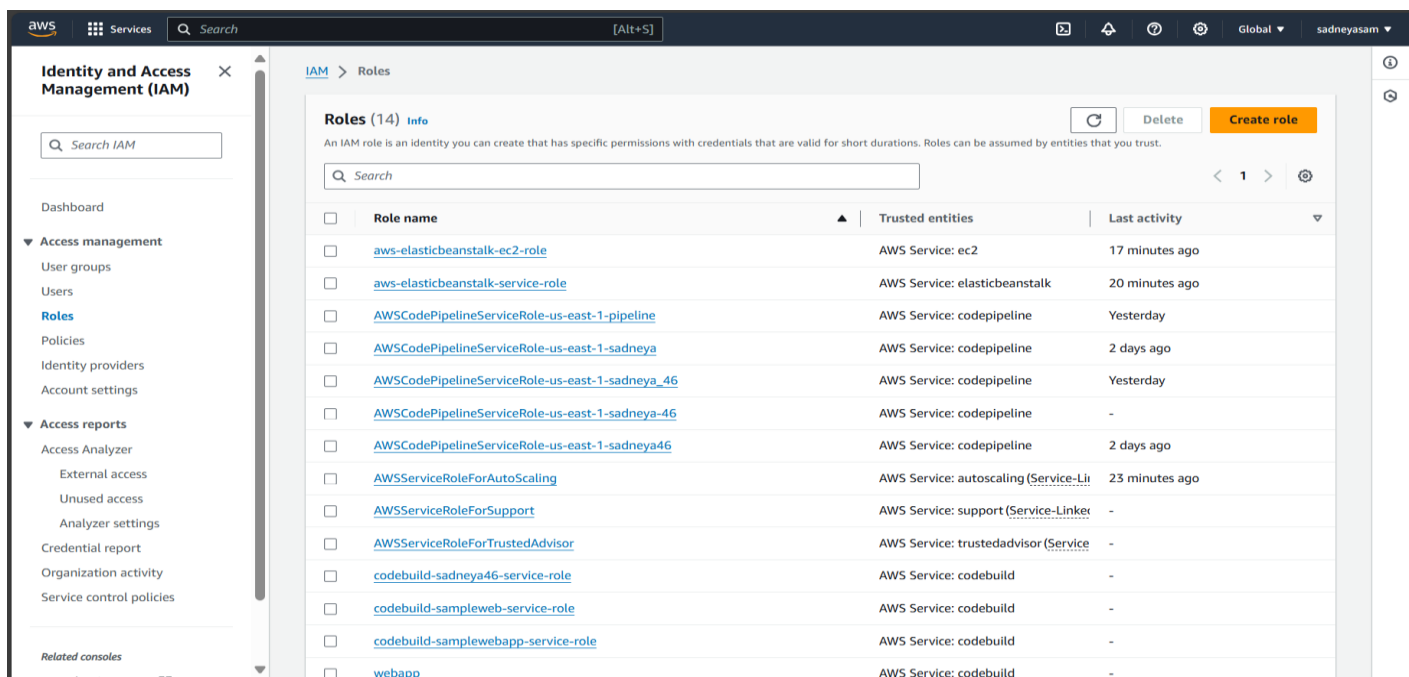
## Experiment No: 2

### Step1:- Creation of role:-

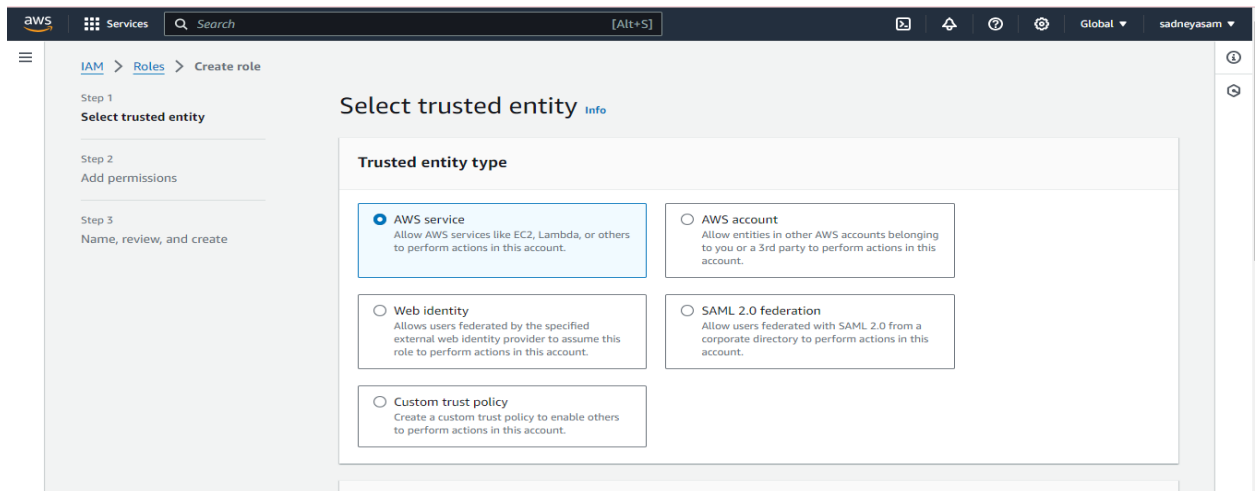
1. Login to your AWS account and search for IAM



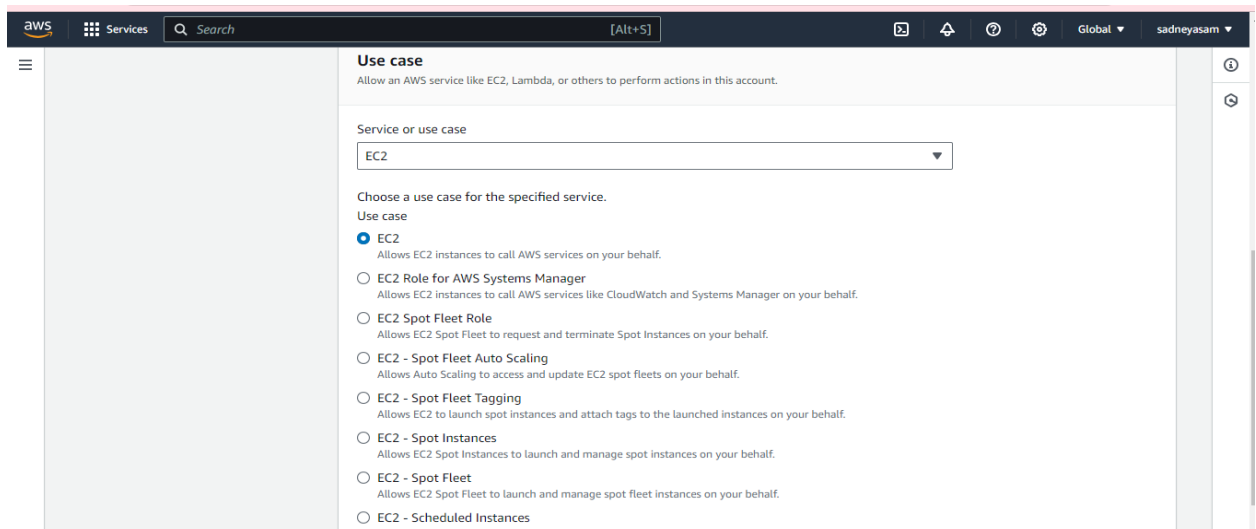
2. Then go into the role section and click on create role.



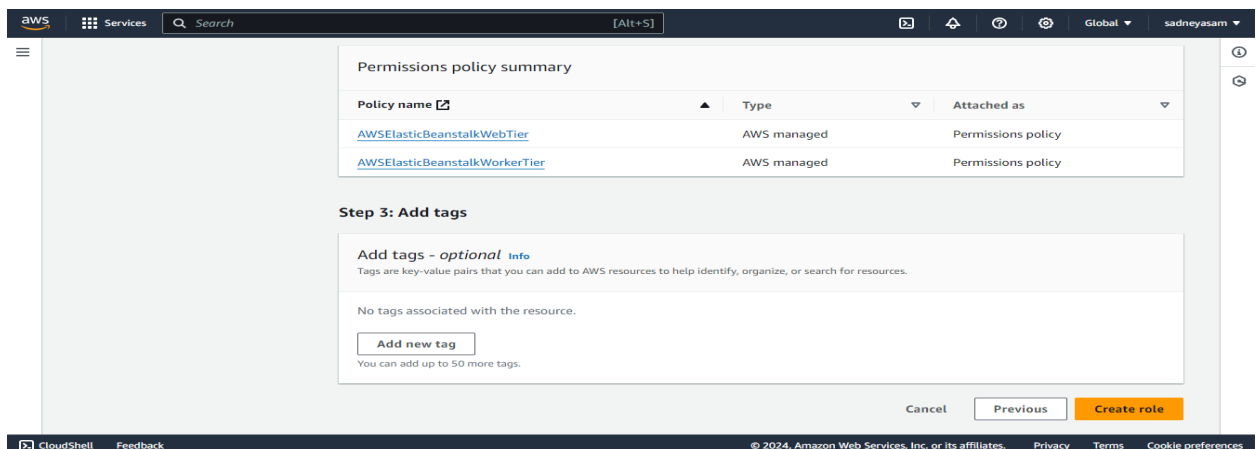
3. Then select a trusted entity as AWS service.



#### 4. Select use case as EC2.



#### 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.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The left sidebar indicates the current step is 'Step 1: Name, review, and create'. The main area is titled 'Name, review, and create' and contains a 'Role details' section. In this section, the 'Role name' field is filled with 'aws-elasticbeanstalk-ec2-role' and the 'Description' field is filled with 'Allows EC2 instances to call AWS services on your behalf.' Below the form, there is a section for 'Step 1: Select trusted entities' with an 'Edit' button.

aws-elasticbeanstalk-ec2-role

Allows EC2 instances to call AWS services on your behalf.

Step 1: Select trusted entities

7. Then the role gets created.

The screenshot shows the details of the 'aws-elasticbeanstalk-ec2-role' in the AWS IAM console. The left sidebar shows the 'Identity and Access Management (IAM)' section. The main area displays the role's summary, including its creation date, ARN, and instance profile ARN. Below the summary, there are tabs for 'Permissions', 'Trust relationships', 'Tags', 'Access Advisor', and 'Revoke sessions'. The 'Permissions' tab is active, showing 'Permissions policies (2)' and buttons for 'Simulate', 'Remove', and 'Add permissions'.

aws-elasticbeanstalk-ec2-role

Allows EC2 instances to call AWS services on your behalf.

Summary

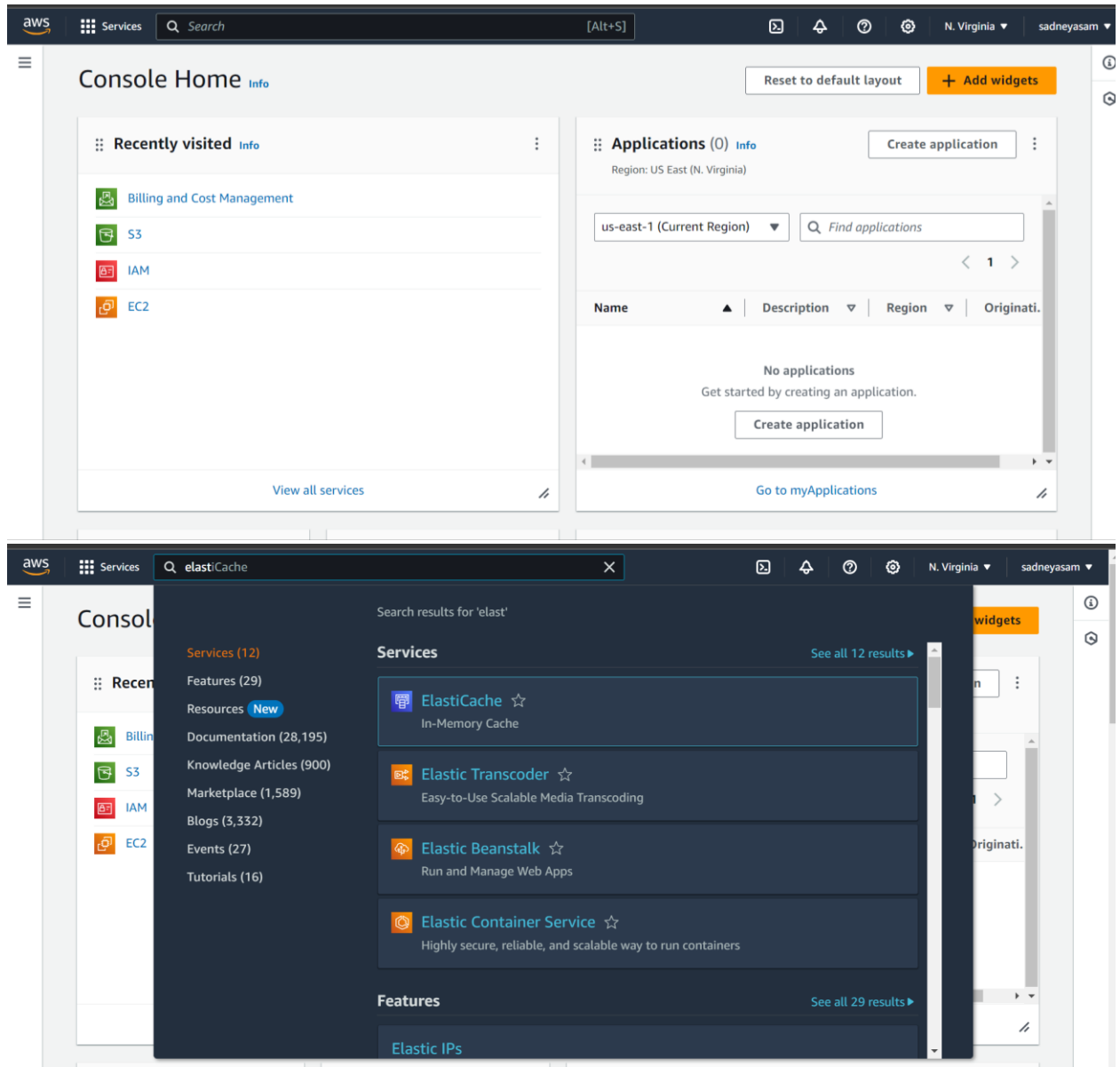
|   |   |  |
|---|---|--|
| Creation date<br>August 09, 2024, 09:33 (UTC+05:30) | ARN<br>arn:aws:iam::851725480355:role/aws-elasticbeanstalk-ec2-role | Instance profile ARN<br>arn:aws:iam::851725480355:instance-profile/aws-elasticbeanstalk-ec2-role |
| Last activity<br>-                                  | Maximum session duration<br>1 hour                                  |  |

Permissions policies (2)

Simulate Remove Add permissions

**Step2:- Creation Elastic Benstalk Environment**

1. search for Elastic Beanstalk in the search box.



2. Open up Elastic Beanstalk and name your web app. (here I have given name sadneya123)

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar lists the configuration steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main panel is titled 'Configure environment' and contains the following sections:

- Environment tier**: Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.
  - ☒ **Web server environment**: Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
  - ☐ **Worker environment**: Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)
- Application information**:
  - Application name**: sadneya123 (Maximum length of 100 characters.)
  - Application tags (optional)**: (Expandable section)

3. Select platform as PHP.

The screenshot shows the 'Platform' configuration section of the AWS Elastic Beanstalk console. It includes the following sections:

- 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**: PHP (dropdown menu)
- Platform branch**: PHP 8.3 running on 64bit Amazon Linux 2023 (dropdown menu)
- Platform version**: 4.3.1 (Recommended) (dropdown menu)
- Application code**:
  - ☒ **Sample application**
  - ☐ **Existing version**: Application versions that you have uploaded.
  - ☐ **Upload your code**: Upload a source bundle from your computer or copy one from Amazon S3.
- Presets**: Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
  - Configuration presets**:
    - ☒ **Single instance (free tier eligible)**
    - ☐ Single instance (using spot instance)
    - ☐ High availability
    - ☐ High availability (using spot and on-demand instances)
    - ☐ Custom configuration

4. 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. Which we created

in 1st part. Select role, then select key you have created then profile will be automatically selected according to role. then click on create application by keeping all the remaining settings as it is.

The screenshot displays the AWS Elastic Beanstalk console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and the user's profile 'sadneyasam'. The left sidebar shows a list of steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The main content area is titled 'Set up networking, database, and tags - optional' and contains three sections: 'Virtual Private Cloud (VPC)', 'Instance settings', and 'Database'. In the 'VPC' section, the 'Launch your environment in a custom VPC' option is selected. The 'Instance settings' section shows 'Public IP address' as 'Not activated' and 'Instance subnets' as 'No instance subnets to display'. The 'Database' section has 'Enable database' checked, 'Restore a snapshot' set to 'None', and 'Database settings' with 'Engine' set to 'MySQL' and 'Engine version' set to '5.7.33'. At the bottom, there are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Step 3 - optional  
[Set up networking, database, and tags](#)

Step 4 - optional  
[Configure instance traffic and scaling](#)

Step 5 - optional  
[Configure updates, monitoring, and logging](#)

Step 6  
[Review](#)

Service role

☐ Create and use new service role

☒ Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

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

key-linux

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](#)

Step 1  
[Configure environment](#)

Step 2  
[Configure service access](#)

Step 3 - optional  
**Set up networking, database, and tags**

Step 4 - optional  
[Configure instance traffic and scaling](#)

Step 5 - optional  
[Configure updates, monitoring, and logging](#)

Step 6  
[Review](#)

### Set up networking, database, and tags - optional [Info](#)

#### Virtual Private Cloud (VPC)

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

-

[Create custom VPC](#)

#### Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

☐ Activated

#### Instance subnets

| Availability Zone              | Subnet | CIDR | Name |
|--------------------------------|--------|------|------|
| No instance subnets            |        |      |      |
| No instance subnets to display |        |      |      |

#### Database [Info](#)

Integrate an RDS SQL database with your environment. [Learn more](#)

☒ Enable database

#### Restore a snapshot - optional

Restore an existing snapshot from a previously used database.

Snapshot

None

#### Database settings

Choose an engine and instance type for your environment's database.

Engine

Engine version

The image displays two screenshots of the AWS Management Console, specifically the Amazon Elastic Beanstalk console, showing the configuration steps for a new environment.

**Top Screenshot: Configure instance traffic and scaling - optional**

- Instances**
  - Root volume (boot device)
    - Root volume type: (Container default)
    - Size: 8 GB
    - IOPS: 100 IOPS
    - Throughput: 125 MB/s
  - Amazon CloudWatch monitoring
    - Monitoring interval: 5 minute
  - Instance metadata service (IMDS)
    - IMDSv1: ☒ Deactivated
  - EC2 security groups
    - EC2 security groups: (3)
    - Group name:
    - Group ID:
    - Name:
    - Groups listed: 

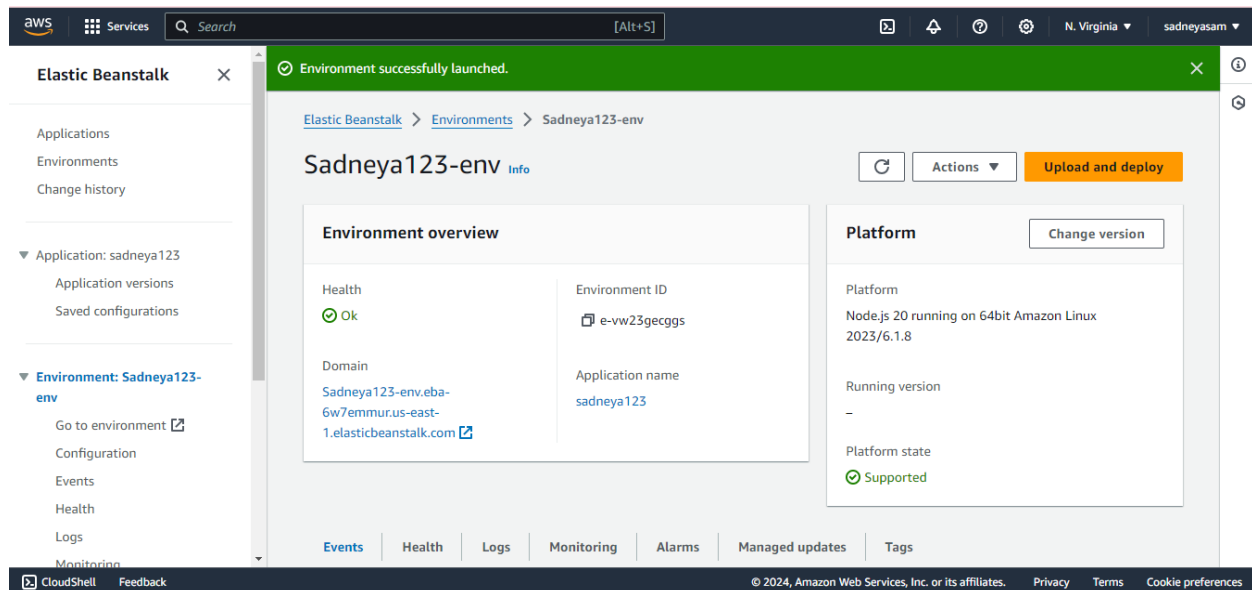
| Group name           | Group ID            | Name           |
|----------------------|---------------------|----------------|
| aws-ec2-vpc-stack... | sg-075e1d8b19c5c8c  | Sadneya123-env |
| default              | sg-0e4c73fbae598b78 |                |
| launch-wizard-1      | sg-0596c3ac3d79b0fc |                |

**Bottom Screenshot: Configure updates, monitoring, and logging - optional**

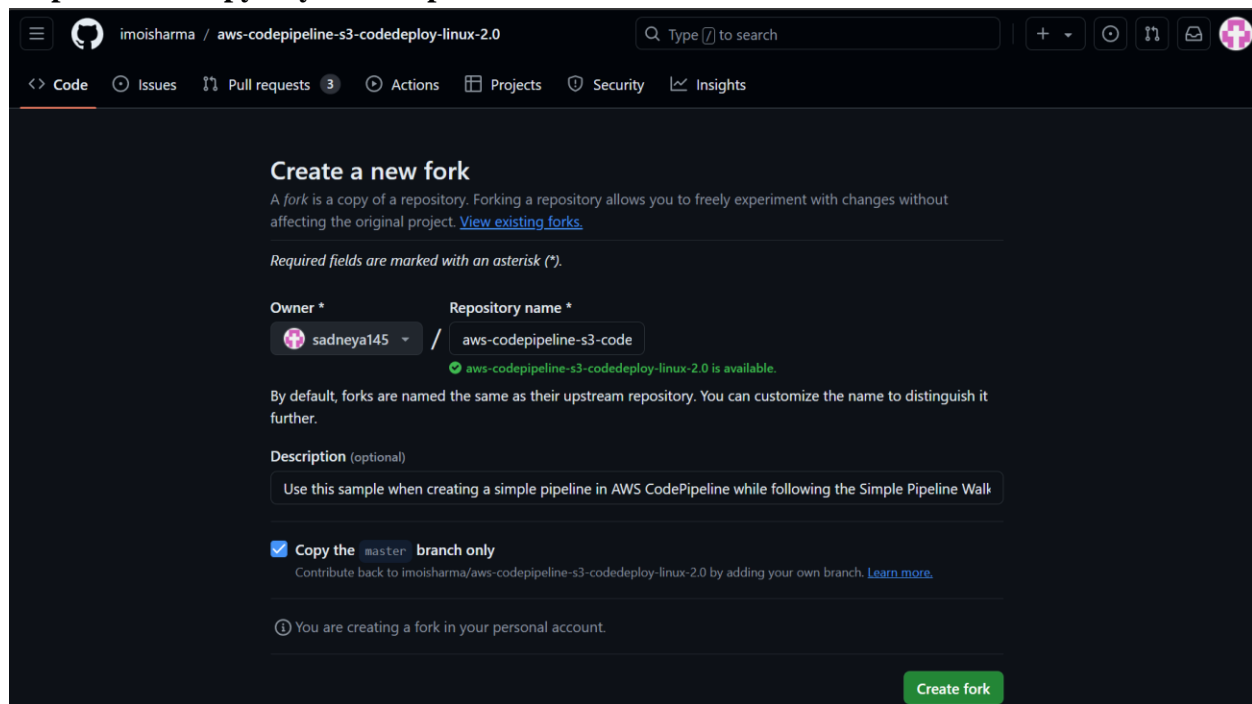
- Monitoring**
  - Health reporting
    - System: ☒ Enhanced
    - CloudWatch Custom Metrics - Instance:
    - CloudWatch Custom Metrics - Environment:
  - Health event streaming to CloudWatch Logs
    - Log streaming: ☐ Activated (standard CloudWatch charges apply)
    - Retention: 7
    - Lifecycle: Keep logs after terminating environment
  - Managed platform updates
    - Managed updates: ☒ Activated
    - Weekly update window: Sunday 00:00 to 11:00 UTC
    - Update level: Minor and patch
    - Instance replacement: ☐ Activated

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

5. 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.



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

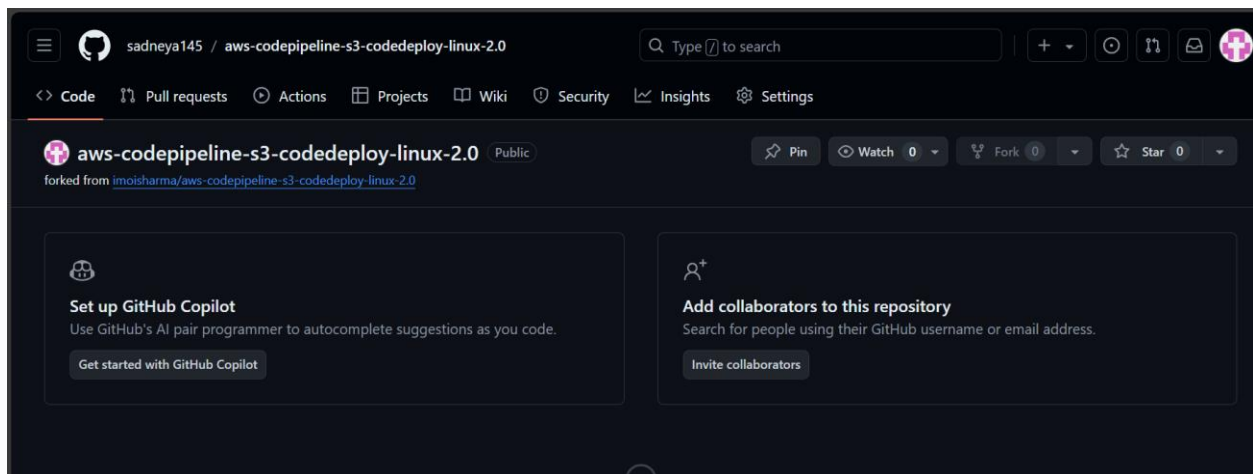


In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

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.

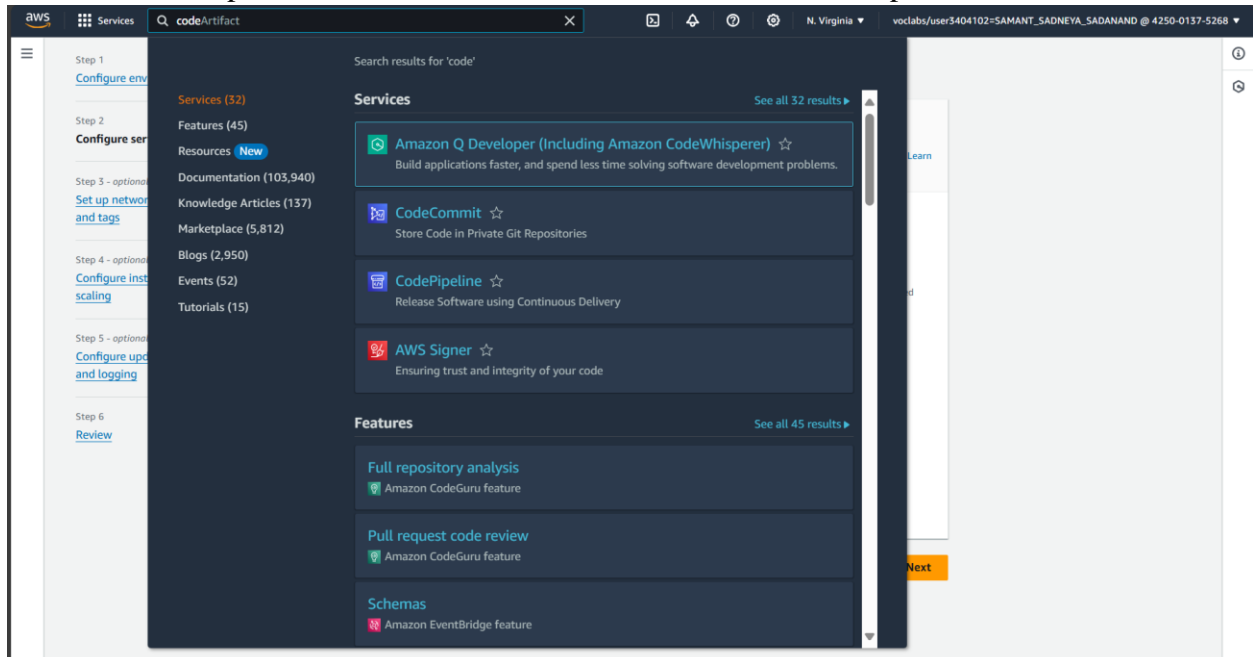


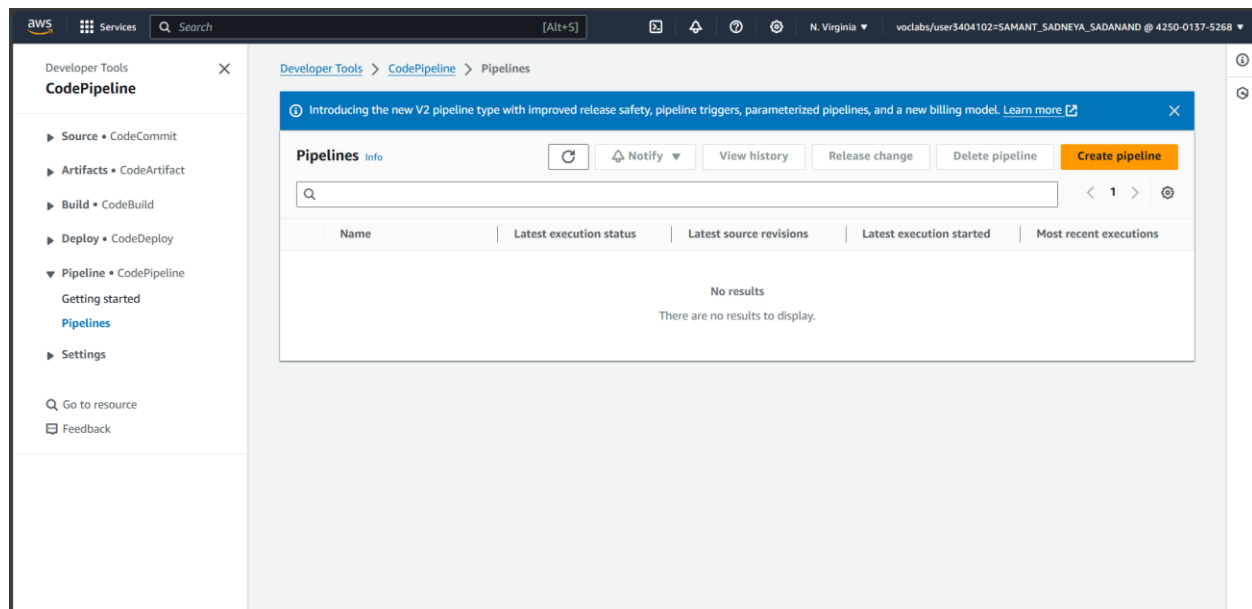


## Step 4: Creating a CodePipeline

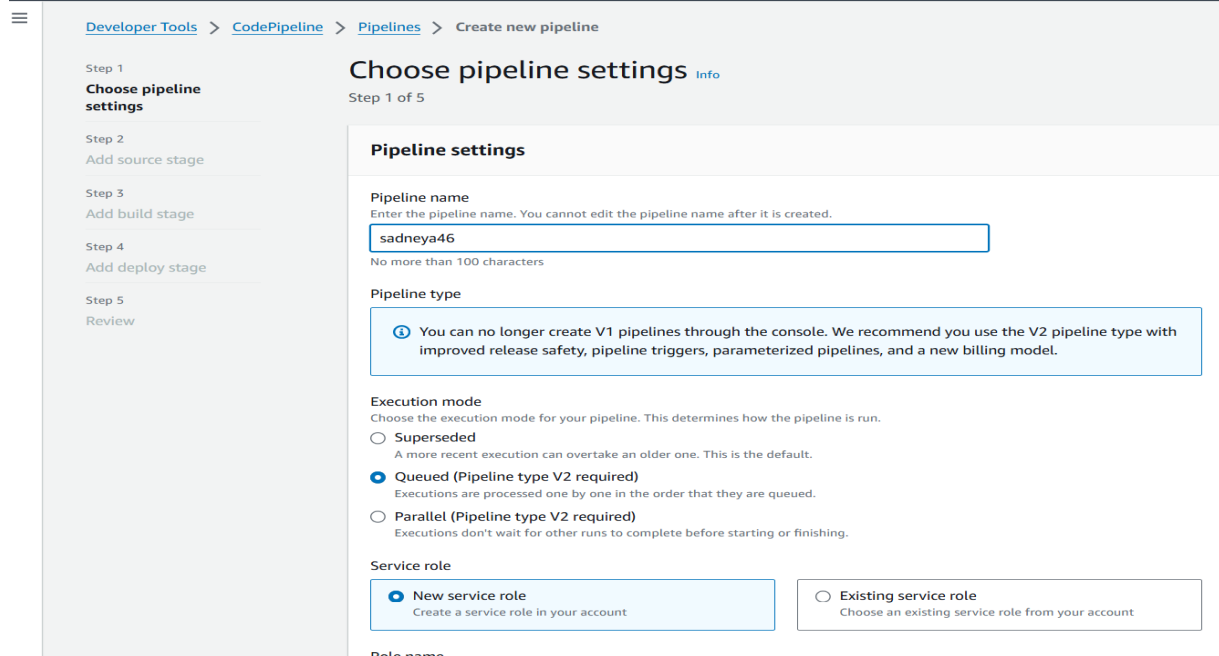
In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.





2. Give a name to your pipeline. Here I have given name as sadenya 46



3. 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.

Create connection | CodePipeline | us-east-1 - Personal - Microsoft Edge

https://us-east-1.console.aws.amazon.com/codesuite/settings/connection...

aws Services N. Virginia sadneyasam

Developer Tools > Connections > Create connection

Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. [Learn more](#)

## Connect to GitHub

### GitHub connection settings [Info](#)

Connection name

pipeline

GitHub Apps


GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.

53565526 or [Install a new app](#)

Tags - optional

[Connect](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Sign in to GitHub  
to continue to **AWS CodePipeline**  
(N. Virginia)

---

Username or email address

sadenyasam05@gmail.com

Password [Forgot password?](#)

.....

[Sign in](#)

[Sign in with a passkey](#)

New to GitHub? [Create an account](#)

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar lists steps: Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main panel is titled 'Source provider' and contains the following sections:

- Source provider:** A dropdown menu is set to 'GitHub (Version 2)'.
- New GitHub version 2 (app-based) action:** A blue box with an information icon stating: '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:** A text input field contains 'arn:aws:codeconnections:us-east-1:851725480355:connection/07d89a9a-42' followed by an 'X' icon. To its right is a 'Connect to GitHub' button.
- Ready to connect:** A green box with a checkmark icon stating: 'Your GitHub connection is ready for use.'
- Repository name:** A text input field contains 'sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0' followed by an 'X' icon. Below it, a note says: '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:** A text input field is empty.

The bottom of the console shows a dark navigation bar with 'CloudShell', 'Feedback', and a copyright notice for 2024 Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

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

This screenshot shows the configuration for the 'Repository name' and 'Default branch' in the AWS CodePipeline console. The 'Repository name' field is filled with 'sadneya145/aws-codepipeline-s3-codedeploy-linux-2.0'. The 'Default branch' field is filled with 'master'. Below these fields, there are two options for the 'Output artifact format':

- CodePipeline default:** Selected with a radio button. Description: 'AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.'
- Full clone:** Unselected with a radio button. Description: 'AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.'

5. Then select trigger type none.

The screenshot shows the 'Trigger' configuration section in the AWS CodePipeline console. It is titled 'Trigger' and contains the following options for 'Trigger type':

- No filter:** Selected with a radio button. Description: 'Starts your pipeline on any push and clones the HEAD.'
- Specify filter:** Unselected with a radio button. Description: 'Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.'
- Do not detect changes:** Unselected with a radio button. Description: '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.

The screenshot shows the AWS CodePipeline console interface. The top navigation bar includes the AWS logo, a 'Services' menu, and a search bar containing 'ElasticCache'. The breadcrumb trail is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. On the left sidebar, the steps of the pipeline are listed: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage - currently selected), and Step 5 (Review). The main content area is titled 'Add deploy stage' with an 'Info' icon and 'Step 4 of 5'.

A blue information box states: '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.'

The 'Deploy' section contains the following configuration options:

- Deploy provider:** Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider. The dropdown menu shows 'AWS Elastic Beanstalk'.
- Region:** The dropdown menu shows 'US East (N. Virginia)'.
- Input artifacts:** Choose an input artifact for this action. [Learn more](#) (with an external link icon). The dropdown menu is empty, with a note '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. The search bar contains '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. The search bar contains 'Sadneya123-env-1'.
- ☐ Configure automatic rollback on stage failure

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next' (highlighted in orange).

## 2. Click Next, Review and create the pipeline.

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  
sadneya46

Pipeline type  
V2

Execution mode  
QUEUED

Artifact location  
codepipeline-us-east-1-204862929919

Service role name  
arn:aws:iam::851725480355:role/service-role/AWSCodePipelineServiceRole-us-east-1-sadneya\_46

Variables

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

aws Services

Step 2: Add source stage

Source action provider

Source action provider  
GitHub (Version 2)

OutputArtifactFormat  
CODE\_ZIP

DetectChanges  
true

ConnectionArn  
arn:aws:codeconnections:us-east-1:851725480355:connection/800ab011-6749-4e3f-8a54-ba529c85155b

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

Default branch  
master

Trigger configuration

You can add additional pipeline triggers after the pipeline is created.

Trigger type  
No filter

Step 3: Add build stage

Build action provider

Build stage  
No build

**Step 4: Add deploy stage**

Deploy action provider

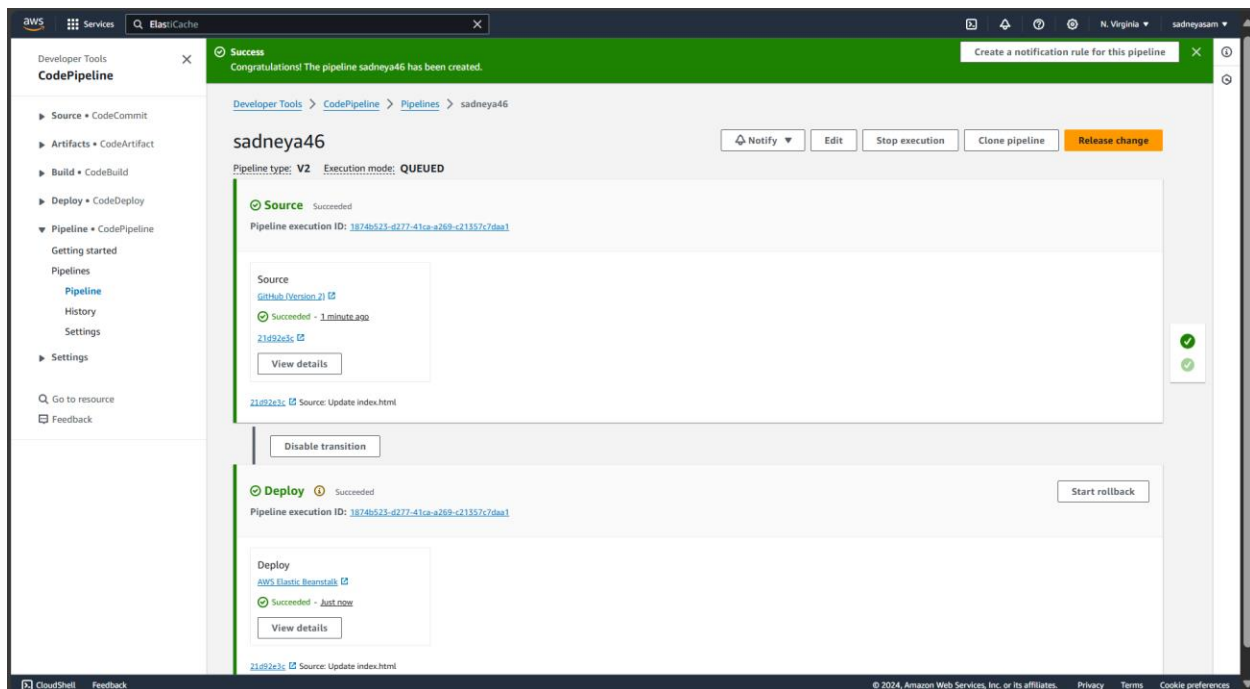
Deploy action provider  
AWS Elastic Beanstalk

ApplicationName  
sadneya123

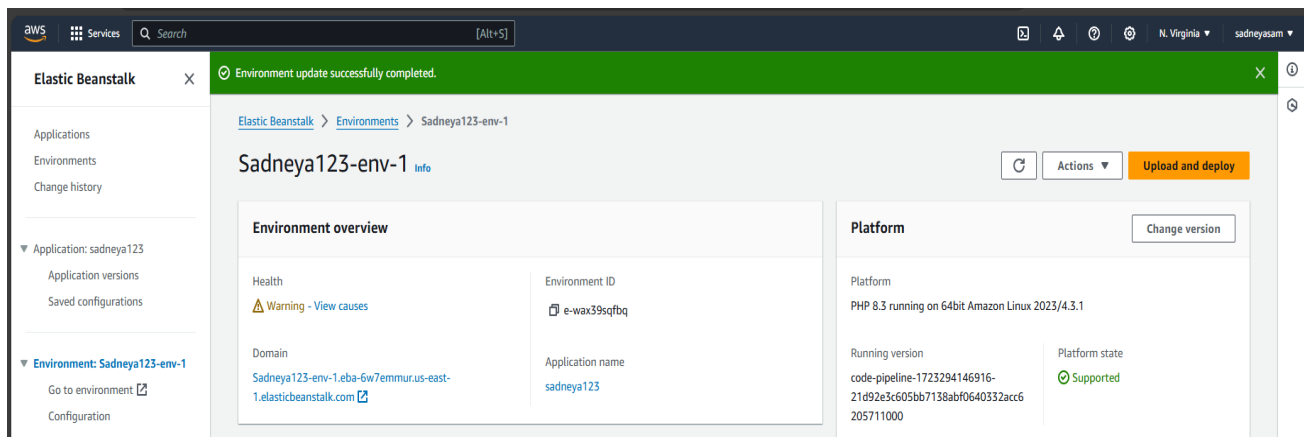
EnvironmentName  
Sadneya123-env

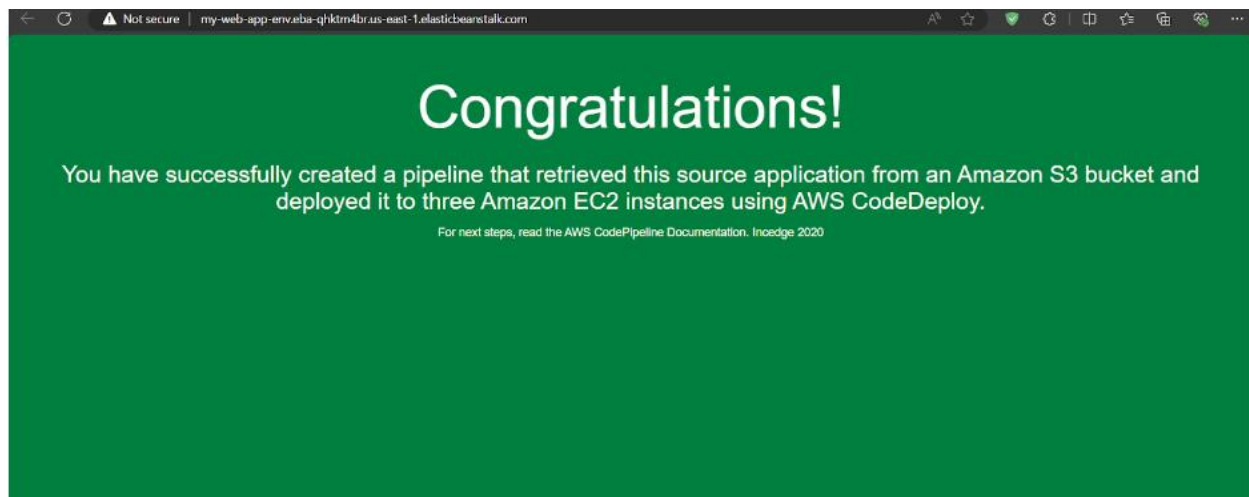
Configure automatic rollback on stage failure  
Disabled

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 Beanstalk.

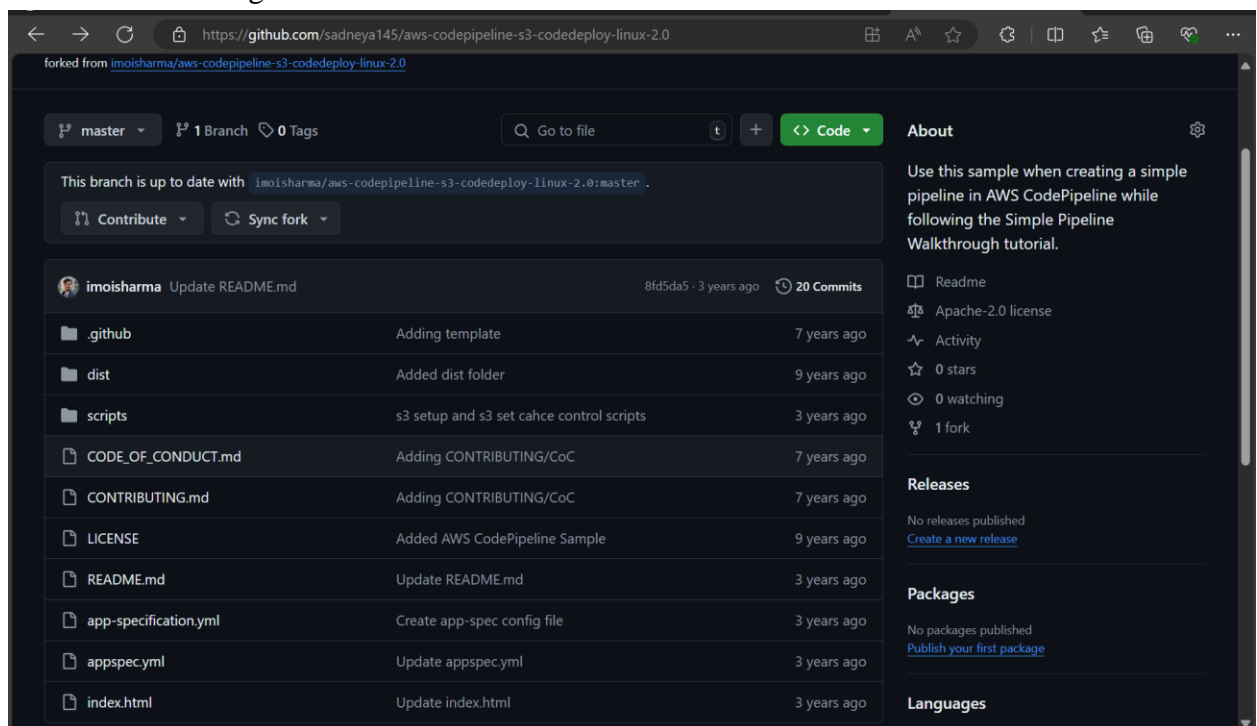




If you can see this, that means that you successfully created an automated software using CodePipeline.

### 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.





### Commit changes

#### Commit message

Update index.html

#### Extended description

Add an optional extended description..

☒ Commit directly to the master branch

☐ Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

Cancel

Commit changes

2. Then again start the deployment of the pipeline.

Developer Tools

CodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Go to resource

Feedback

Success

Congratulations! The pipeline sadneya46 has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > sadneya46

sadneya46

Pipeline type: V2 Execution mode: QUEUED

Source

Succeeded

Pipeline execution ID: 1874b625-4277-41ca-a269-c21357c7d9a1

Source

[GitHub \(Version 2\)](#)

Succeeded - 1 minute ago

21d92e3c

View details

21d92e3c: Source: Update index.html

Disable transition

Deploy

Succeeded

Pipeline execution ID: 1874b625-4277-41ca-a269-c21357c7d9a1

Deploy

[AWS Elastic Beanstalk](#)

Succeeded - Just now

View details

21d92e3c: Source: Update index.html

Start rollback

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

