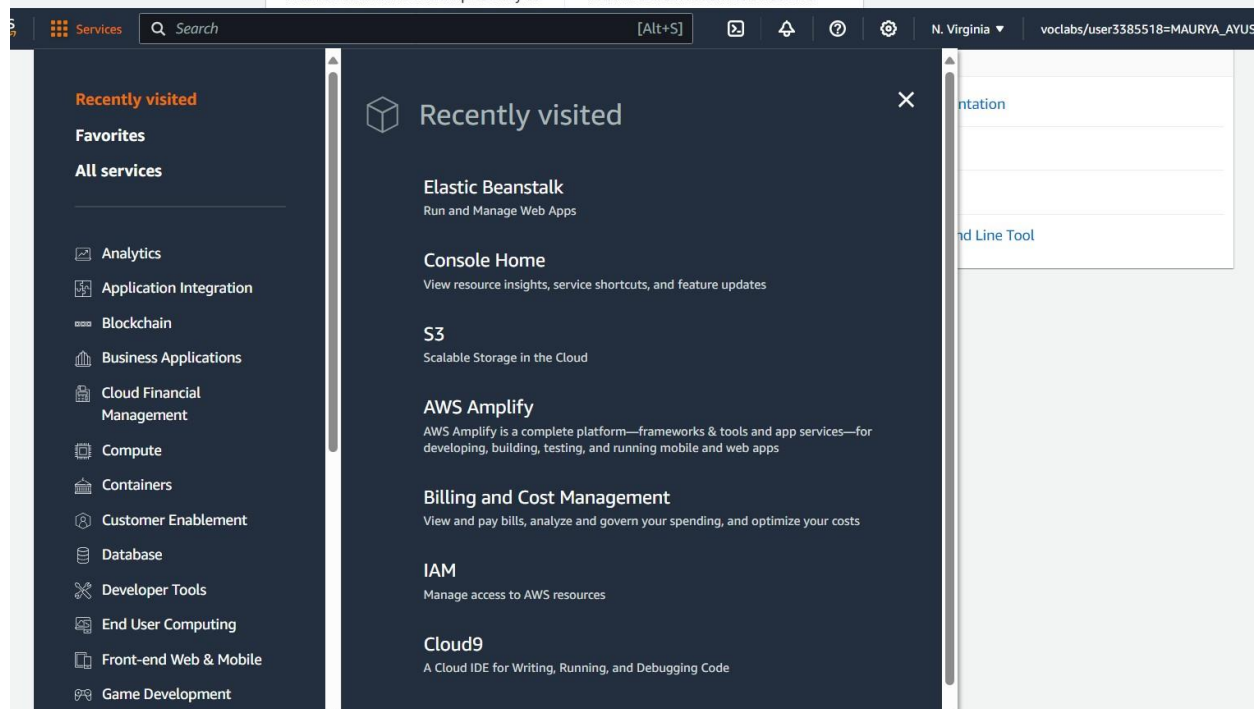


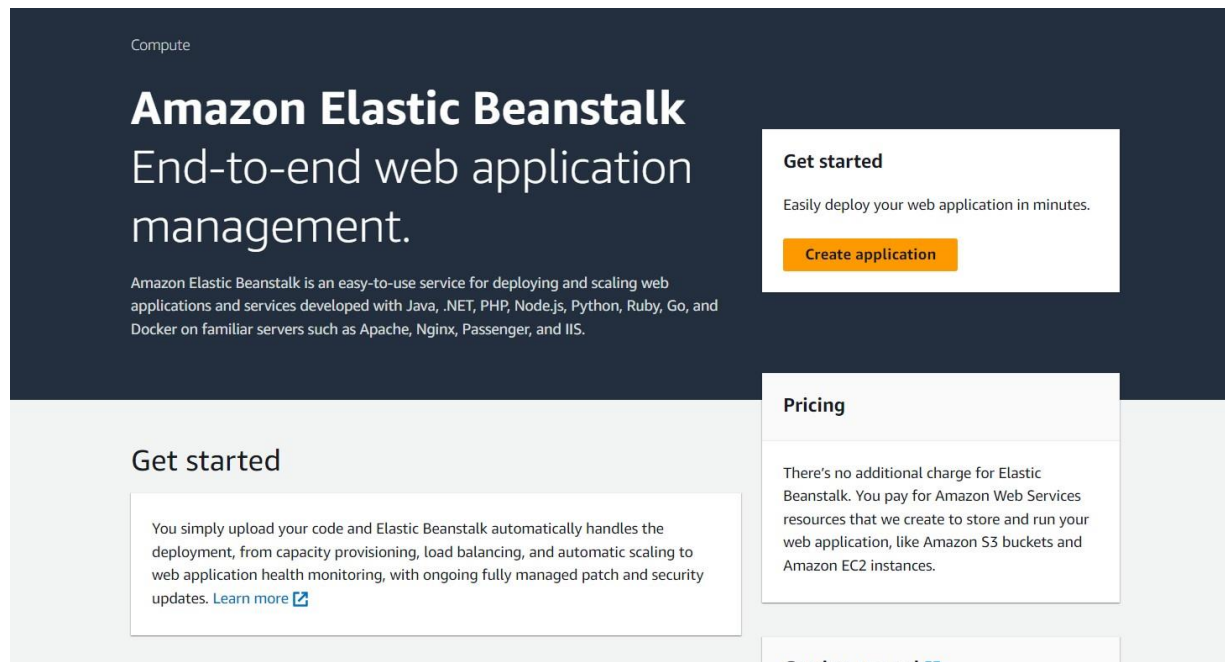
Name :Aditya kirtane ,Roll No.26 , Div:D15C

## Experiment No.2

**Step 1:** Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services.



**Step 2:** Go to Elastic Beanstalk and click on Create Application



**Step 3:** Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to single instance. Click on 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

## Configure environment [Info](#)

### Environment tier [Info](#)

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 [Info](#)

Application name

Maximum length of 100 characters.

► Application tags (optional)

### Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

.us-east-1.elasticbeanstalk.com

Check availability

Environment description

### 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

PHP ▼

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023 ▼

PHP 8.3 running on 64bit Amazon Linux 2023

[Alt+S]



N. Virginia ▾

voclabs/user3385518=

4.3.1 (Recommended) ▾

## Application code [Info](#)

- ☒ 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 [Info](#)

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

Cancel

Next

**Step 4 :** Use an existing service role and choose whatever service role is available on your account.

The screenshot shows the 'Configure service access' step in the AWS Elastic Beanstalk console. On the left, a sidebar lists the 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 'Configure service access' and includes a 'Service access' section. This section has two radio buttons: 'Create and use new service role' (unselected) and 'Use an existing service role' (selected). Below the radio buttons, there is a text box for 'Existing service roles' and a dropdown menu showing 'AWSCloud9SSMAccessRole'. There are also dropdowns for 'EC2 key pair' (showing 'Choose a key pair') and 'EC2 instance profile' (showing 'AWSCloud9SSMInstanceProfile'). A 'View permission details' button is at the bottom.

**Step 5 :** Review the settings that you have set up for your application and submit your application.

The screenshot shows the 'Review' step in the AWS Elastic Beanstalk console. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Review' and shows a summary of the configuration. It includes a 'Step 1: Configure environment' section with an 'Edit' button, showing environment information like 'Environment tier' (Web server environment), 'Application name' (my\_web\_34), 'Environment name' (Myweb34-env), 'Application code' (Sample application), and 'Platform' (arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1). Below this is a 'Step 2: Configure service access' section with an 'Edit' button, showing 'Service role' (arn:aws:iam::011528263337:role/service-role/AWSCloud9SSMAccessRole) and 'EC2 instance profile' (AWSCloud9SSMInstanceProfile). At the bottom, there is a 'Step 3: Set up networking, database, and tags' section with an 'Edit' button. A dark navigation bar at the bottom contains a 'Services' menu, a search bar, and a keyboard shortcut '[Alt+S]'.

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

### Monitoring

System enhanced	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false

### Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	

[Alt+S]



### Platform software

Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	Document root	Max execution time
Off	—	60
Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		

### Environment properties

Key ▲	Value ▼
No environment properties There are no environment properties defined	

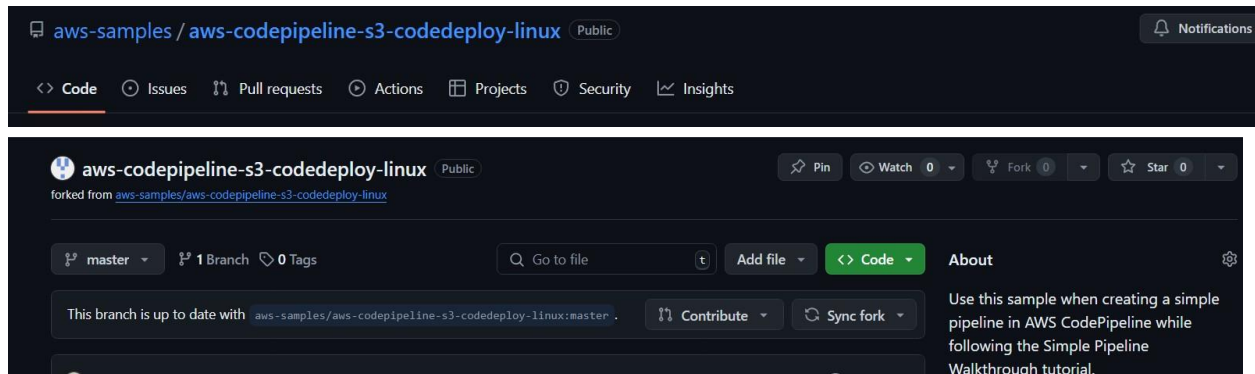
Cancel

Previous

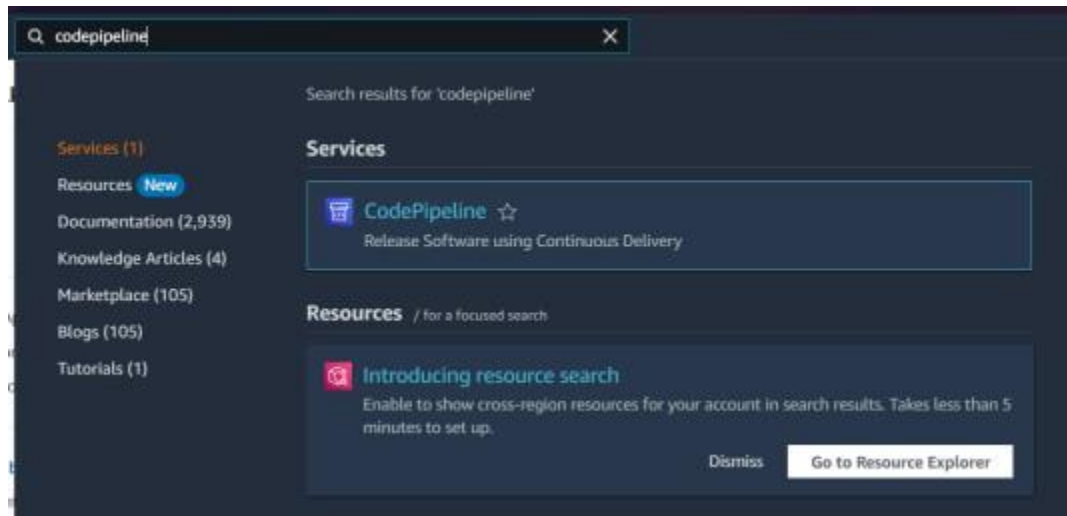
Submit

**Step 6:** Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

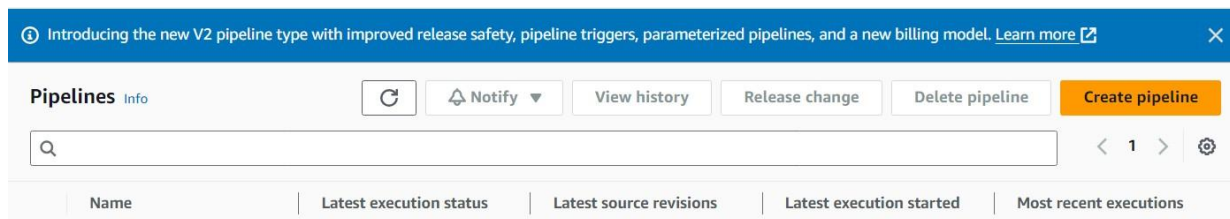
<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>



**Step 7:** Search CodePipeline in the services tab and click on it.



**Step 8:** Click on Create Pipeline.



**Step 9:** Give a name to your Pipeline. A new service role would be created with the name of 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

## Choose pipeline settings Info

Step 1 of 5

### Pipeline settings

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

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.

**Service role**

☒ New service role

☐ Existing service role

**Role name**

Type your service role name

☒ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

### Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

Add variable

You can add up to 50 variables.

**?** The first pipeline execution will fail if variables have no default values.

► **Advanced settings**

Cancel

Next



**Step 10 :** Select a source provider ( as Github ( Version 2 ) ). Click on Connect to Github to connect your github.

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

## 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 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.

or

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

**Step 11:** Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app

Developer Tools > Connections > Create connection

## Create a connection Info

### Create GitHub App connection Info

Connection name

MyGithub

► Tags - optional

Developer Tools > Connections > Create connection

Beginning July 1, 2024, the console will create connections with codecommit@aws in the resource ARN. Provision with both service profiles will continue to display in the console. [Learn more](#)

## Connect to GitHub

### GitHub connection settings

Connection name

MyGithub

GitHub Apps

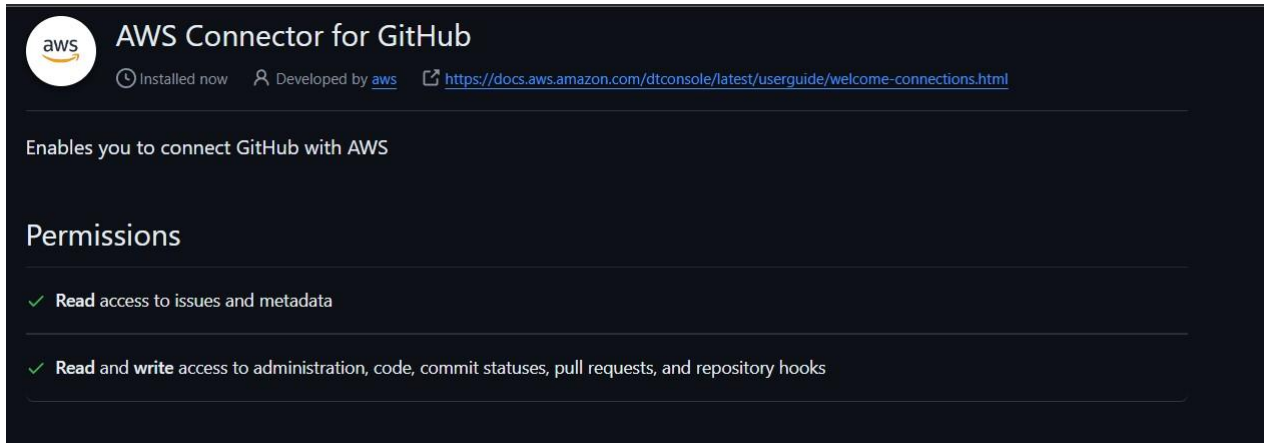
GitHub Apps require a link for your connection with GitHub. Install a new app and save this connection.

or

► Tags - optional

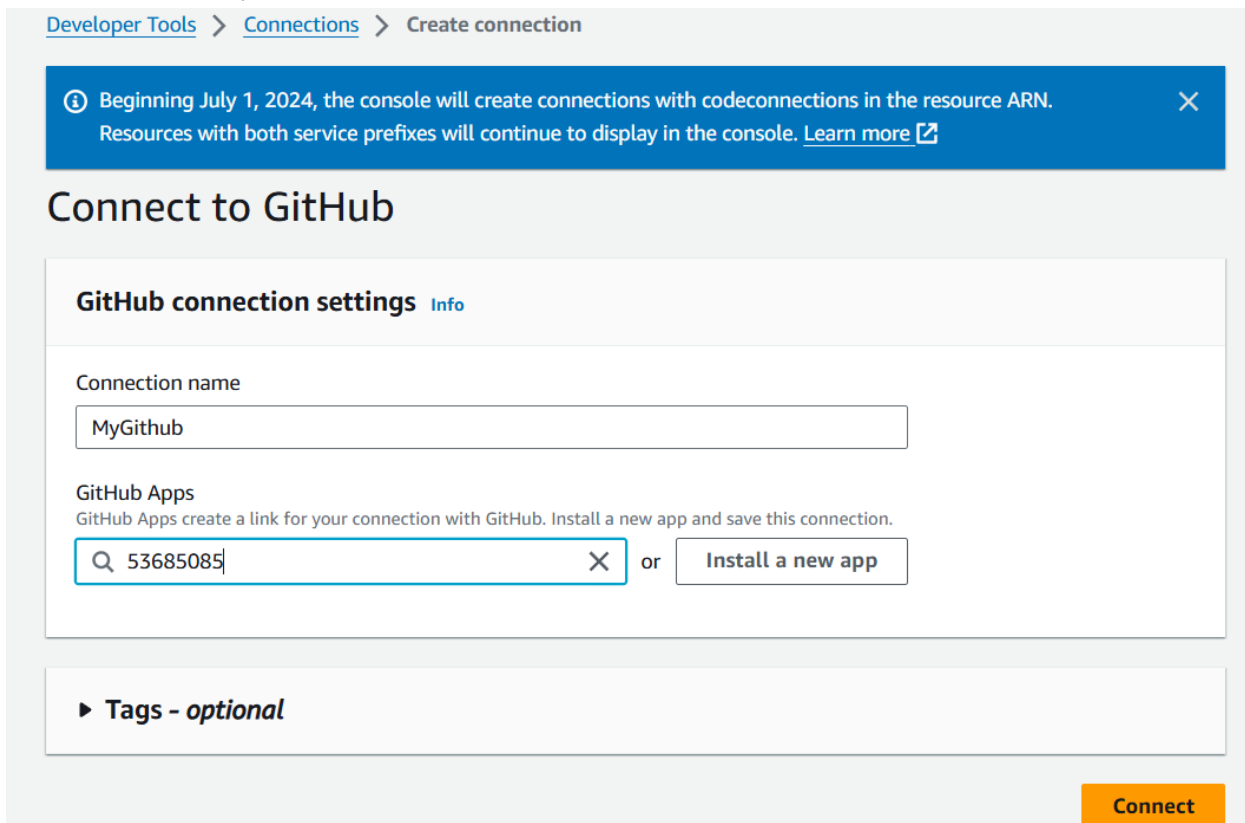


**Step 12 :** This will direct you to install AWS Connector On Your GitHub. Install it to your account and give it its permissions.



The screenshot shows the 'AWS Connector for GitHub' app page. At the top, there's an AWS logo and the app name. Below that, it says 'Installed now' and 'Developed by aws'. A link to the user guide is provided. The main heading is 'Permissions'. Underneath, there are two permission rows, both marked with a green checkmark: 'Read access to issues and metadata' and 'Read and write access to administration, code, commit statuses, pull requests, and repository hooks'.

**Step 13:** After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use



The screenshot shows the 'Connect to GitHub' page. At the top, there's a breadcrumb trail: 'Developer Tools > Connections > Create connection'. Below this is a blue information banner with a close button (X) and text: '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'. The main heading is 'Connect to GitHub'. Below that is a section titled 'GitHub connection settings' with an 'Info' link. Inside this section, there's a 'Connection name' field with the value 'MyGithub'. Below that is a 'GitHub Apps' section with the text: 'GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.' There's a search input field containing '53685085' with a close button (X), followed by the word 'or' and a button labeled 'Install a new app'. At the bottom of the settings section is a link 'Tags - optional'. At the very bottom right of the page is an orange 'Connect' button.

Step 1

Choose pipeline settings

Step 2

**Add source stage**

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

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 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.

arn:aws:codeconnections:us-east-1:011528263337:connection/3dfa55ab-a5c X

 or 

Connect to GitHub

 **Ready to connect**

Your GitHub connection is ready for use.

Repository name

**Step 14:** Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

more


Connection

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

arn:aws:codeconnections:us-east-1:011528263337:connection/3dfa55ab-a5c X

 or 

Connect to GitHub

 **Ready to connect**

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.

Q AyushMaurya3114/aws-codepipeline-s3-codedeploy-linux 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 branch

Default branch will be used only when pipeline execution starts from a different source or manually started.

Q master X

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.

**Step 15:** Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

The screenshot shows the 'Trigger' configuration screen in the AWS CodePipeline console. At the top, there are two radio button options: 'CodePipeline default' (selected) and 'Full clone'. Below this is the 'Trigger' section with the title 'Trigger type' and a description 'Choose the trigger type that starts your pipeline.' There are three radio button options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'. A blue information box at the bottom states: 'You can add additional sources and triggers by editing the pipeline after it is created.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

☒ **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.

### 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.

**i** You can add additional sources and triggers by editing the pipeline after it is created.

Cancel Previous Next

**Step 16:** Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

The screenshot shows the 'Add build stage' screen in the AWS CodePipeline console. On the left, there is a sidebar with a list of steps: 'Step 1: Choose pipeline settings', 'Step 2: Add source stage', 'Step 3: Add build stage' (highlighted), 'Step 4: Add deploy stage', and 'Step 5: Review'. The main area is titled 'Add build stage' with a sub-header 'Step 3 of 5'. Below this is a section titled 'Build - optional' with a sub-section 'Build provider' and a description 'This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.' There is a dropdown menu for the build provider. At the bottom right, there are four buttons: 'Cancel', 'Previous', 'Skip build stage', and 'Next'.

Step 1  
Choose pipeline settings

Step 2  
Add source stage

Step 3  
**Add build stage**

Step 4  
Add deploy stage

Step 5  
Review

### Add build stage Info

Step 3 of 5

**Build - optional**

**Build provider**  
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Cancel Previous Skip build stage Next

Add deploy stage

Step 5

Review

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.

Q my\_web\_34

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.

Q Myweb34-env

☐ Configure automatic rollback on stage failure

Create Pipeline

**Step 17:** Check all the information and click on create Pipeline.

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

my\_web\_34

EnvironmentName

Myweb34-env

Configure automatic rollback on stage failure

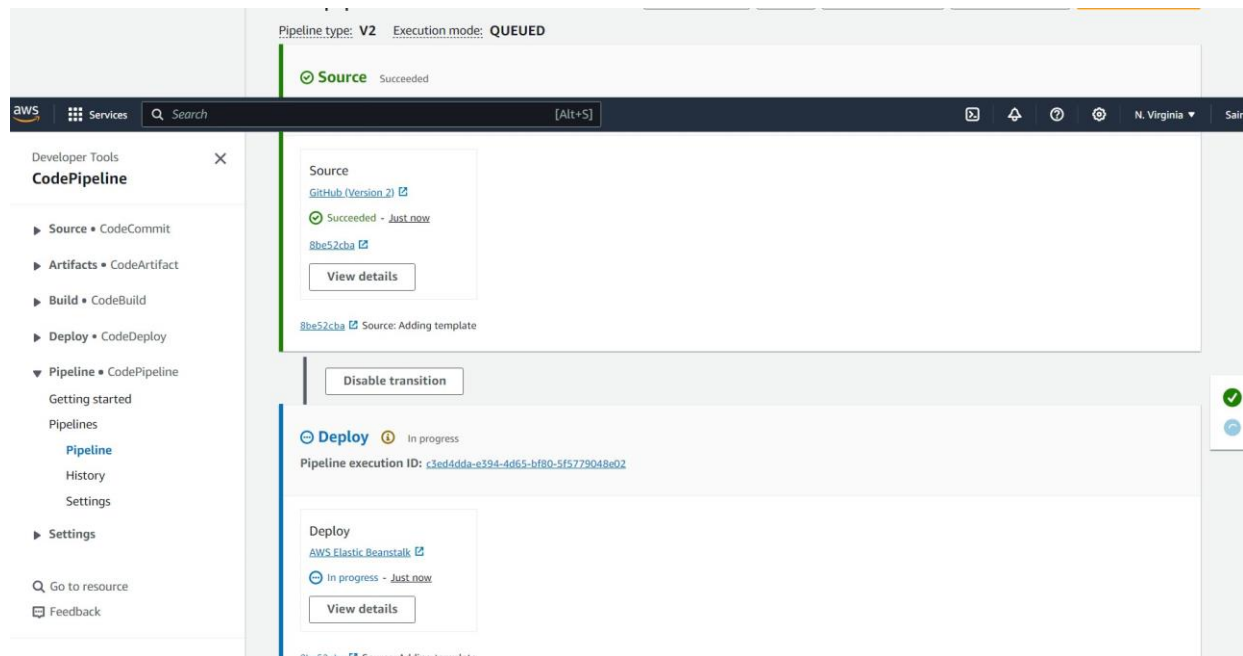
Disabled

Cancel

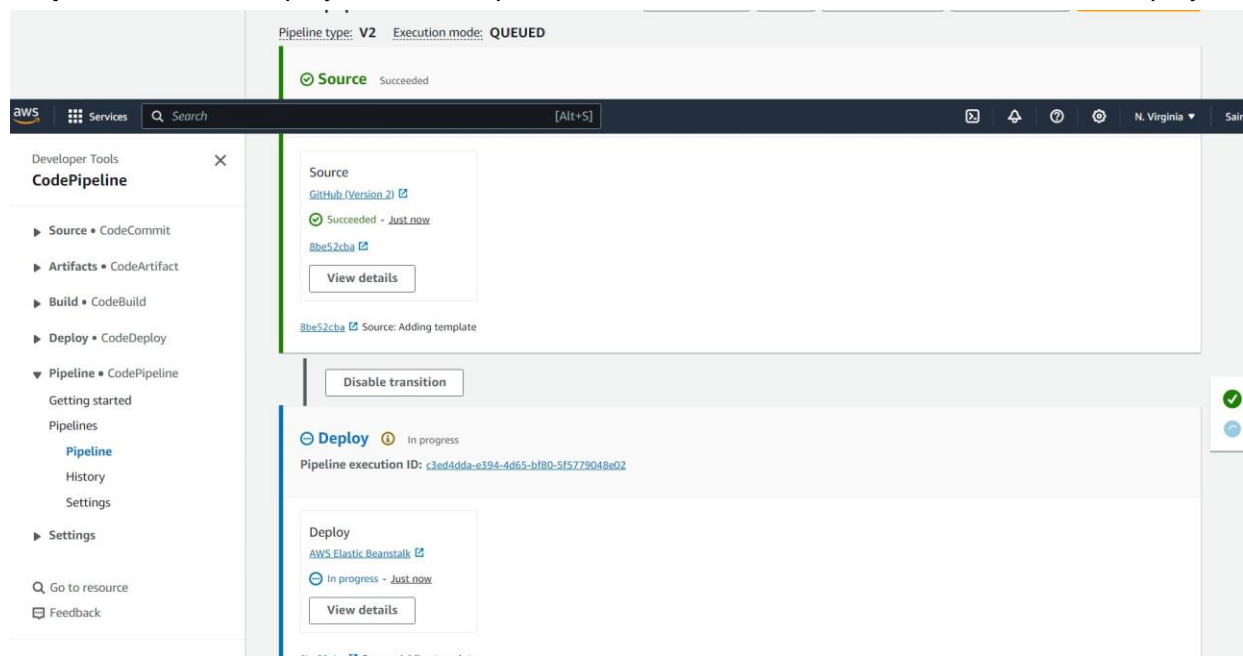
Previous

Create pipeline

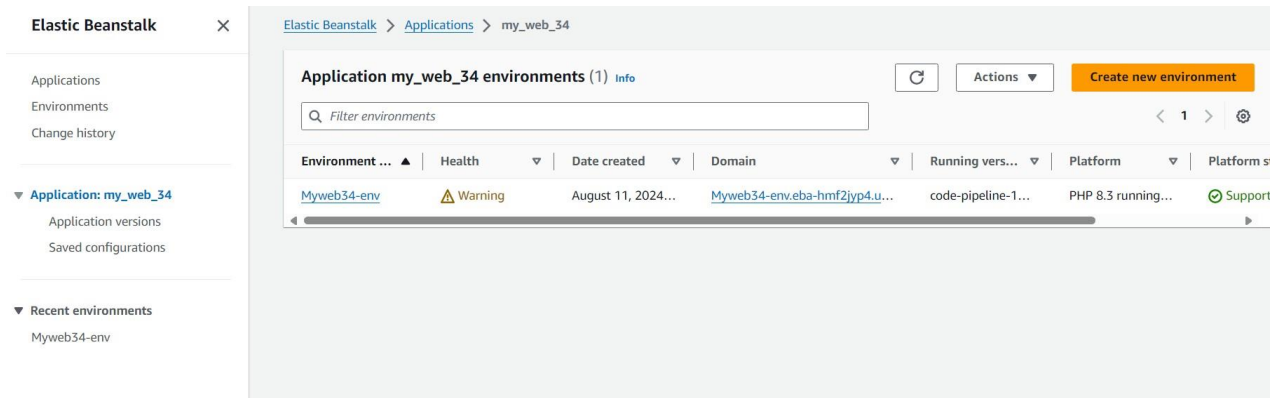
**Step 18:** If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy.



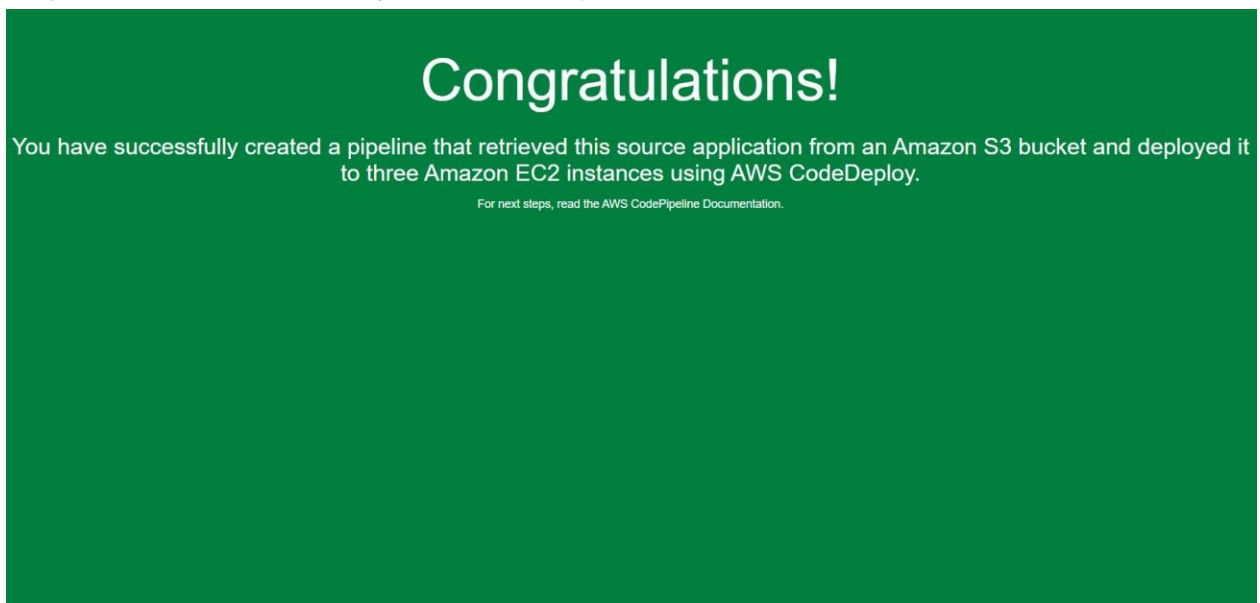
**Step 19:** Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



**Step 20:** This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain.



**Step 21:** This will successfully show the sample website hosted.



**Step 22:** Now, we make some changes to the index.html file in the github.

For eg: If you make some changes to the <h2>tag.

Once the changes are committed, when the website is refreshed, the changes can be seen.

