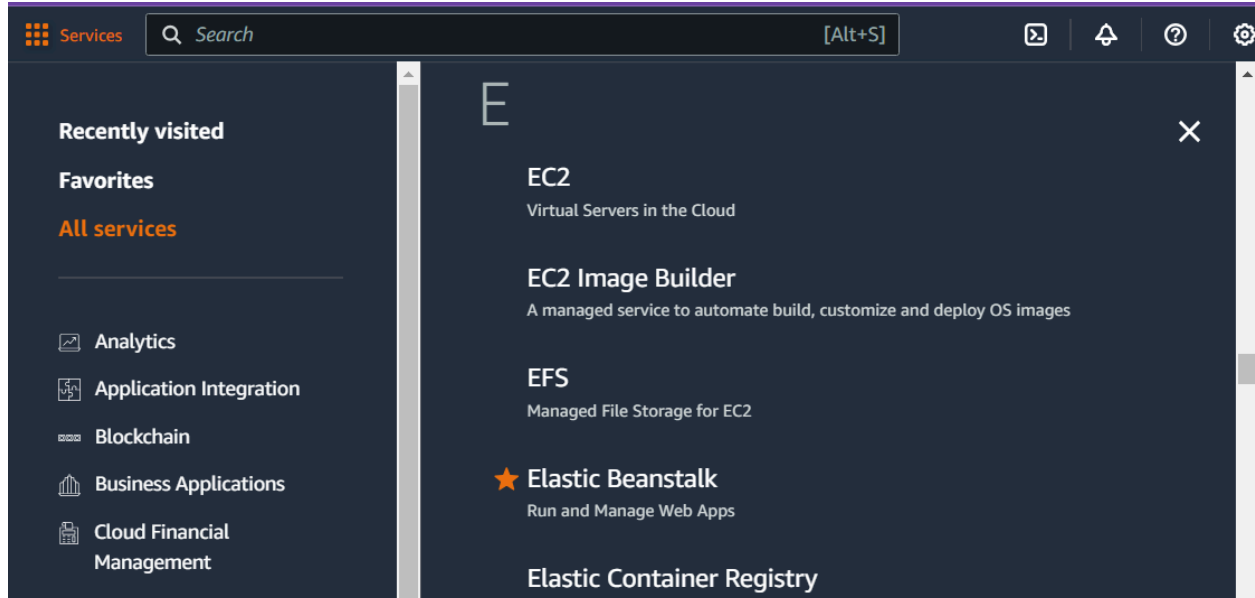


AdvDevOps Exp 2

Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

1. Login to your AWS account and search for Elastic Beanstalk in the search box.



2. Open the Elastic Beanstalk and name your web app as shown below.

A screenshot of the AWS Elastic Beanstalk 'Configure environment' page. The page is divided into a left sidebar with a step-by-step navigation menu and a main content area. The sidebar includes: 'Step 1: Configure environment' (selected), '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 environment' and contains two sections: 'Environment tier' and 'Application information'. The 'Environment tier' section has two radio button options: 'Web server environment' (selected) and 'Worker environment'. The 'Application information' section has a text input field for 'Application name' with the value 'Vedant' entered. Below the input field, it says 'Maximum length of 100 characters.' At the bottom, there is a link for 'Application tags (optional)'.

3. Automatically your environment name is set.

Environment information [Info](#)
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Vedant-env

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

Leave blank for autogenerated value

.us-east-1.elasticbeanstalk.com

Check availability

Environment description

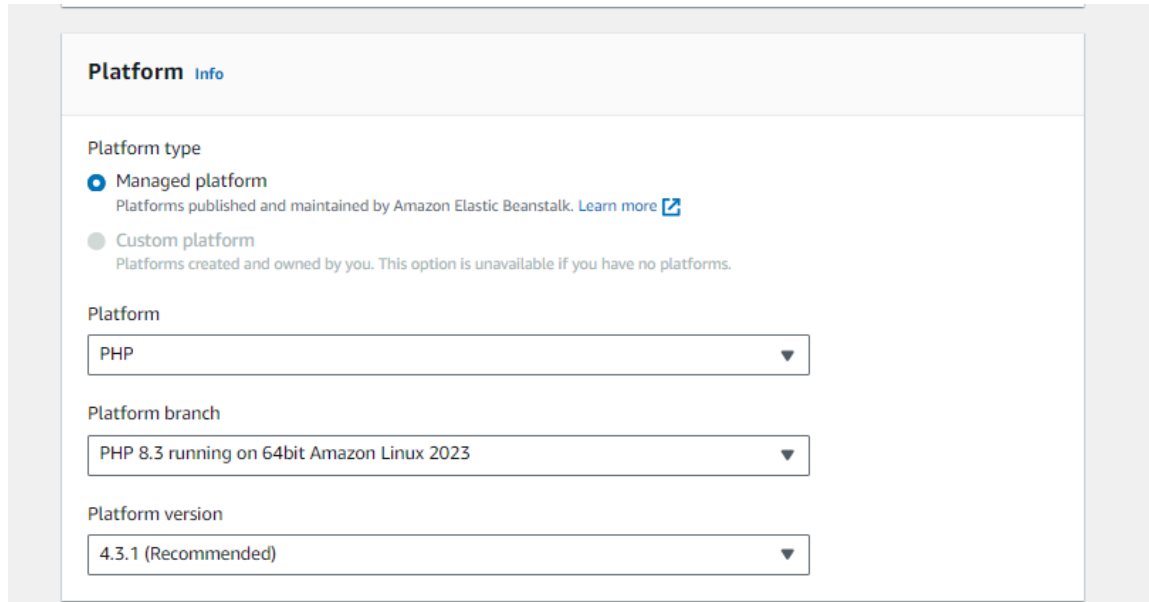
Platform [Info](#)

Platform type

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

☐ Custom platform

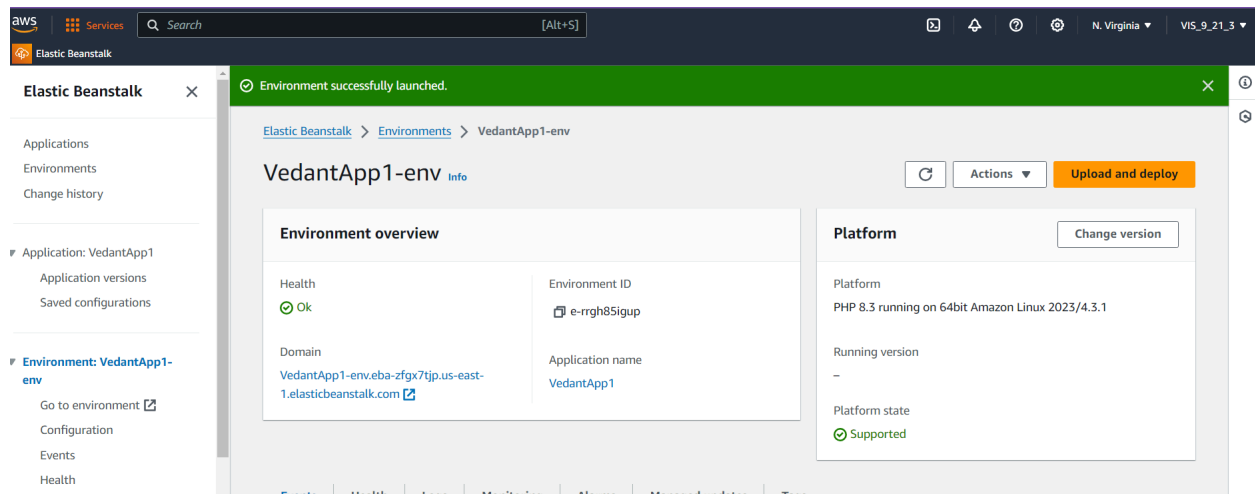
4. Choose PHP from the drop-down menu and then click Create Application.



The screenshot shows the 'Platform Info' configuration page in the AWS Elastic Beanstalk console. It includes the following fields:

- Platform type:** Radio buttons for 'Managed platform' (selected) and 'Custom platform'. A link 'Learn more' is next to 'Managed platform'.
- Platform:** A dropdown menu with 'PHP' selected.
- Platform branch:** A dropdown menu with 'PHP 8.3 running on 64bit Amazon Linux 2023' selected.
- Platform version:** A dropdown menu with '4.3.1 (Recommended)' selected.

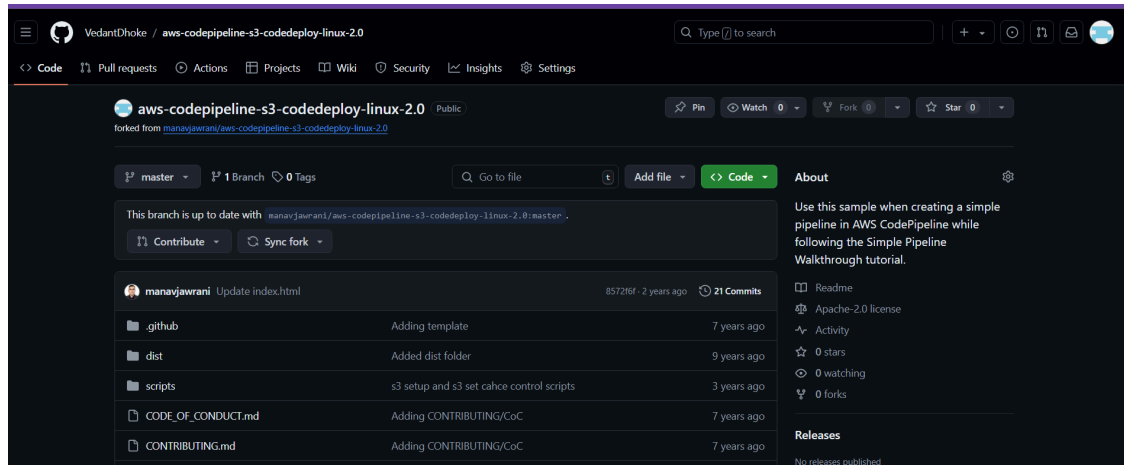
5. We can see that our Environment is created and launched successfully



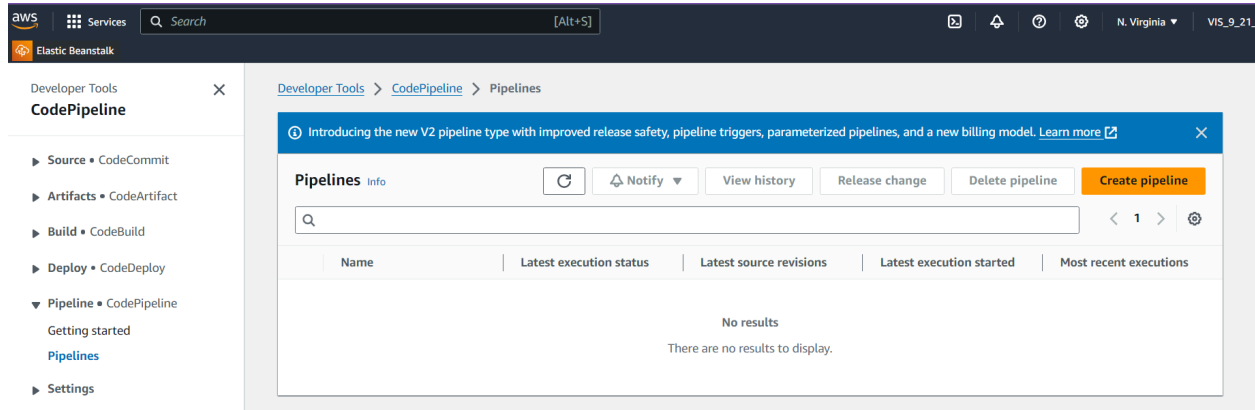
6. In this step, we'll obtain the sample code from a GitHub repository to host it later. The pipeline pulls the code from the source and executes specific actions.

To get started, navigate to the provided GitHub repository and fork it to your account. For this experiment, we will use a forked GitHub repository as our source.

We can see that in VedantDhoke repository the provided GitHub repository is forked.



7. Now we want to create a Pipeline. Navigate to Developer Tools -> CodePipeline and Click on create pipeline.



8. Give name to your pipeline and do the settings in step 1.

age

ge

age

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
Create a service role in your account

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

Role name

Type your service role name

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

9. In the Source Stage, Choose GitHub (Version 2) and click on Connect to GitHub.
You'll need your GitHub credentials to authorize and integrate AWS with your forked GitHub repository.

The image shows two screenshots from the AWS CodePipeline console. The top screenshot is the 'Add source stage' page, Step 2 of 5. It shows the 'Source' provider selected as 'GitHub (Version 2)'. A message indicates that a new GitHub version 2 action will be created, which uses GitHub Apps to access the repository. Below this, a connection is shown as 'Ready to connect' with the message 'Your GitHub connection is ready for use.' The bottom screenshot is the 'Connect to GitHub' page. It shows a notification about the July 1, 2024, update to connection creation. The 'GitHub connection settings' section shows the 'Connection name' as 'VedantConnect' and the 'GitHub Apps' ID as '53638499'. A 'Connect' button is visible at the bottom right.

er Tools > CodePipeline > Pipelines > Create new pipeline

pipeline settings

Step 2 of 5

Source stage

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-1:022499016110:connection/c8e5449b-9ft or [Connect to GitHub](#)

Ready to connect

Your GitHub connection is ready for use.

aws Services Search [Alt+S] N. Virginia

Elastic Beanstalk

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

Connection name

VedantConnect

GitHub Apps

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

53638499 or [Install a new app](#)

Tags - optional

[Connect](#)

10.Next, select the forked repository and the appropriate branch from the search box. Click Continue, then skip the build stage and proceed directly to the Deployment stage.

Repository name
Choose a repository in your GitHub account.

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.

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.

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.

11. Select Deploy and specify the application name and environment name, then click "Next." Review the details and create the pipeline.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Region

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.

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.

☐ Configure automatic rollback on stage failure

Cancel Previous Next

12. Review all the settings and click on create pipeline.

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

Review Info

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name

VedantPipeline

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-VedantPipeline

Step 4: Add deploy stage

Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

VedantApp1

EnvironmentName

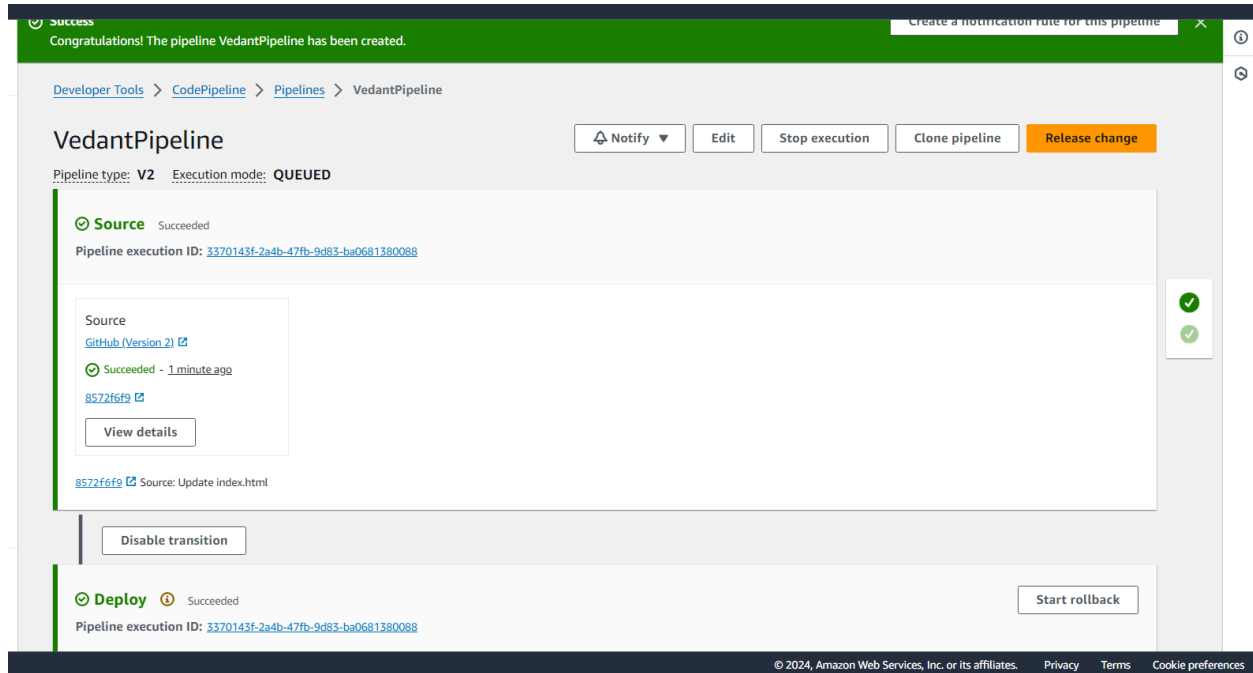
VedantApp1-env

Configure automatic rollback on stage failure

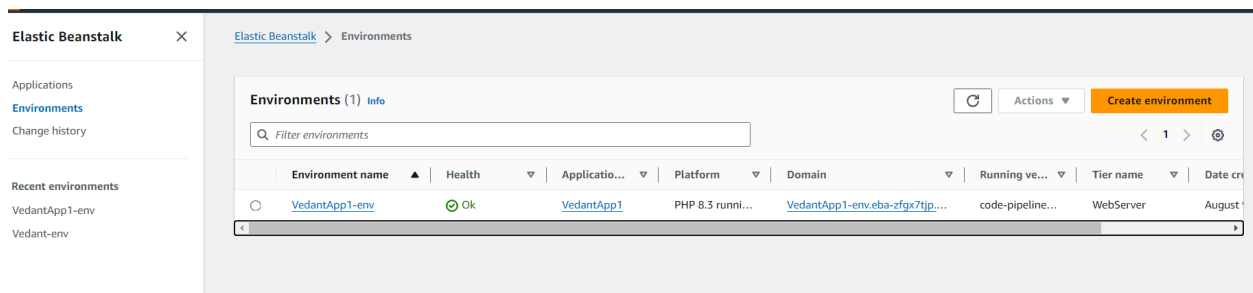
Disabled

[Cancel](#)[Previous](#)[Create pipeline](#)

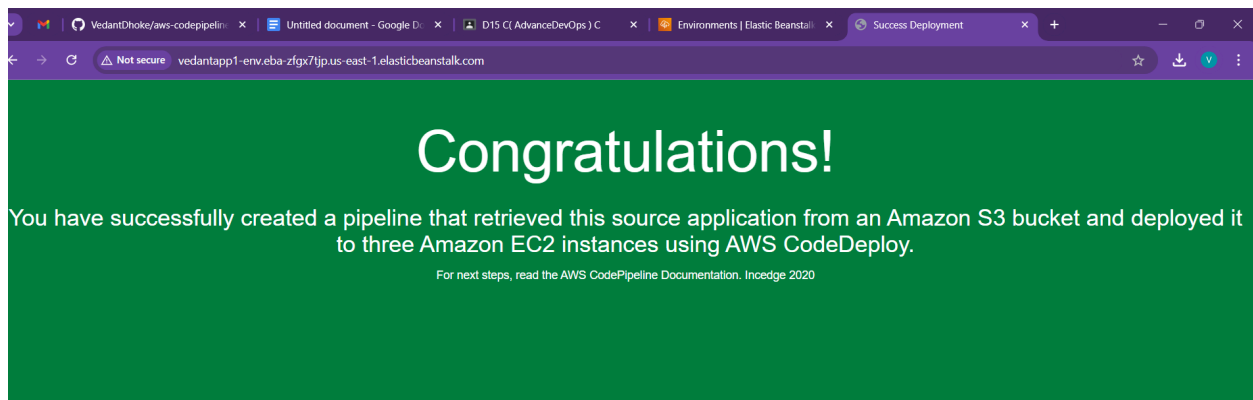
13. We can see our pipeline is successfully created. We need to ensure that both stages i.e Source and Deploy shows Succeeded as shown below.



14. Later Click on the URL which is provided in the Domain. It will navigate us to our website.

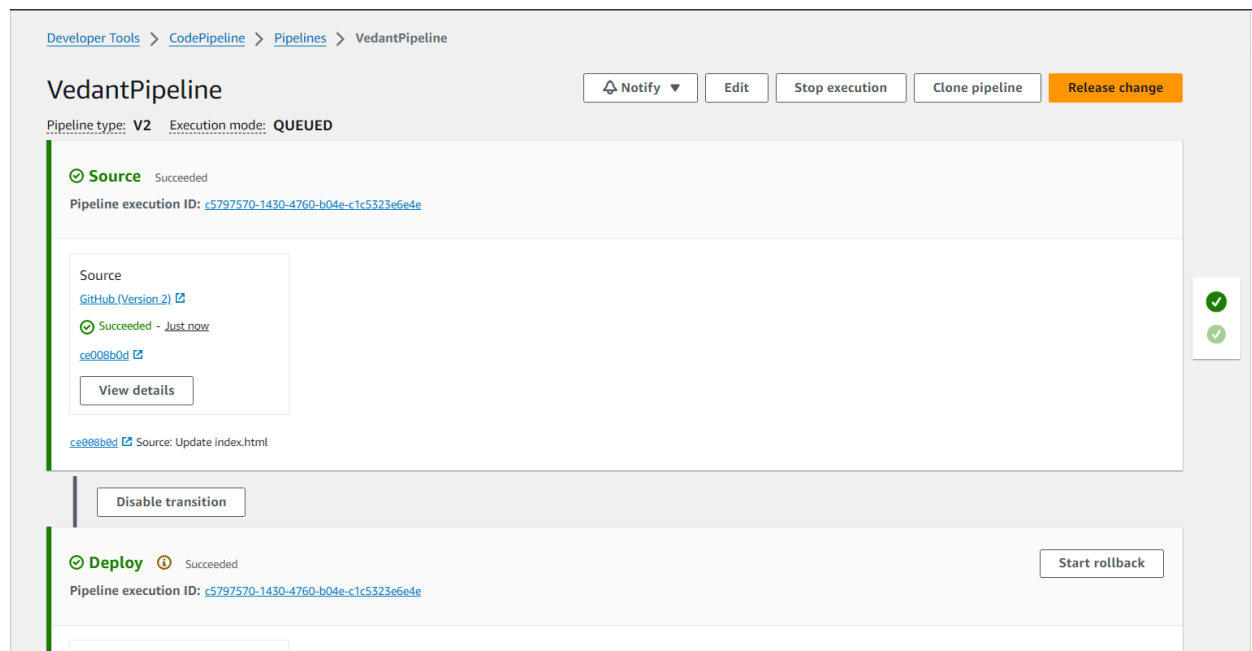


15. This is our sample website.



15. Committing the changes to update app .

1. Update the HTML file in your forked repository.
2. In GitHub, open index.html and modify the heading or paragraph tag. Commit the changes directly on GitHub.
3. Observe the real-time updates in the Source panel after committing.
4. Once the deployment is successful, view the changes on the website using the same URL.



16. Again click on the URL and we can see our Updated App.

