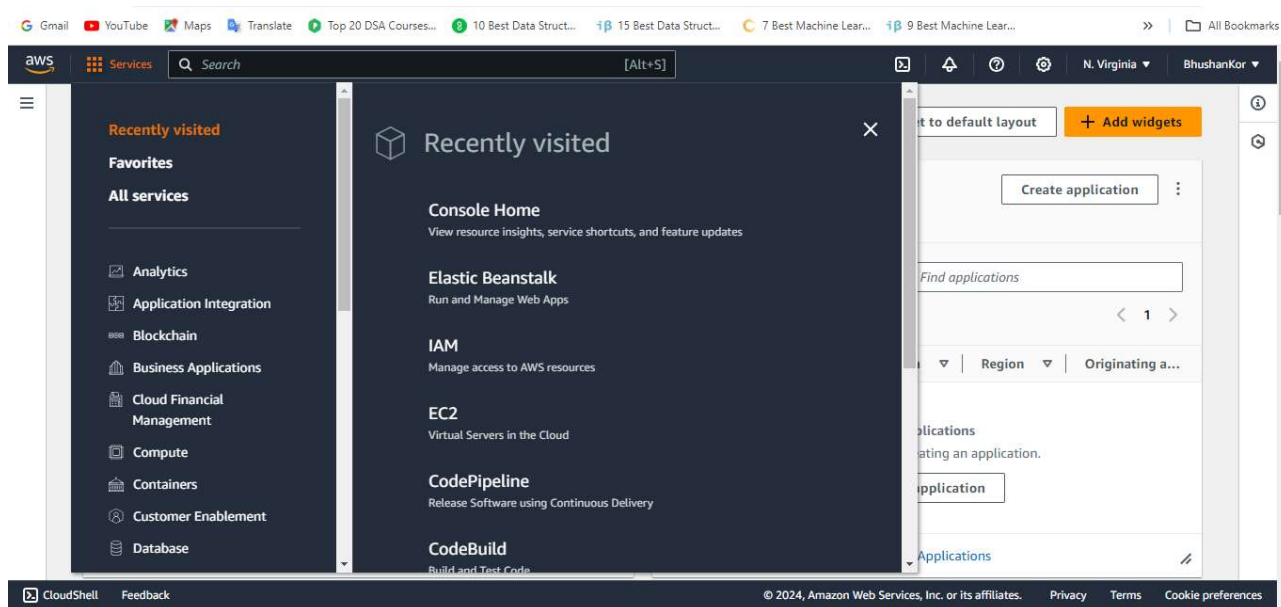
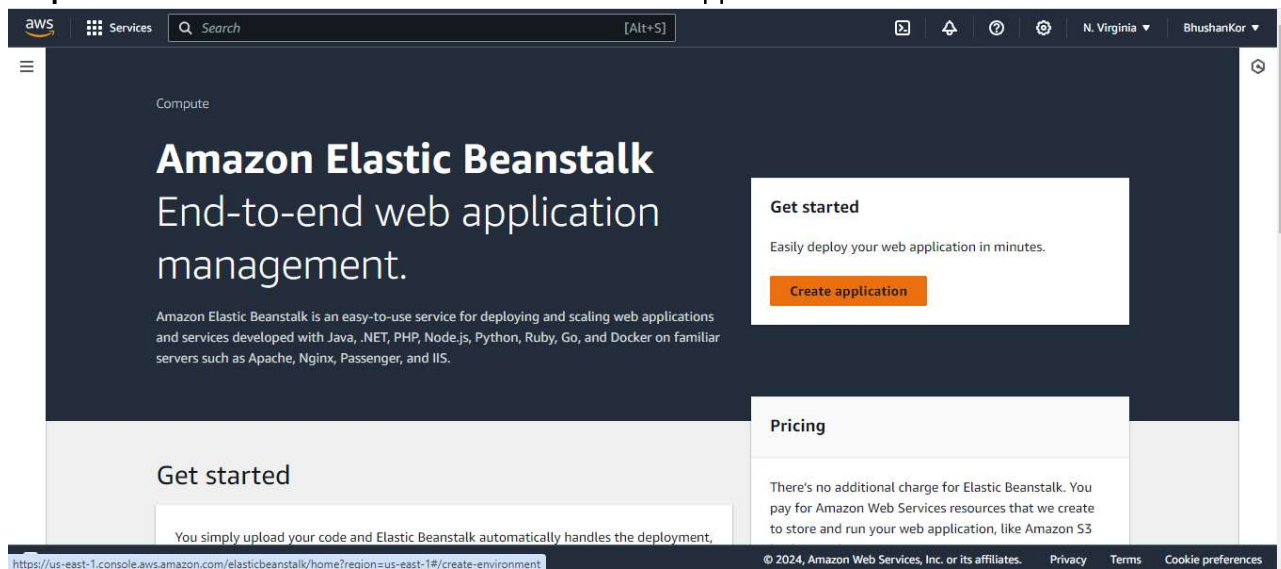


Step 1: Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services.



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

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

Platform branch

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.2 (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.

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 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

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.

kaushalrole1

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

k11

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

kaushalrole1

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

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

Review Info

Step 1: Configure environment

Environment information

Environment tier	Application name
Web server environment	kaushalapp
Environment name	Application code
Kaushalapp-env	Sample application
Platform	
amazonelasticbeanstalk-us-east-1:platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

Step 2: Configure service access

Service access Info
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
arn:aws:iam::010928205712:role/kaushalrole1	k11	kaushalrole1

Step 3: Set up networking, database, and tags

Networking, database, and tags Info

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	
false	false	
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		
<div><div>Key</div><div>Value</div><div>No environment properties There are no environment properties defined</div></div>		
<div>Cancel</div> <div>Previous</div> <div>Submit</div>		

Environment successfully launched.

Elastic Beanstalk > Environments > Kaushalsapp-env

Kaushalsapp-env

Info

Actions

Upload and deploy

Environment overview

Health
Warning

Domain
Kaushalsapp-env.eba-2ex22i22.us-east-1.elasticbeanstalk.com

Environment ID
e-5rjwzivmxy

Application name
kaushalsapp

Platform

Change version

Platform
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2

Running version
-

Platform state
Supported

Events | Health | Logs | Monitoring | Alarms | Managed updates | Tags

Events (10) Info

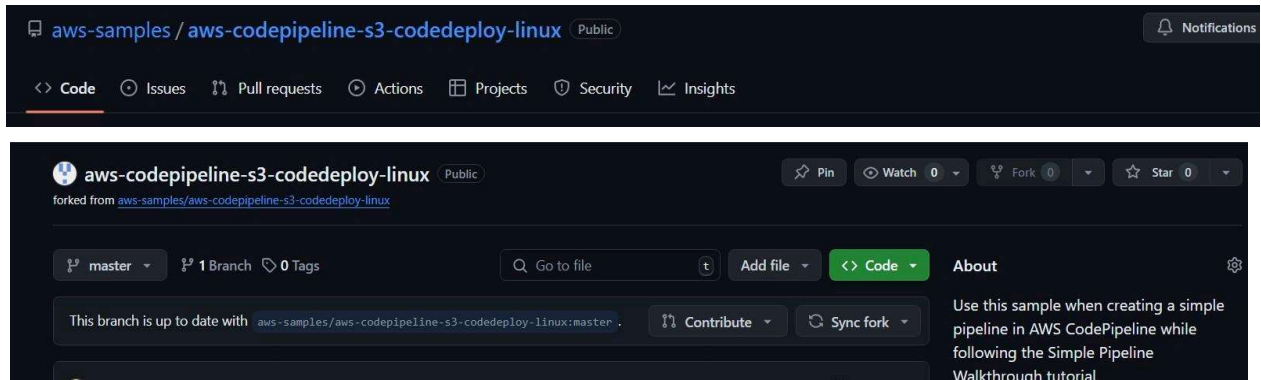
Filter events by text, property or value

Time	Type	Details
August 16, 2024 21:46:10 (UTC+5:30)	INFO	Successfully launched environment: Kaushalsapp-env
August 16, 2024 21:46:08 (UTC+5:30)	INFO	Application available at Kaushalsapp-env.eba-2ex22i22.us-east-1.elasticbeanstalk.com
August 16, 2024 21:45:52 (UTC+5:30)	INFO	Instance deployment completed successfully.
August 16, 2024 21:45:47 (UTC+5:30)	INFO	Instance deployment: You didn't include a 'composer.json' file in your source bundle. The deployment didn't install Composer dependencies.
August 16, 2024 21:45:33 (UTC+5:30)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 12 seconds). There are no instances.

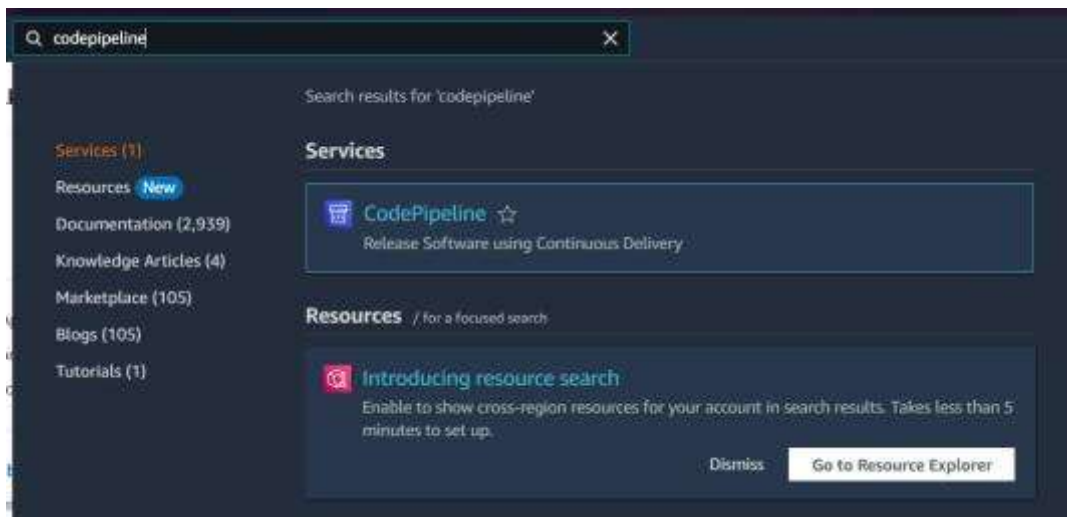
© 2024 Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies

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.

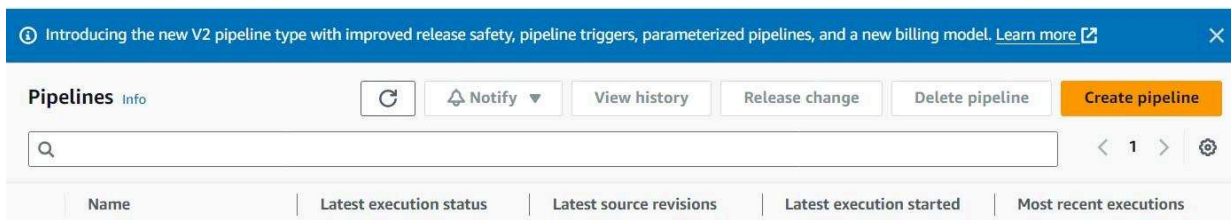
<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>



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

Step 1 of 5

Pipeline settings

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

kaushalpipeline

No more than 100 characters.

Pipeline type

i 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

AWSCodePipelineServiceRole-us-east-1-kaushalpipeline

Type your service role name

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

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

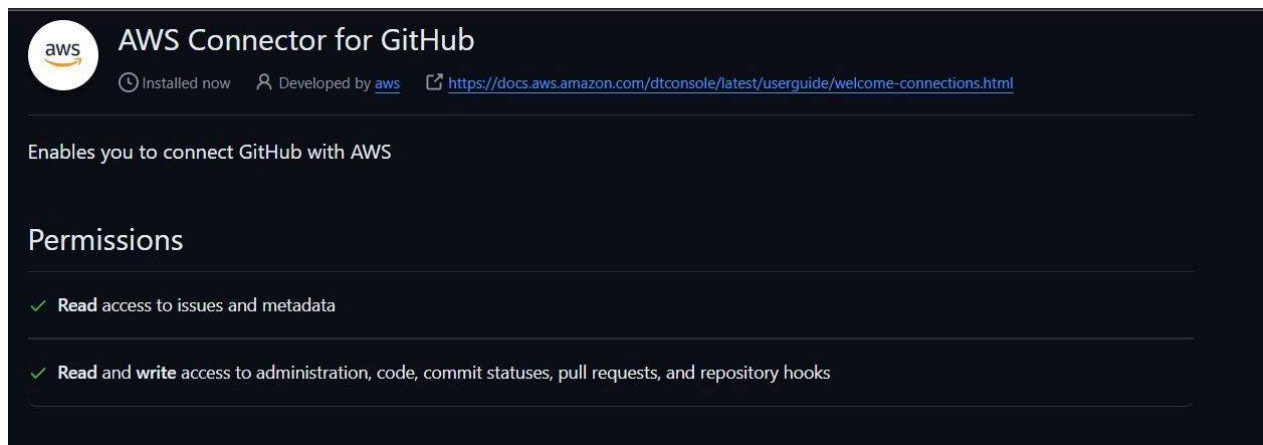
The screenshot shows the 'Add source stage' screen in the AWS CodePipeline console. The breadcrumb trail is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a progress list: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main heading is 'Add source stage' with an 'Info' link. Below the heading, it says 'Step 2 of 5'. The 'Source' section has a 'Source provider' dropdown set to 'GitHub (Version 2)'. A blue information box states: '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'. The 'Connection' section prompts to 'Choose an existing connection that you have already configured, or create a new one and then return to this task.' It features a search input, a 'Connecting' button, and a 'Repository name' search input.

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

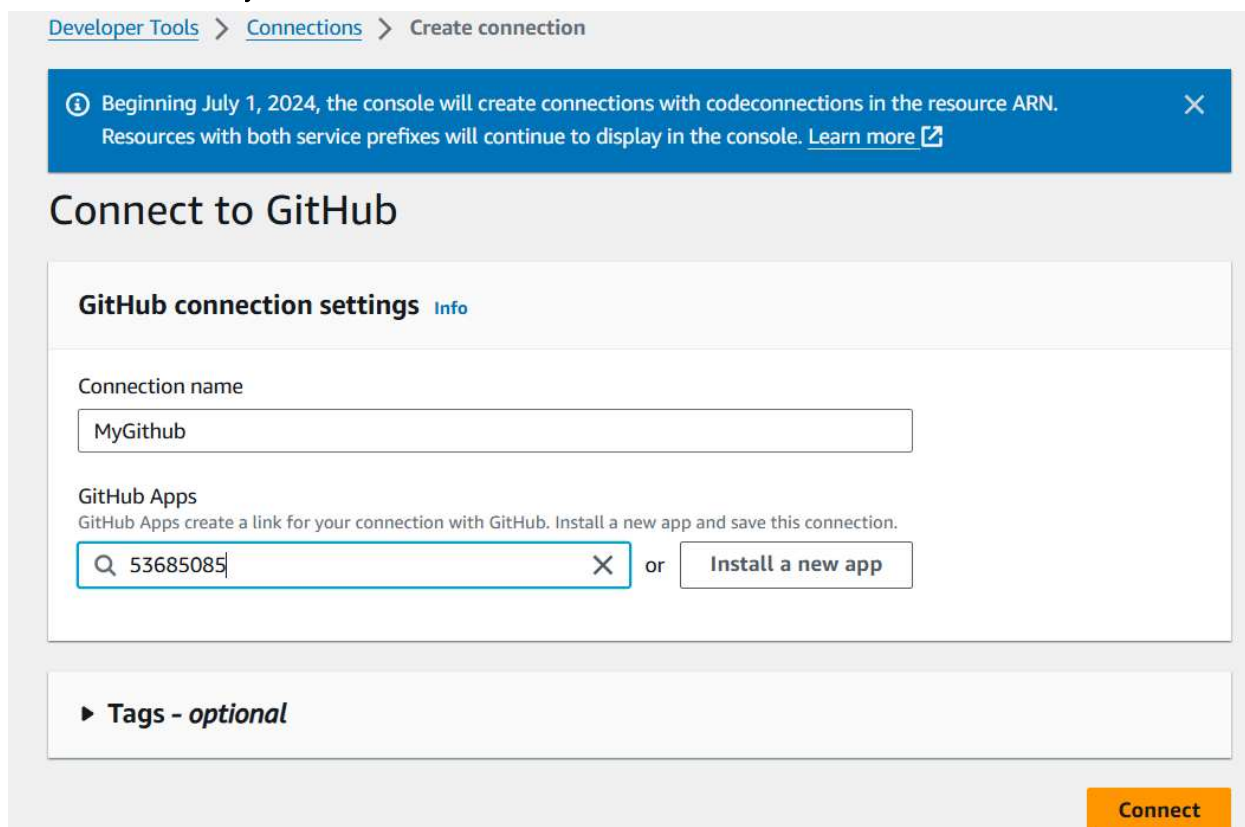
The screenshot shows the 'Create connection' screen in the AWS CodePipeline console. The breadcrumb trail is 'Developer Tools > Connections > Create connection'. The heading is 'Create a connection' with an 'Info' link. The 'Create GitHub App connection' section has a 'Connection name' input field containing 'MyGithub'. Below this is a section for 'Tags - optional'. At the bottom right, there is a yellow 'Connect to GitHub' button.

The screenshot shows the 'Connect to GitHub' dialog box. At the top, a blue banner states: 'Beginning July 1, 2024, the console will create connections with codecommit@aws.com in the resource ARN. Resources with both service profiles will continue to display in the console. Learn more >'. The dialog title is 'Connect to GitHub'. It contains a 'GitHub connection settings' section with a 'Connection name' input field (containing 'MyGithub') and a 'GitHub App' section with a search input and an 'Install a new app' button. At the bottom right, there is a yellow 'Connect' button.

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



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



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).

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 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 step in the AWS CodePipeline console. At the top, there are two radio button options: 'CodePipeline default' (selected) and 'Full clone'. Below this is a section titled 'Trigger' with a sub-section 'Trigger type'. It contains three radio button options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'. 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.

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' step 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 'Build - optional'. It contains a section 'Build provider' with a dropdown menu. 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

Choose pipeline settings Step 4 of 5

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

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.

Q kaushalsapp X

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 Kaushalsapp-env X

☐ Configure automatic rollback on stage failure

Cancel Previous Next

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

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

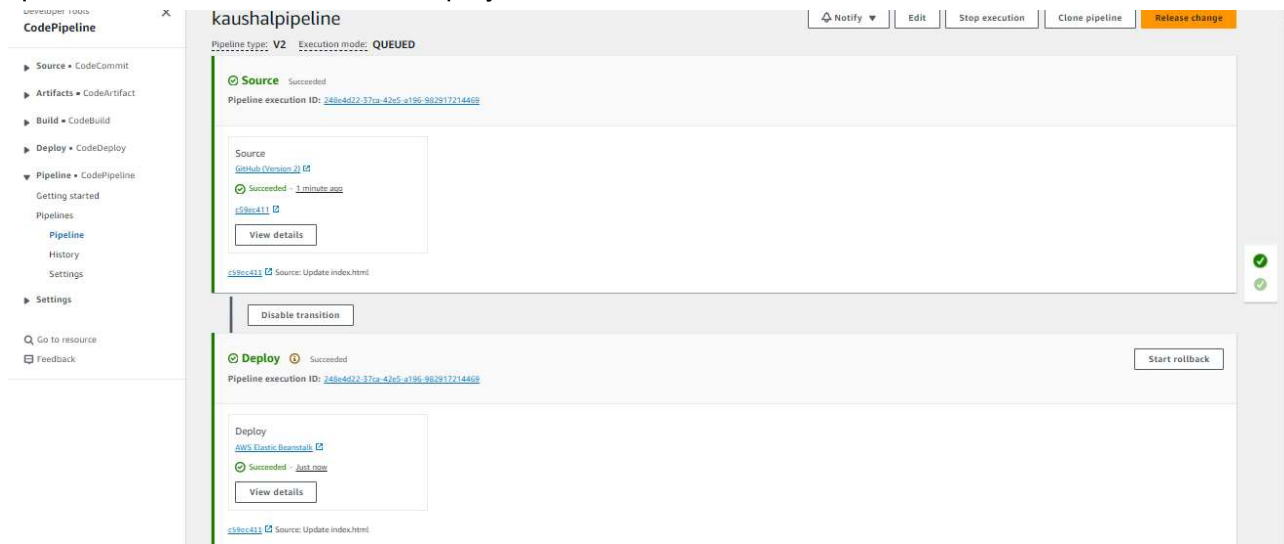
ApplicationName
kaushalsapp

EnvironmentName
Kaushalsapp-env

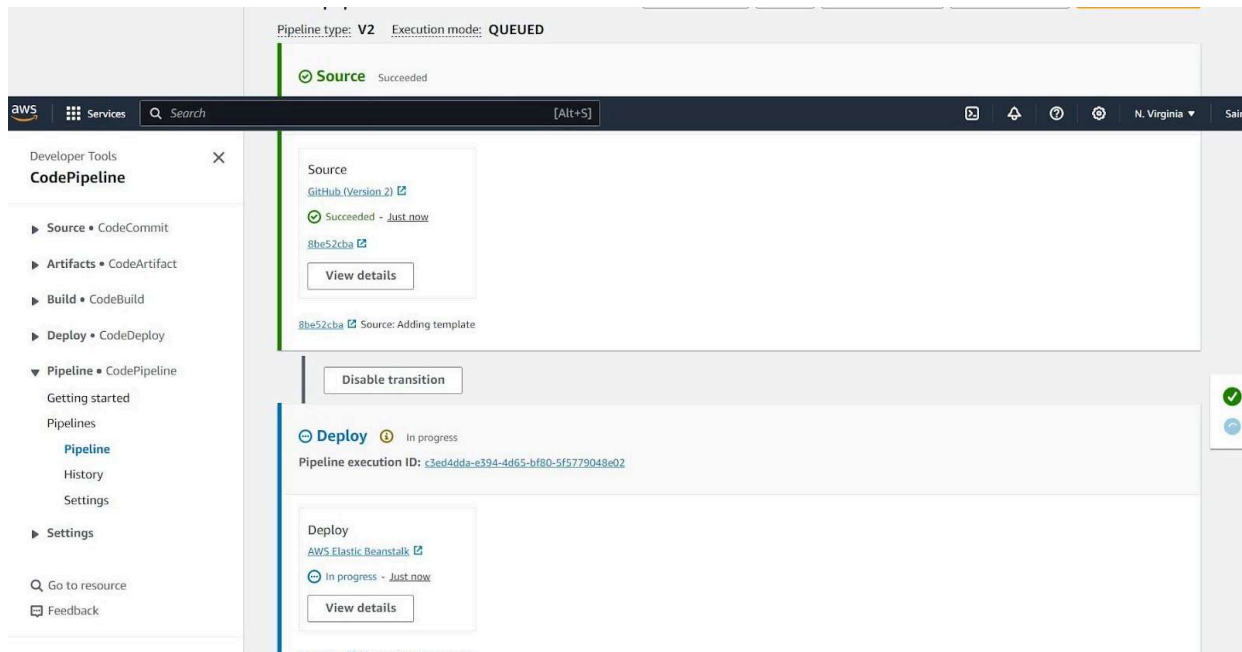
Configure automatic rollback on stage failure
Disabled

Cancel Previous Create pipeline

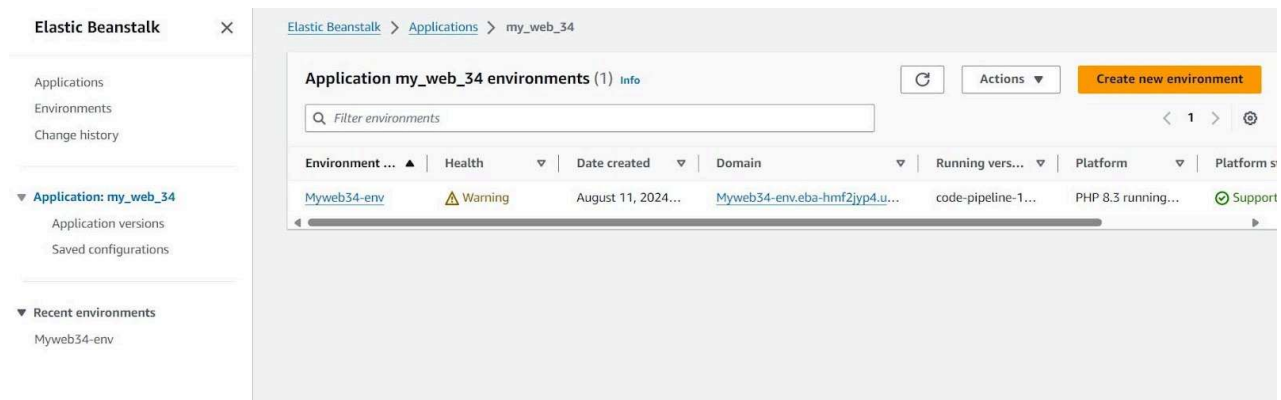
Step 18: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy.



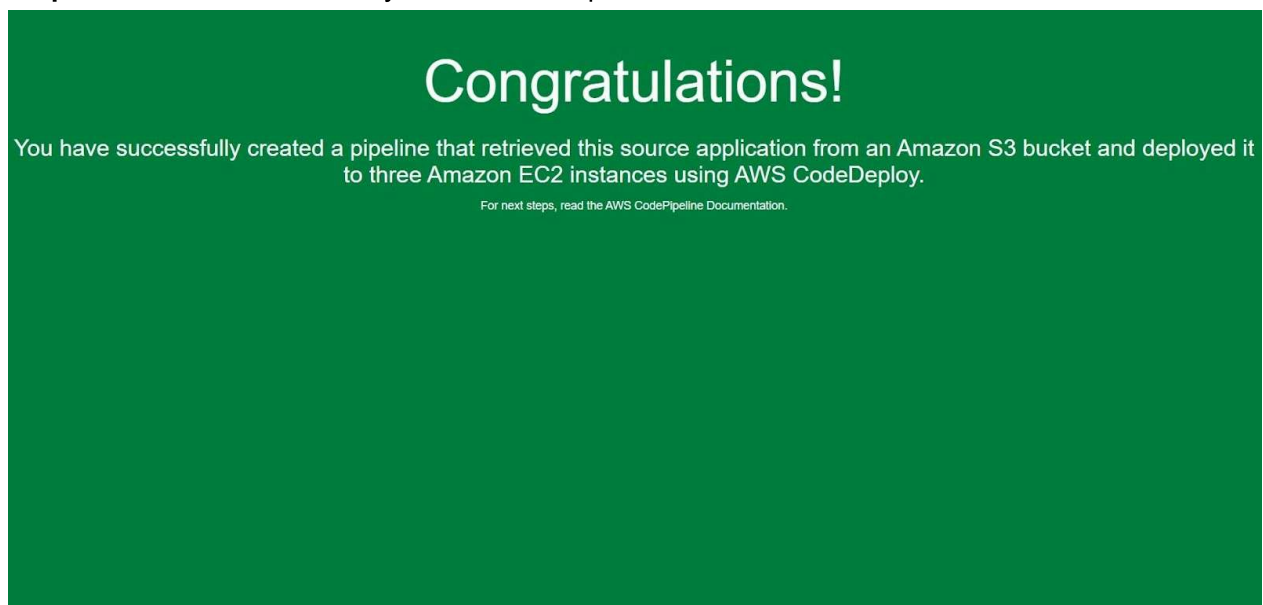
Step 19: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



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.

