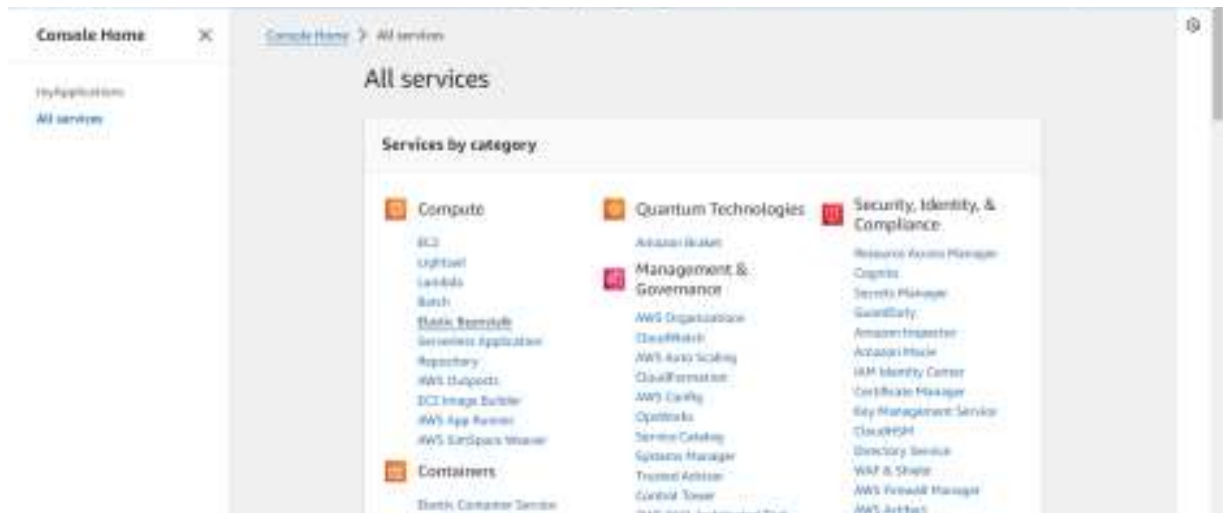


Experiment No :2

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.

The screenshot shows the 'Configure environment' page in the AWS IAM console. On the left, a sidebar lists steps: Step 1: Configure environment (selected), Step 2: Configure service system, Step 3: Review, Step 4: Review, Step 5: Review, 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 buttons: 'Web server environment' (selected) and 'Worker environment'. The 'Application information' section has a text input field for 'Application name' containing 'FirstWebApp' and a button for 'Application tags (optional)'.

The screenshot shows the 'Environment information' section. It has a text input field for 'Environment name' containing 'FirstWebApp-env'. Below it, a text input field for 'Domain' contains 'iam-test-1.amazonaws.com'. To the right of the domain field is a 'Check availability' button. Below these fields is a text input field for 'Environment description'.

The screenshot shows the 'Platform' section. It has a radio button for 'Managed platform' (selected) and a radio button for 'Custom platform'. Below the radio buttons are three dropdown menus: 'Platform' (set to 'PHP'), 'Platform branch' (set to 'PHP 5.3 running on 64bit Amazon Linux 2013'), and 'Platform version' (set to '3.3.1 (Recommended)').

The screenshot shows the 'Application code' section with two options: 'Sample application' (selected) and 'Upload your code'. Below this is the 'Presets' section, which includes a description and a list of configuration presets. The 'Single instance (free tier eligible)' preset is selected. At the bottom, there are 'Cancel' and 'Next' buttons.

Application code [info](#)

☒ **Sample application**
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](#)

☐ **Upload your code**
Upload a source bundle from your computer or repository or use Elastic Beanstalk S3

Presets [info](#)

Start from a preset that matches your use case or choose custom configuration to tailor recommendations when you set the service's default values.

Configuration presets

☒ **Single instance (free tier eligible)**

☐ Single instance (long spot instance)

☐ High availability

☐ High availability (using spot and on-demand instances)

☐ Custom configuration

Cancel Next

Use an existing service role and choose whatever service role is present on your account

The screenshot shows the 'Configure service access' step in the AWS IAM console. The 'Service role' section has 'Use an existing service role' selected. The 'Existing service roles' dropdown shows 'LabRole'. The 'EC2 key pair' dropdown shows 'yoddy'. The 'EC2 instance profile' dropdown shows 'LabInstanceProfile'. At the bottom, there are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

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

LabRole

EC2 key pair

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

yoddy

EC2 instance profile

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

LabInstanceProfile

Cancel Skip to review Previous Next

Step 5: Click on Skip to Review

This screenshot is identical to the previous one, showing the 'Configure service access' step. The 'Skip to review' button is highlighted in the bottom navigation bar.

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

LabRole

EC2 key pair

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

yoddy

EC2 instance profile

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

LabInstanceProfile

View permission details

Cancel Skip to review Previous Next

Review the settings that you have set up for your application and submit your application

The screenshot shows the 'Review' page for Step 1: Configure environment. The left sidebar lists steps: Step 1 (selected), Step 2, Step 3, Step 4, and Step 5. The main content area is titled 'Review' and 'Step 1: Configure environment'. It contains a table for 'Environment information' with the following data:

Environment ID	Application name
Web server environment	FirstWebApp
Environment name	Application code
FirstWebApp-env	Sample application
Platform	
aws-elasticbeanstalk-ec2-1:platform/PHP-6.5.3 running on 64bit Amazon Linux 2023/4.3.1	

Below the table is a 'Step 2: Configure service access' section with an 'Edit' button.

The screenshot shows the 'Review' page for Step 2: Configure service access. The left sidebar lists steps: Step 1, Step 2 (selected), Step 3, Step 4, and Step 5. The main content area is titled 'Review' and 'Step 2: Configure service access'. It contains a table for 'Environment properties' with the following data:

Property	Value
Deployment timeout	300
Memory limit	256M
Log retention	7
X-Ray enabled	Disabled
Document root	-
ZIP output compression	Off
Auto logs	Disabled
Max execution time	60
Proxy server	nginx
Update level	minor

Below the table is a 'Environment properties' section with a 'Key' and 'Value' input field. A message states: 'No environment properties. There are no environment properties defined.' At the bottom are 'Cancel', 'Previous', and 'Submit' buttons.

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar lists 'Applications', 'Environments', and 'Change history'. The main content area is titled 'FirstWebApp-env' and shows the 'Environment overview' section. It contains a table for 'Environment overview' with the following data:

Health	Environment ID
On	FirstWebApp-env
Source	Application name
FirstWebApp-ec2-elasticbeanstalk-ec2-1:platform/PHP-6.5.3	FirstWebApp

Below the table is a 'Platform' section with a 'Change version' button. The platform details are:

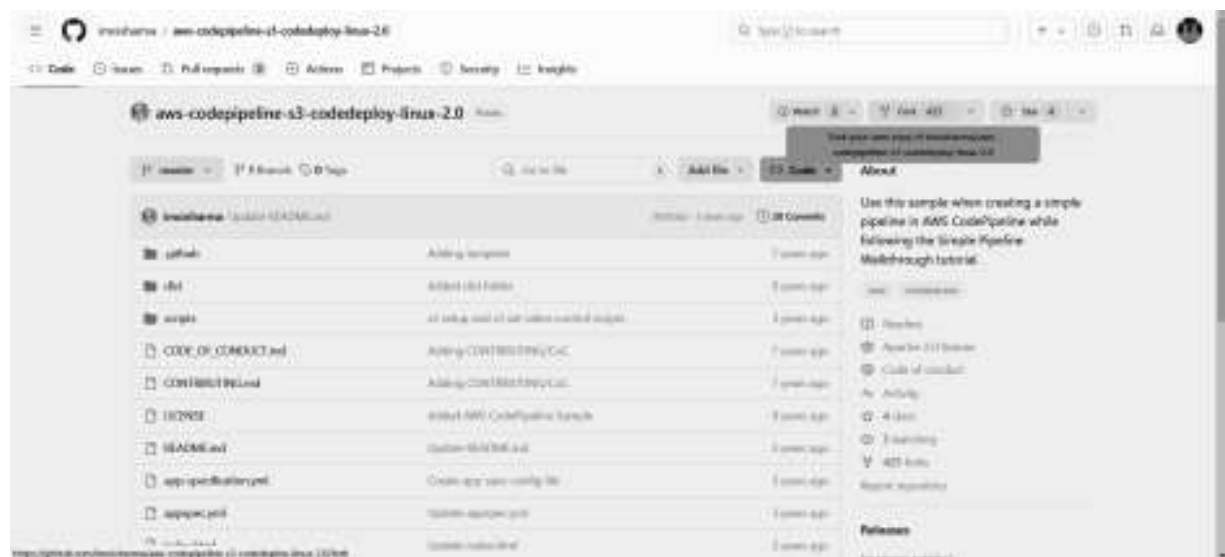
Platform
PHP 6.5.3 running on 64bit Amazon Linux 2023/4.3.1
Running version
-
Platform state
Supported

At the bottom are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'.



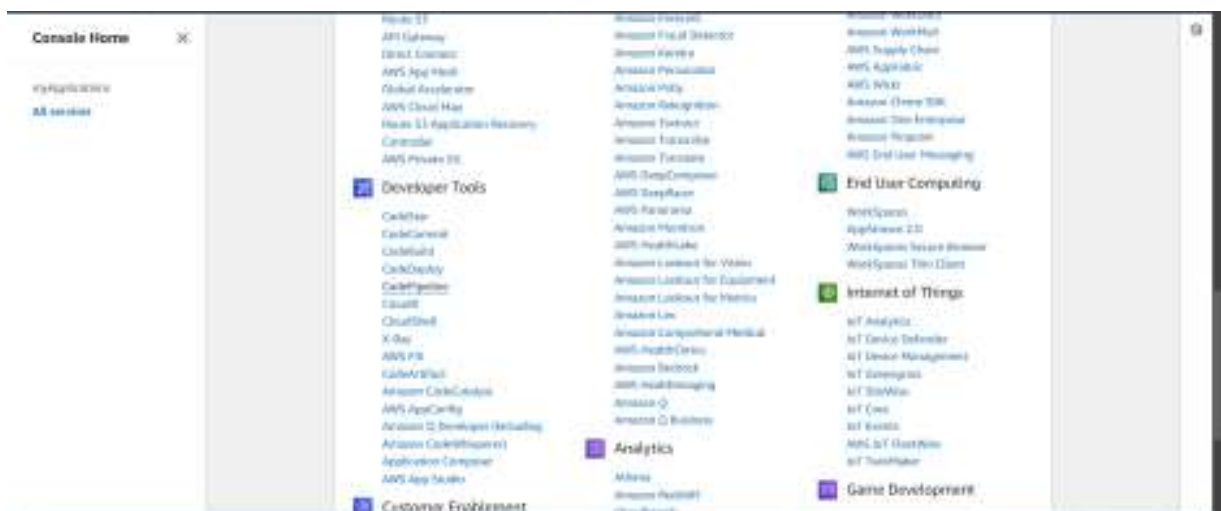
Step 7 : Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

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

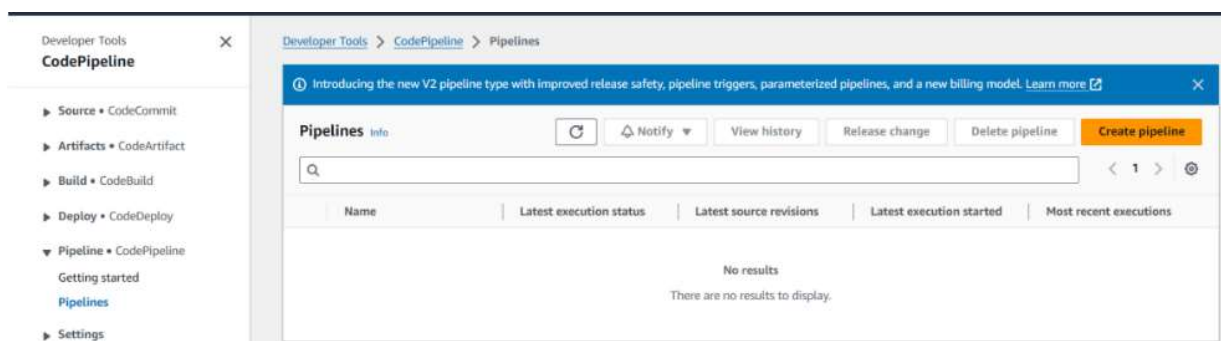




Step 8: Search CodePipeline in the services tab and click on it.



Step 9: Click on Create Pipeline.



Step 10: Give a name to your Pipeline. A new service role would be created with the name of the pipeline.

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-MyPipeline

Type your service role name

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

Step 11: Select a source provider (as GitHub Version (2)). Click on connect to Github

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.

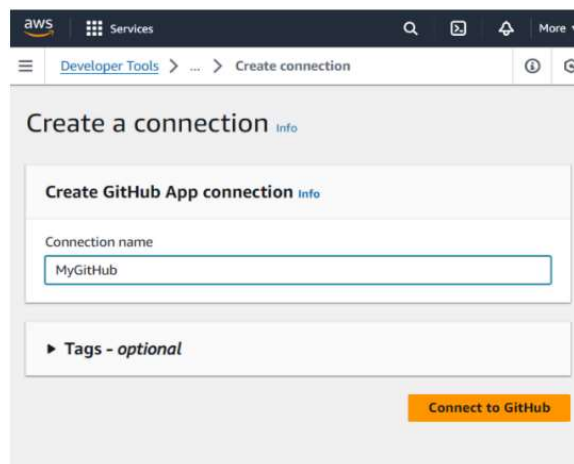
or

Repository name
Choose a repository in your GitHub account.

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.

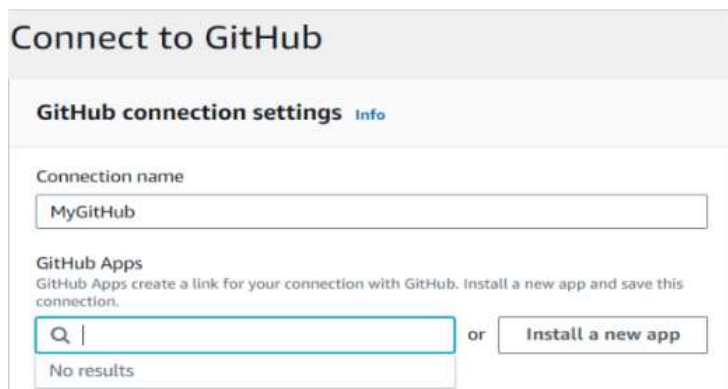
Step 12: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app.



The screenshot shows the AWS Developer Tools interface. The breadcrumb trail is 'Developer Tools > ... > Create connection'. The main heading is 'Create a connection' with an 'Info' link. Below this is a section titled 'Create GitHub App connection' with another 'Info' link. A text input field for 'Connection name' contains the text 'MyGitHub'. Below the input field is a section for 'Tags - optional'. At the bottom right of this section is an orange button labeled 'Connect to GitHub'.

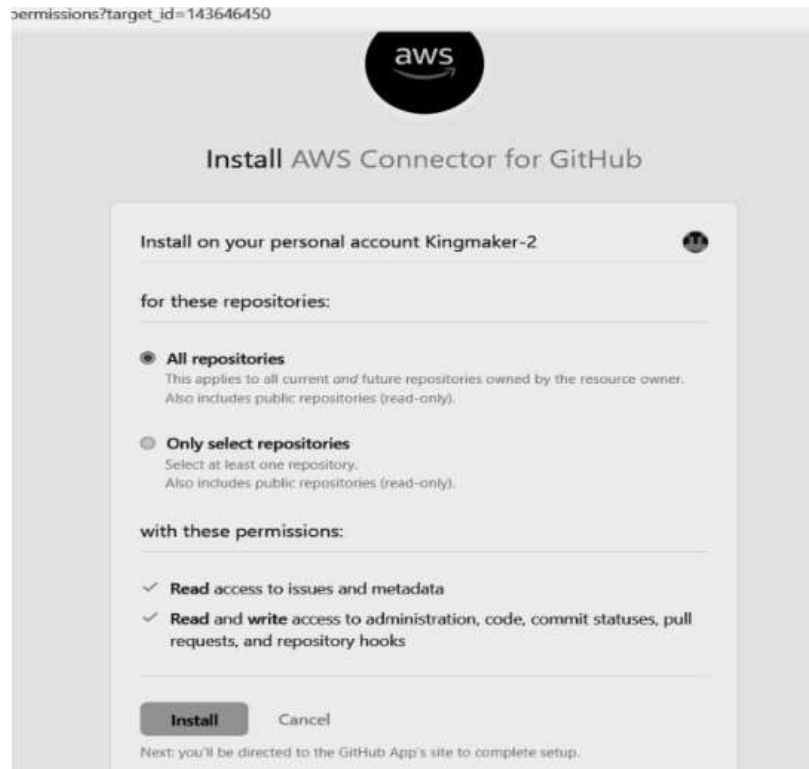


The screenshot shows an authorization prompt titled 'AWS Connector for GitHub by Amazon Web Services would like permission to:'. It lists three permissions: 'Verify your GitHub identity (Kingmaker-2)', 'Know which resources you can access', and 'Act on your behalf' with a 'Learn more' link. Below the permissions is a link 'Learn more about AWS Connector for GitHub'. At the bottom are two buttons: 'Cancel' and 'Authorize AWS Connector for GitHub'. Below the buttons, it says 'Authorizing will redirect to https://redirect.codestar.aws'.

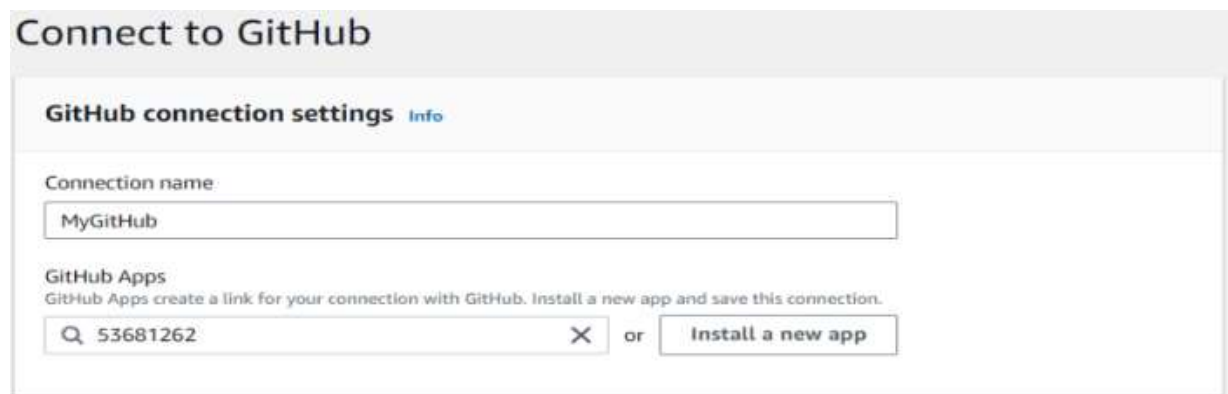


The screenshot shows the 'Connect to GitHub' page. The heading is 'Connect to GitHub'. Below it is a section titled 'GitHub connection settings' with an 'Info' link. A text input field for 'Connection name' contains the text 'MyGitHub'. Below this is a section titled 'GitHub Apps' with the text 'GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.' There is a search input field with a magnifying glass icon and the text 'No results' below it. To the right of the search field is the text 'or' and a button labeled 'Install a new app'.

Step 13: This will direct you to install AWS connector on your GitHub .Install it to your account and give it its permissions



Step 14: 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.



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.

Q `arn:aws:codeconnections:us-east-1:011528263337:connection/b7859e8a-5f1` X

or

Connect to GitHub

Step 15: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

Repository name

Choose a repository in your GitHub account.

Q `Kingmaker-2/aws-codepipeline-s3-codedeploy-linux-2.0` 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

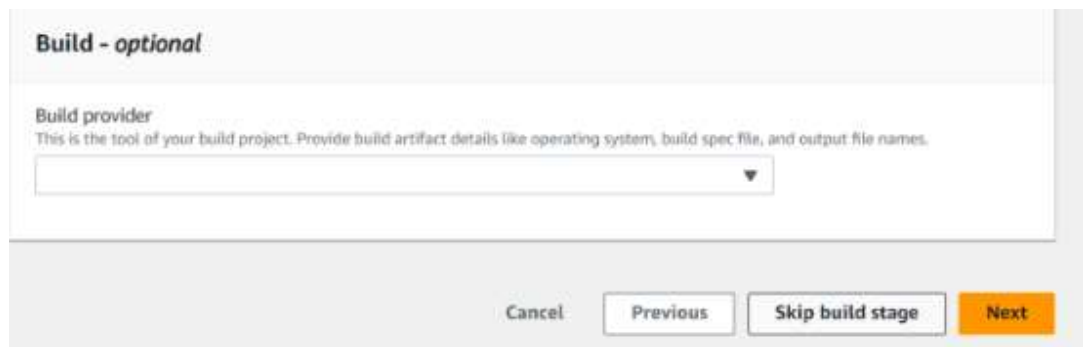
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Step 16: 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. The title is 'Trigger'. Below it, the section is 'Trigger type' with the instruction 'Choose the trigger type that starts your pipeline.' There are three radio button options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'. Each option has a description: 'No filter' starts the pipeline on any push and clones the HEAD; 'Specify filter' starts the pipeline on a specific filter and clones the exact commit, requiring Pipeline type V2; 'Do not detect changes' means not to automatically trigger the pipeline.

Step 17: 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 'Build - optional' configuration screen. The title is 'Build - optional'. Below it, the section is 'Build provider' with the instruction '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 selecting the build provider. At the bottom, there are four buttons: 'Cancel', 'Previous', 'Skip build stage', and 'Next'.

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 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

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.

FirstWebApp

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.

FirstWebApp-env

☒ Configure automatic rollback on stage failure

Cancel

Previous

Next

Step 18: Check all the information and click on create Pipeline

Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
FirstWebApp

EnvironmentName
FirstWebApp-env

Configure automatic rollback on stage failure
Enabled

Cancel Previous **Create pipeline**

Step 19: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy

Developer Tools **CodePipeline**

Success
Congratulations! The pipeline MyPipeline1 has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > MyPipeline1

MyPipeline1 Notify Edit Stop execution Clone pipeline Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: [c33b1771-5488-423a-853a-57f5d81245d4](#)

Source
[GitHub \(Repository\)](#)
Succeeded · Add new
[@c33b1771](#)
View details

Source Update READING.txt

Deploy Succeeded

Pipeline execution ID: [c33b1771-5488-423a-853a-57f5d81245d4](#)

Deploy
[AWS Elastic Beanstalk](#)
Succeeded · Add new
View details

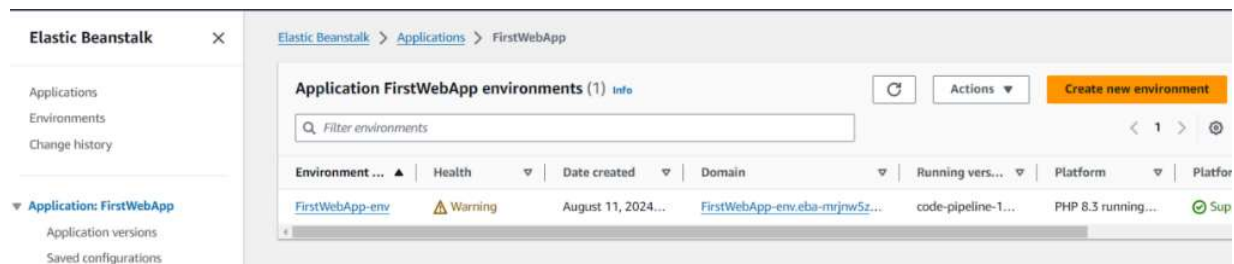
Source Update READING.txt

Start rollback

Step 20: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



Step 21: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain



Step 22: This will successfully show the sample website hosted.

