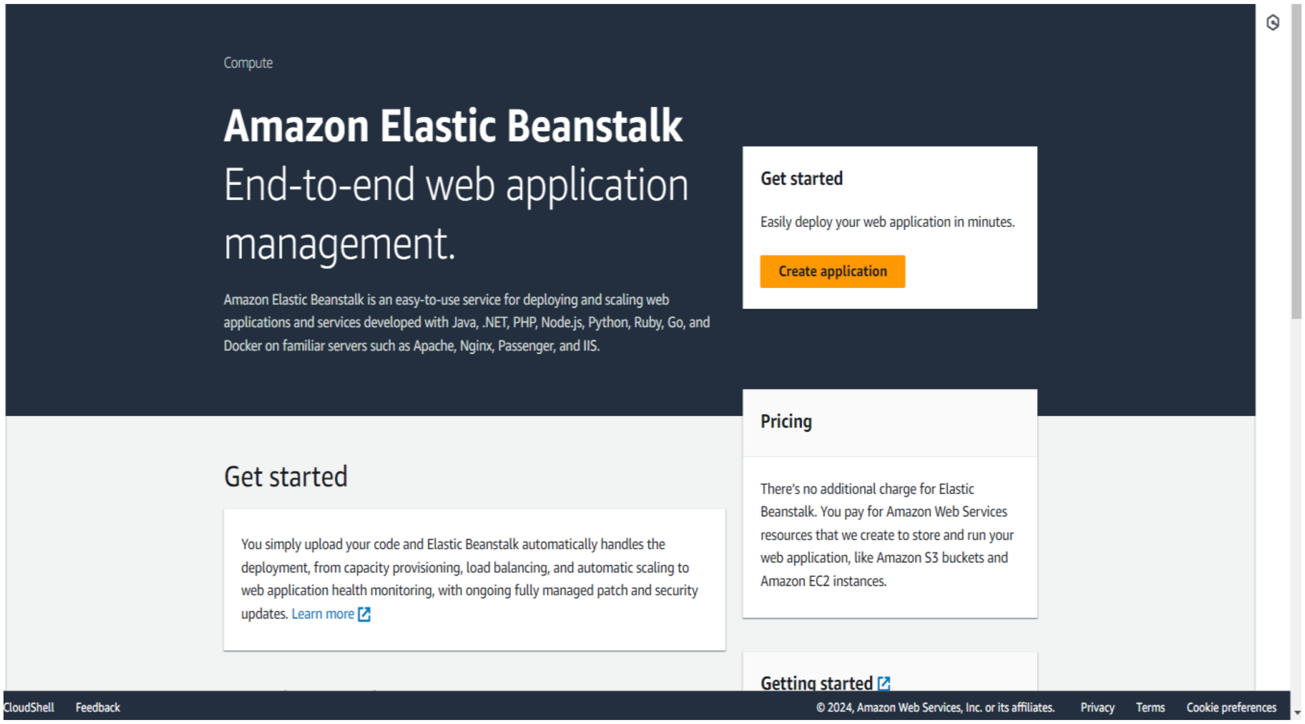
Name: Shubham Jha
Div: D15C
Roll No:19

Experiment 2 : Elastic Beanstalk

1) Go to services and choose elastic Beanstalk. following page will appear.



Compute

Amazon Elastic Beanstalk
End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create application

Pricing

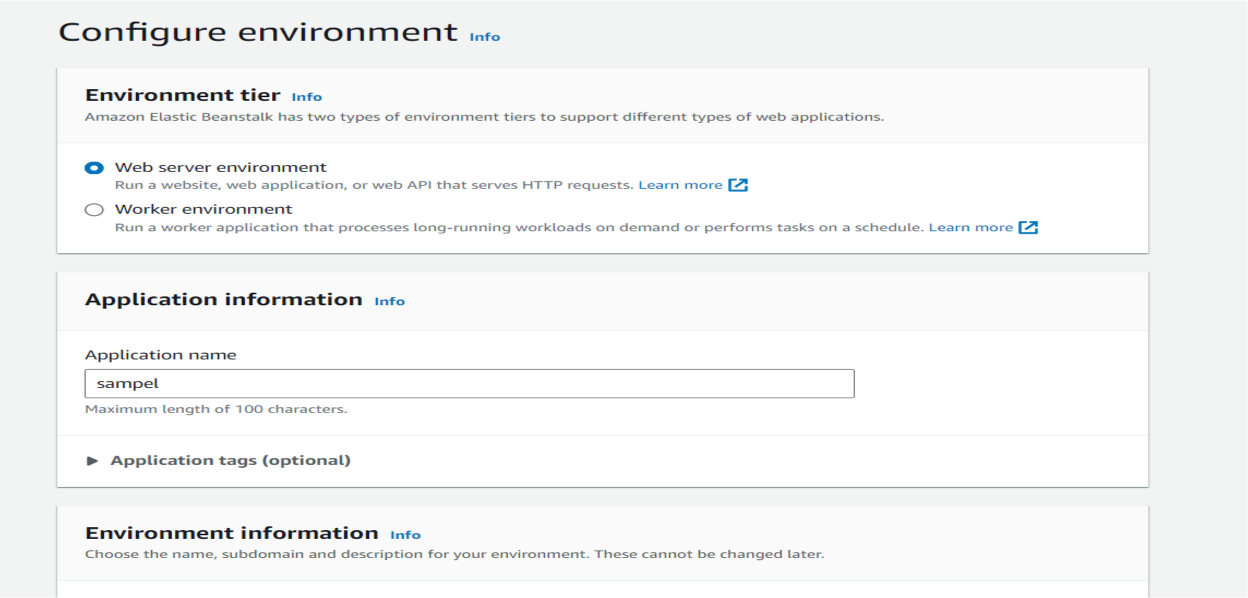
There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. Learn more

Getting started

2) Configure the environment. Give the application name, check domain availability and choose PHP as platform.Then click next.



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

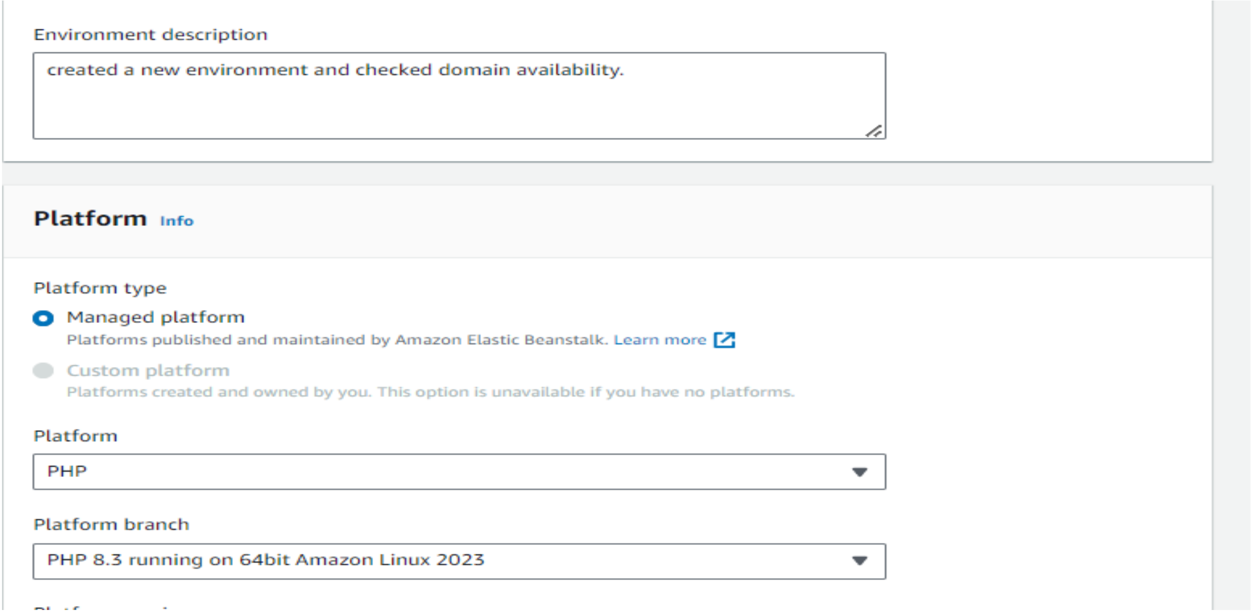
Application information Info

Application name
sampel
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 description
created a new environment and checked domain availability.

Platform Info

Platform type

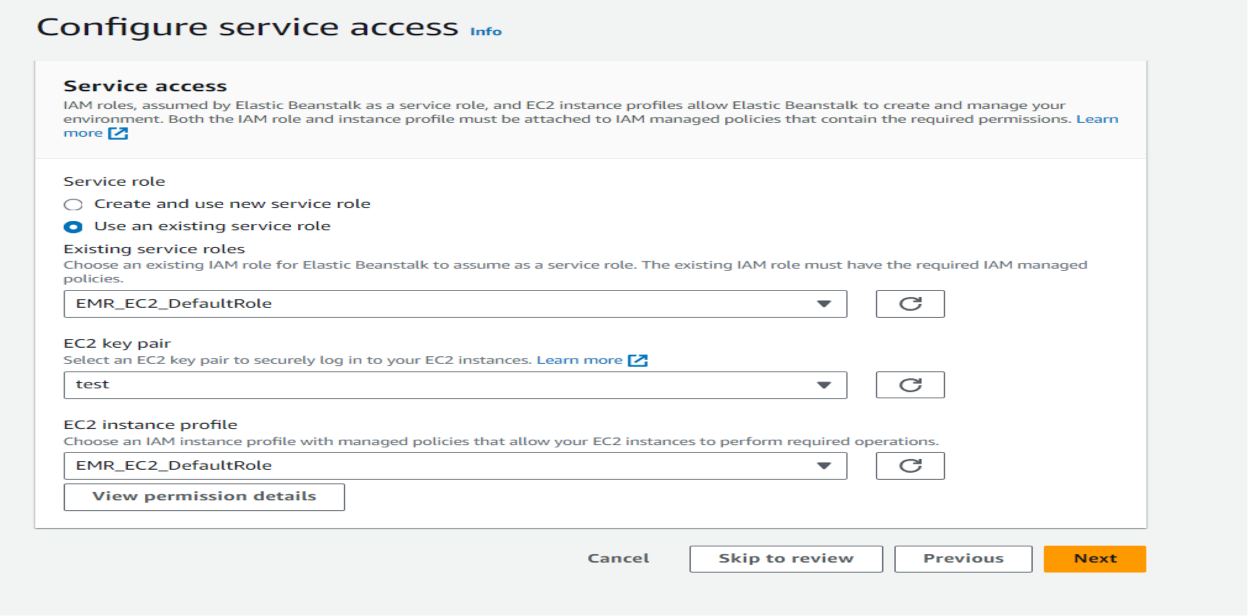
● Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

○ 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

3) Configure the service access.



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.

EMR_EC2_DefaultRole

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

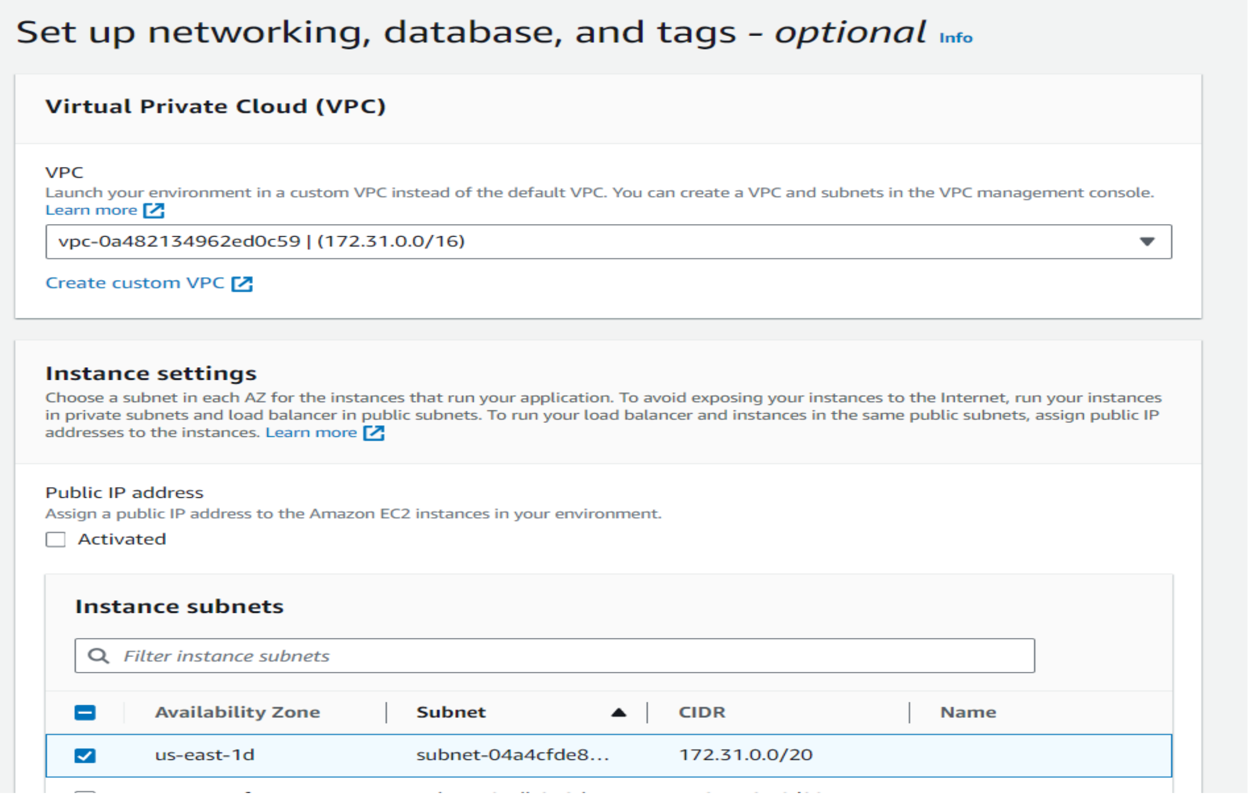
test

EC2 instance profile

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

EMR_EC2_DefaultRole

4) Choose one of the available VPC and instance subnet. Click next.



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

vpc-0a482134962ed0c59 | (172.31.0.0/16)

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

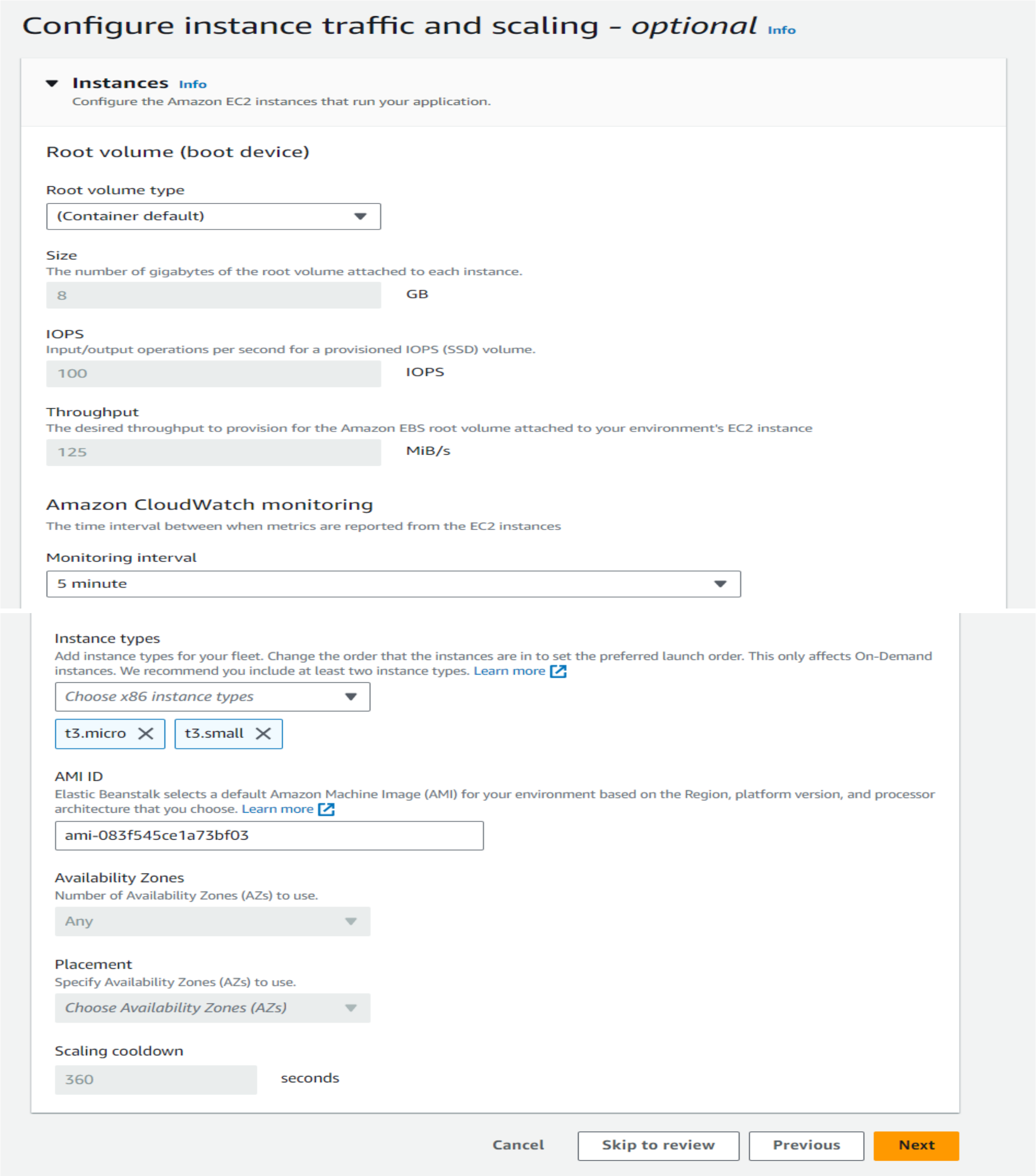
Assign a public IP address to the Amazon EC2 instances in your environment.

□ Activated

Instance subnets

▬	Availability Zone	Subnet ▲	CIDR	Name
☑	us-east-1d	subnet-04a4cfde8...	172.31.0.0/20	

5) Configure instance traffic and scaling. Keep all the options as default.



Configure instance traffic and scaling - *optional* Info

▼ **Instances** Info

Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

(Container default)

Size

The number of gigabytes of the root volume attached to each instance.

8 GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

100 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

125 MiB/s

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances

Monitoring interval

5 minute

Instance types

Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. Learn more

Choose x86 instance types

t3.micro ✕ t3.small ✕

AMI ID

Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. Learn more

ami-083f545ce1a73bf03

Availability Zones

Number of Availability Zones (AZs) to use.

Any

Placement

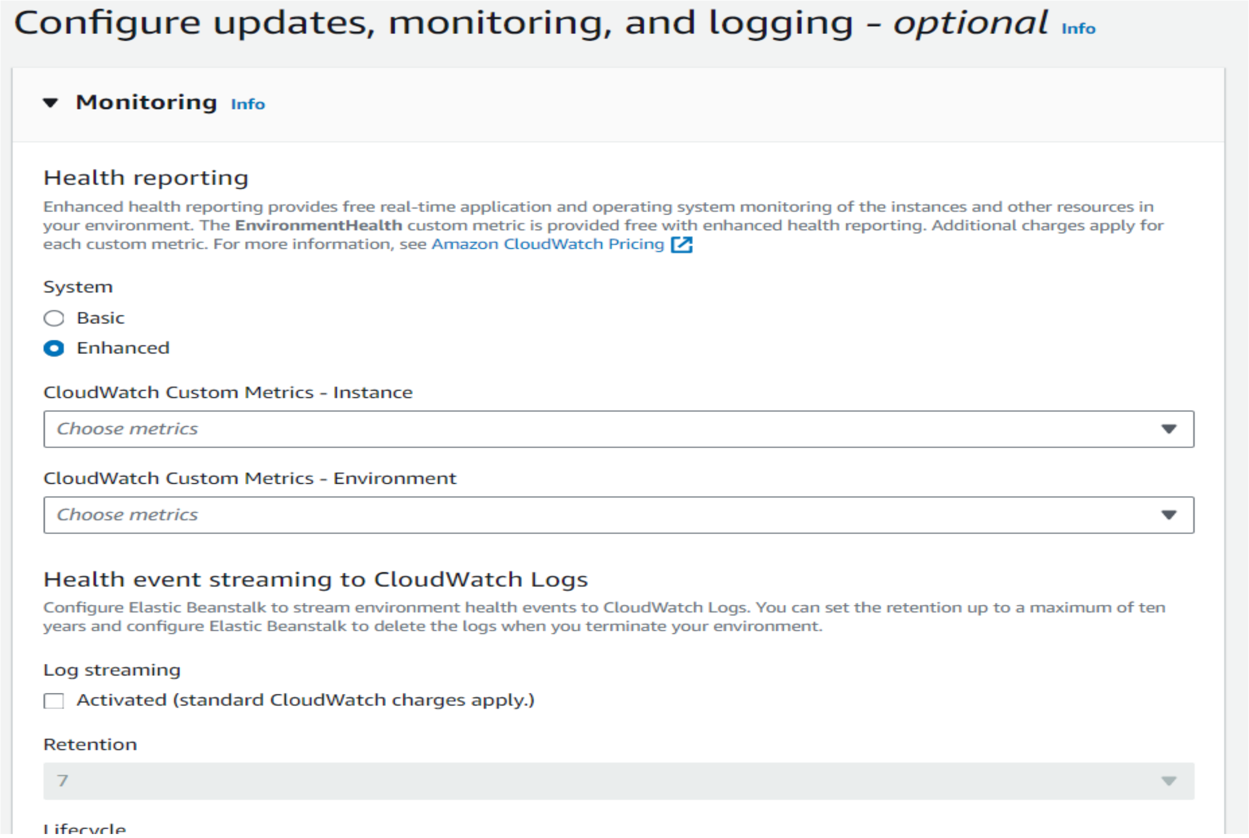
Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs)

Scaling cooldown

360 seconds

6) Configure updates, monitoring, and logging. Keep everything as default and click next.



Configure updates, monitoring, and logging - *optional* Info

▼ **Monitoring** Info

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see Amazon CloudWatch Pricing

System

○ Basic

● Enhanced

CloudWatch Custom Metrics - Instance

Choose metrics ▼

CloudWatch Custom Metrics - Environment

Choose metrics ▼

Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

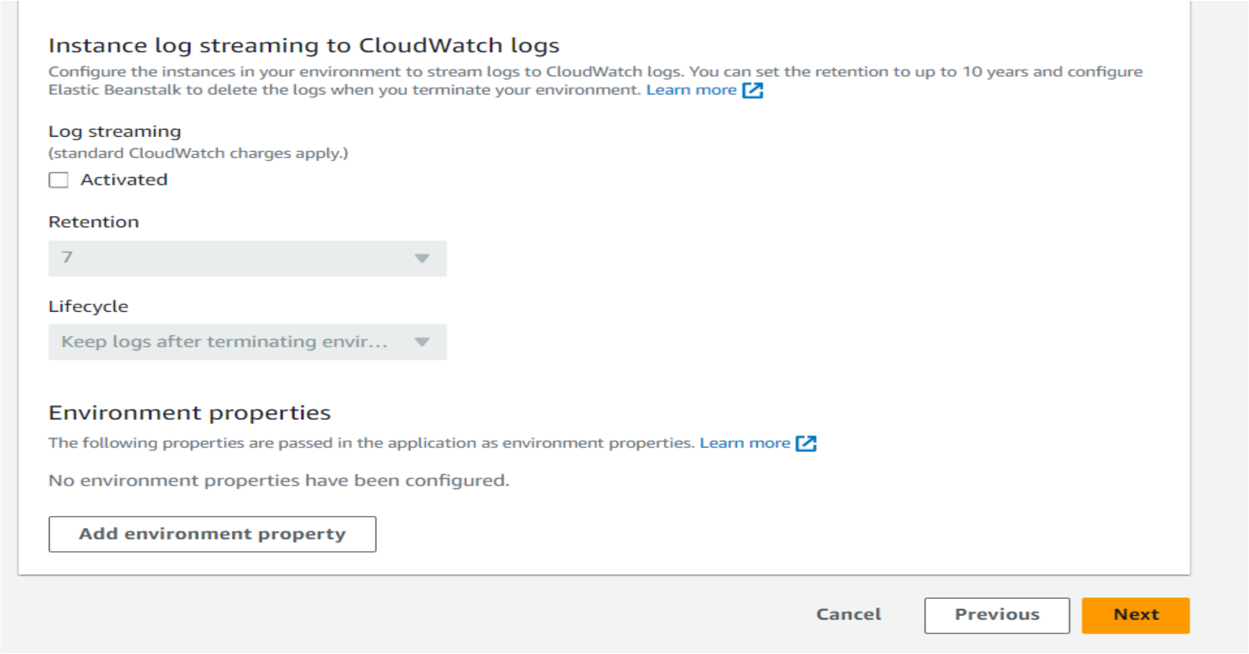
Log streaming

□ Activated (standard CloudWatch charges apply.)

Retention

7 ▼

Lifecycle



Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more

Log streaming
(standard CloudWatch charges apply.)

□ Activated

Retention

7 ▼

Lifecycle

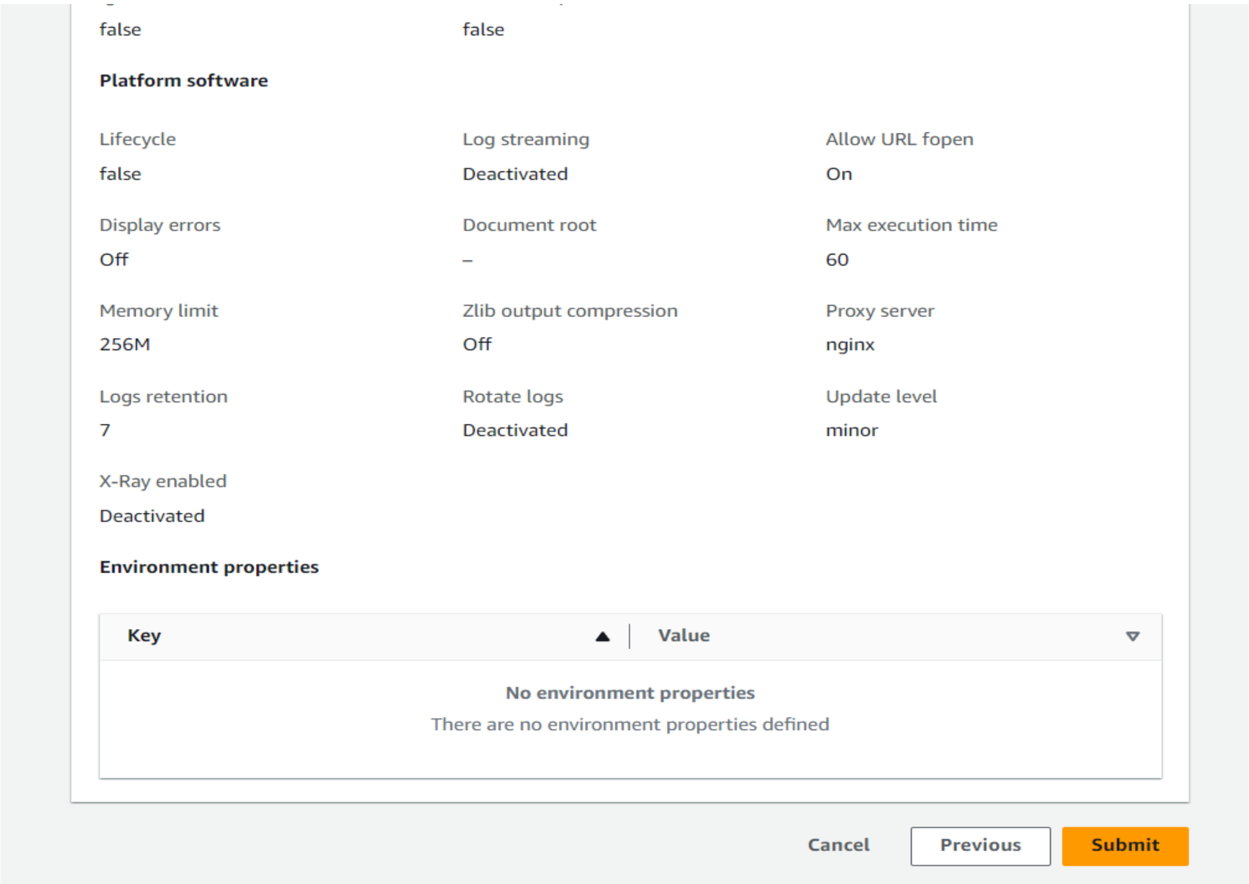
Keep logs after terminating envir... ▼

Environment properties

The following properties are passed in the application as environment properties. Learn more

No environment properties have been configured.

7) Click submit.



false	false	

Platform software

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

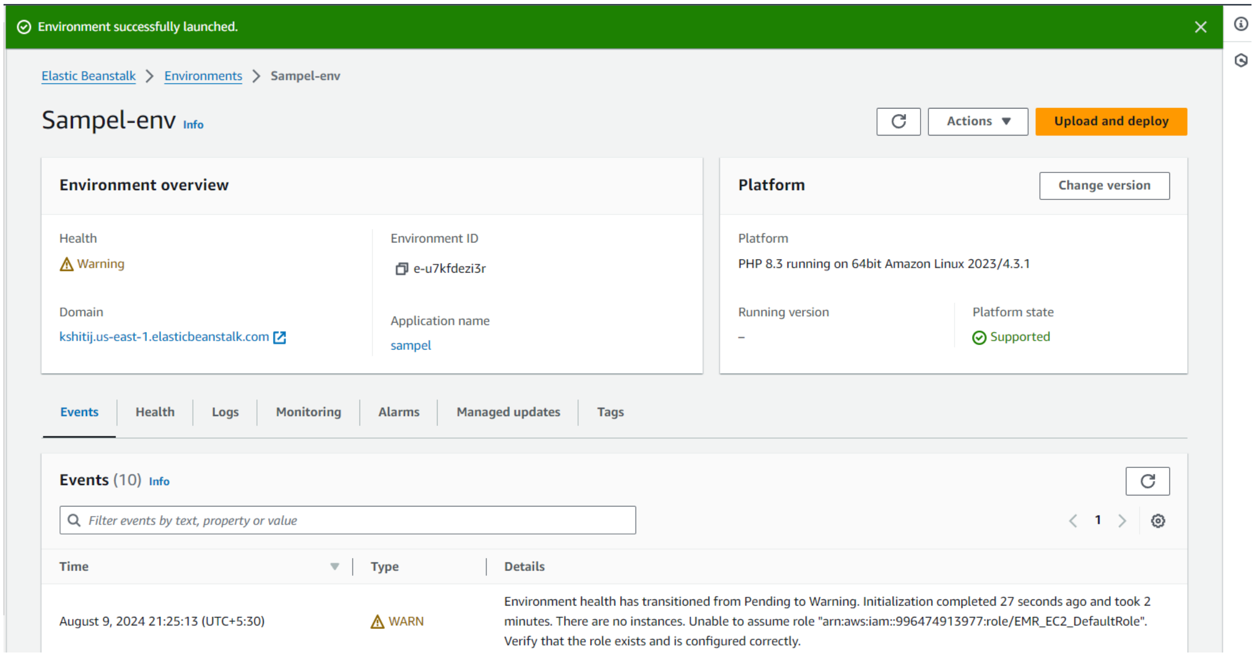
Environment properties

Key	Value

No environment properties

There are no environment properties defined

8) Environment has been created successfully.



Environment successfully launched.

Sampel-env

Environment overview

Health ⚠ Warning	Environment ID e-u7kfdezi3r
Domain kshitij.us-east-1.elasticbeanstalk.com	Application name sampel

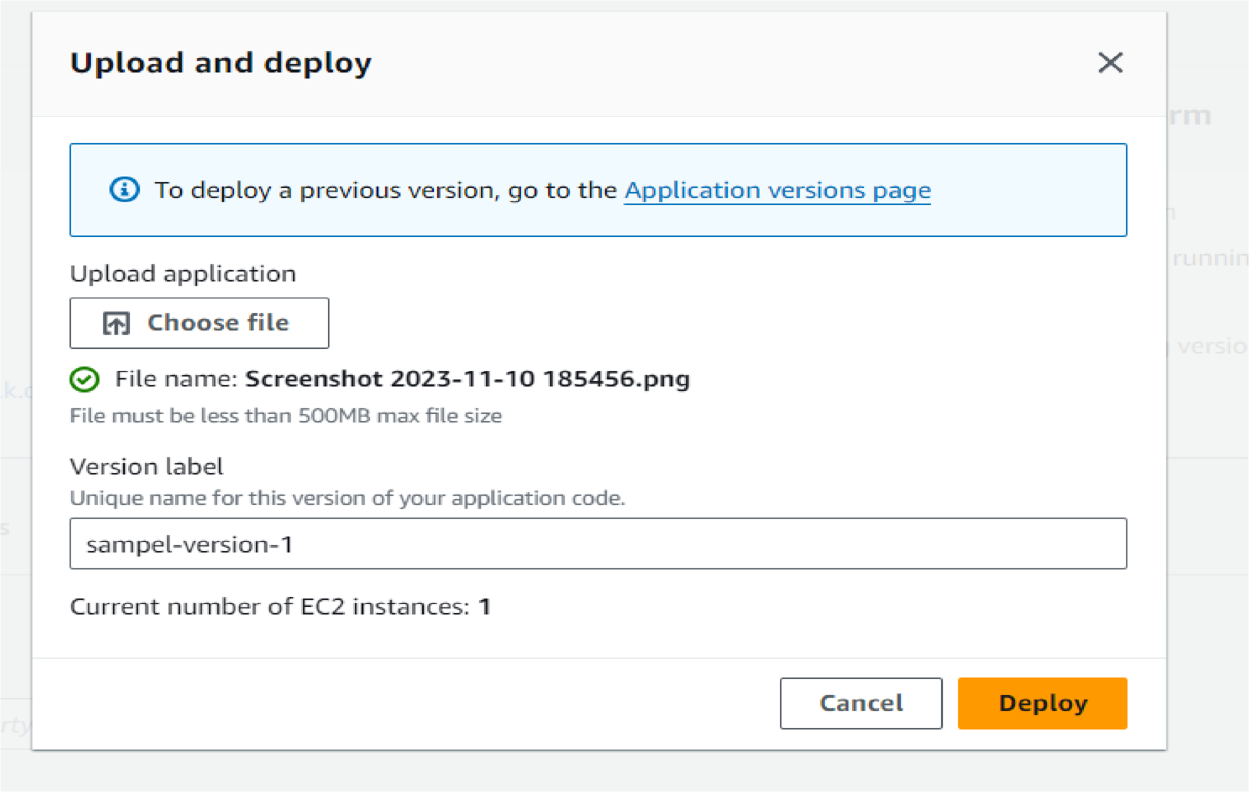
Platform

Platform PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	
Running version –	Platform state ✓ Supported

Events (10) Info

Time	Type	Details
August 9, 2024 21:25:13 (UTC+5:30)	⚠ WARN	Environment health has transitioned from Pending to Warning. Initialization completed 27 seconds ago and took 2 minutes. There are no instances. Unable to assume role "arn:aws:iam::996474913977:role/EMR_EC2_DefaultRole". Verify that the role exists and is configured correctly.

9) Deploy something on the recently created environment.



Upload and deploy

ⓘ To deploy a previous version, go to the Application versions page

Upload application

✓ File name: **Screenshot 2023-11-10 185456.png**

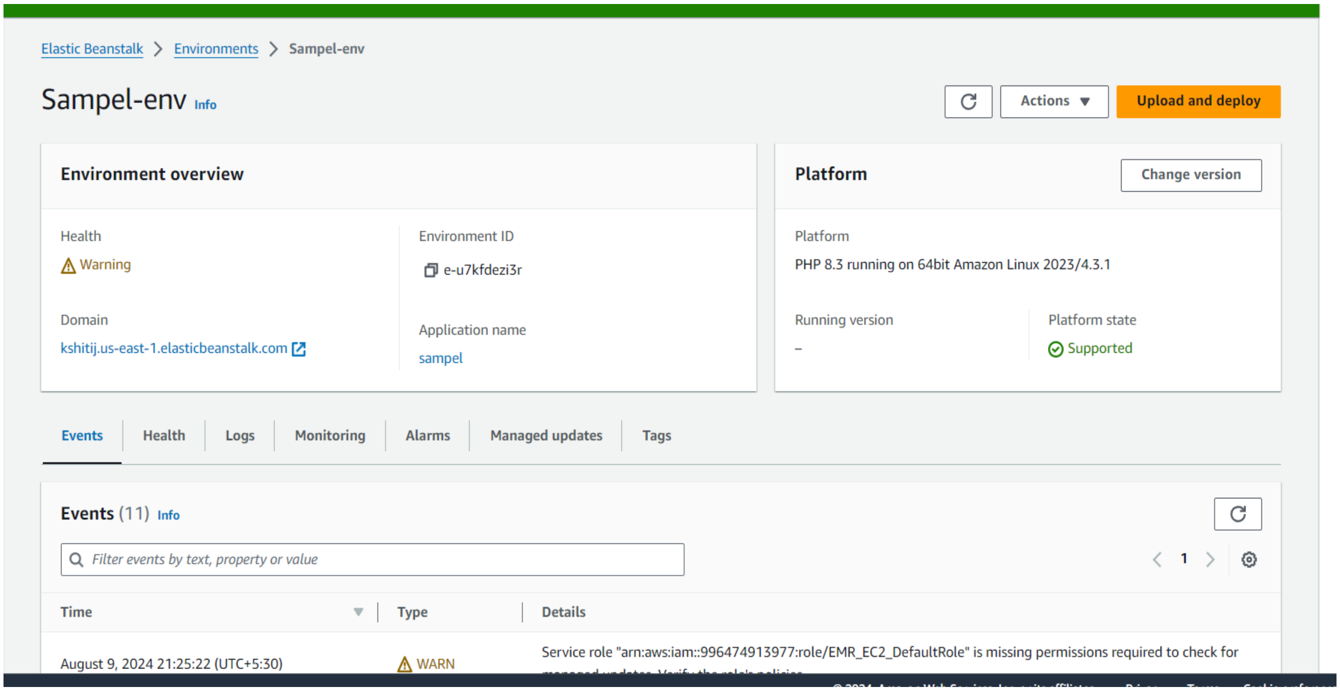
File must be less than 500MB max file size

Version label

Unique name for this version of your application code.

sampel-version-1

Current number of EC2 instances: **1**



Sampel-env Info

Environment overview

Health: ⚠ Warning

Environment ID: e-u7kfdezi3r

Domain: kshitij.us-east-1.elasticbeanstalk.com

Application name: sampel

Platform

Platform: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1

Running version: –

Platform state: ✓ Supported

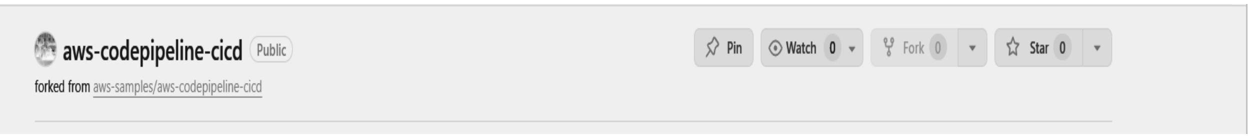
Events (11) Info

Filter events by text, property or value

Time	Type	Details
August 9, 2024 21:25:22 (UTC+5:30)	⚠ WARN	Service role "arn:aws:iam::996474913977:role/EMR_EC2_DefaultRole" is missing permissions required to check for managed updates. Verify the role's policies.

Pipeline Creation:

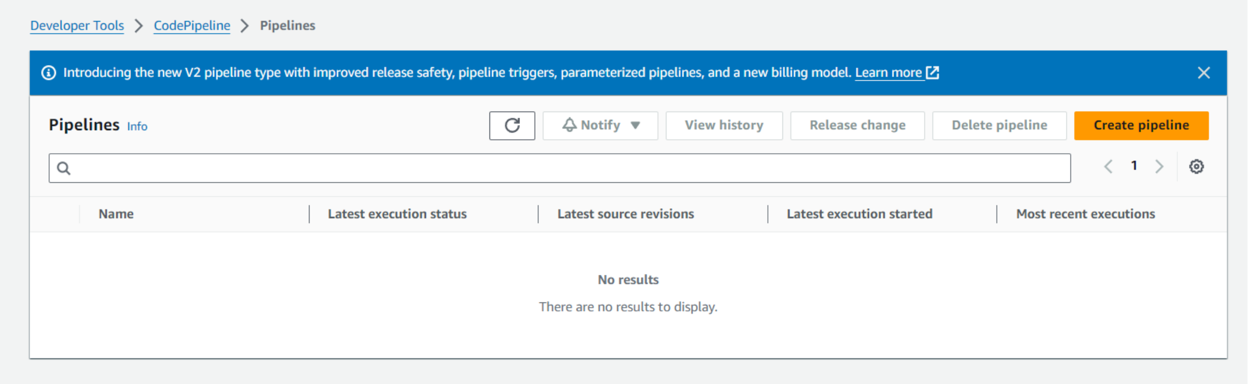
1) Fork a git-hub repository. This forked repository will act as source for your code pipeline.



aws-codepipeline-cicd Public

forked from aws-samples/aws-codepipeline-cicd

2) Go to developer tools and select CodePipeline and create a new pipeline



Developer Tools > CodePipeline > Pipelines

Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more

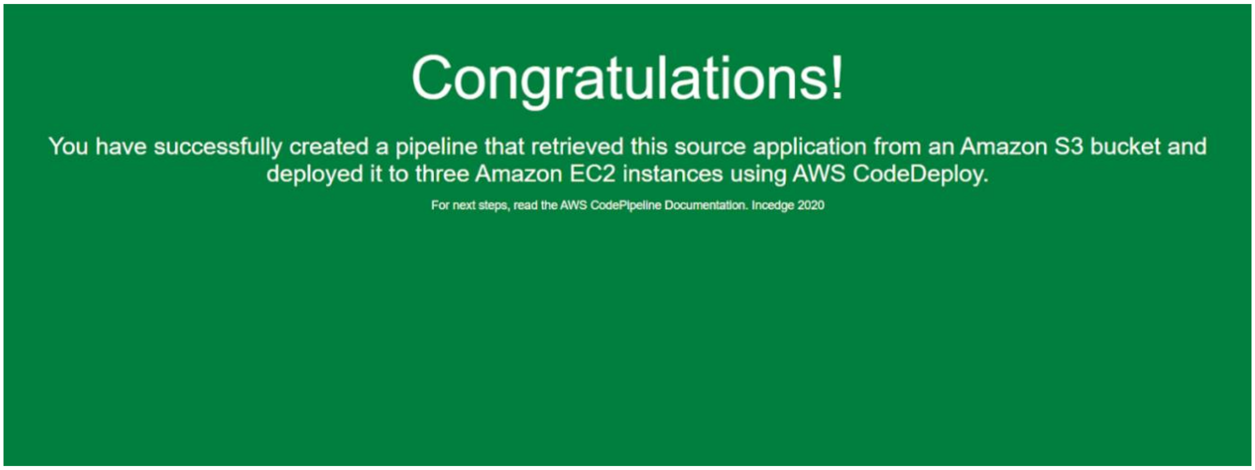
Pipelines Info

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions

No results

There are no results to display.

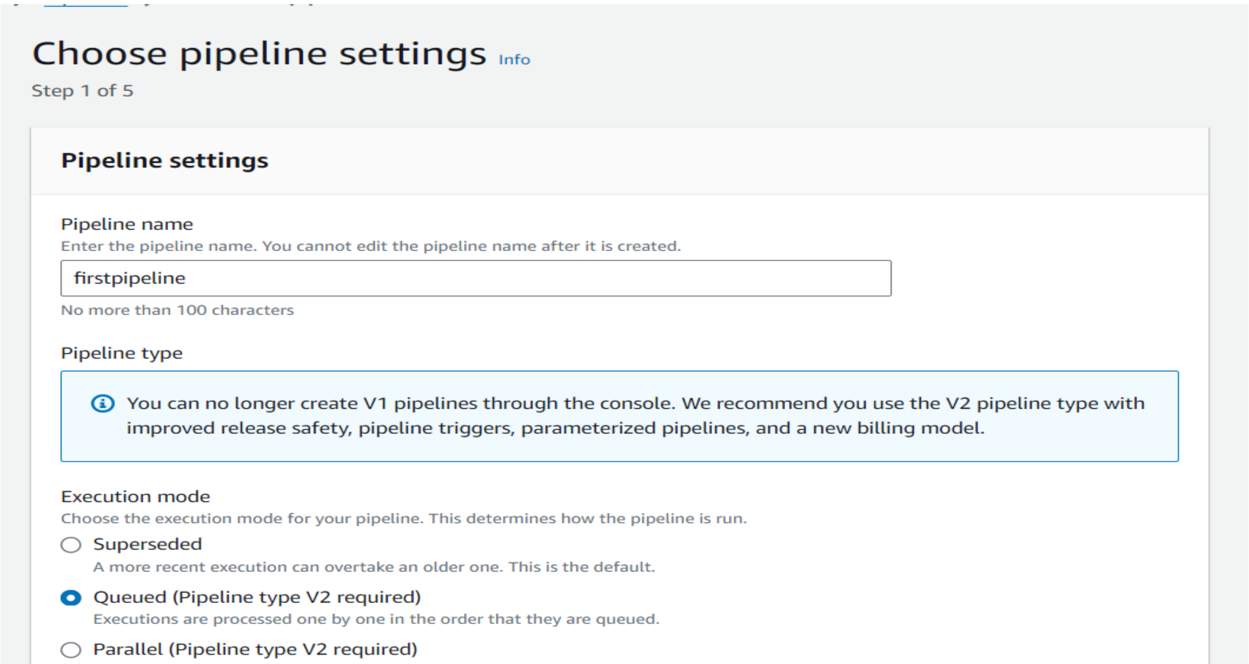
3) Create a pipeline:



Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020



Choose pipeline settings

Step 1 of 5

Pipeline settings

Pipeline name

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

firstpipeline

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

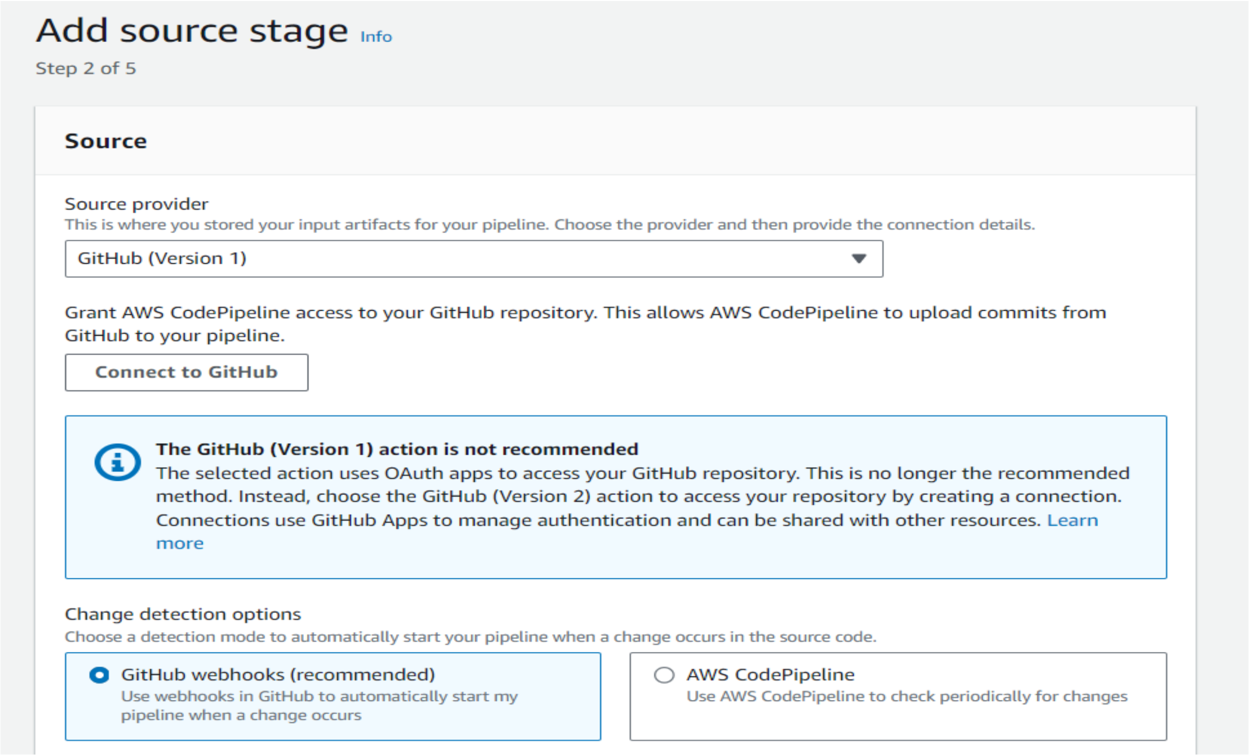
A more recent execution can overtake an older one. This is the default.

● Queued (Pipeline type V2 required)

Executions are processed one by one in the order that they are queued.

○ Parallel (Pipeline type V2 required)

4)



Add source stage

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 1)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more

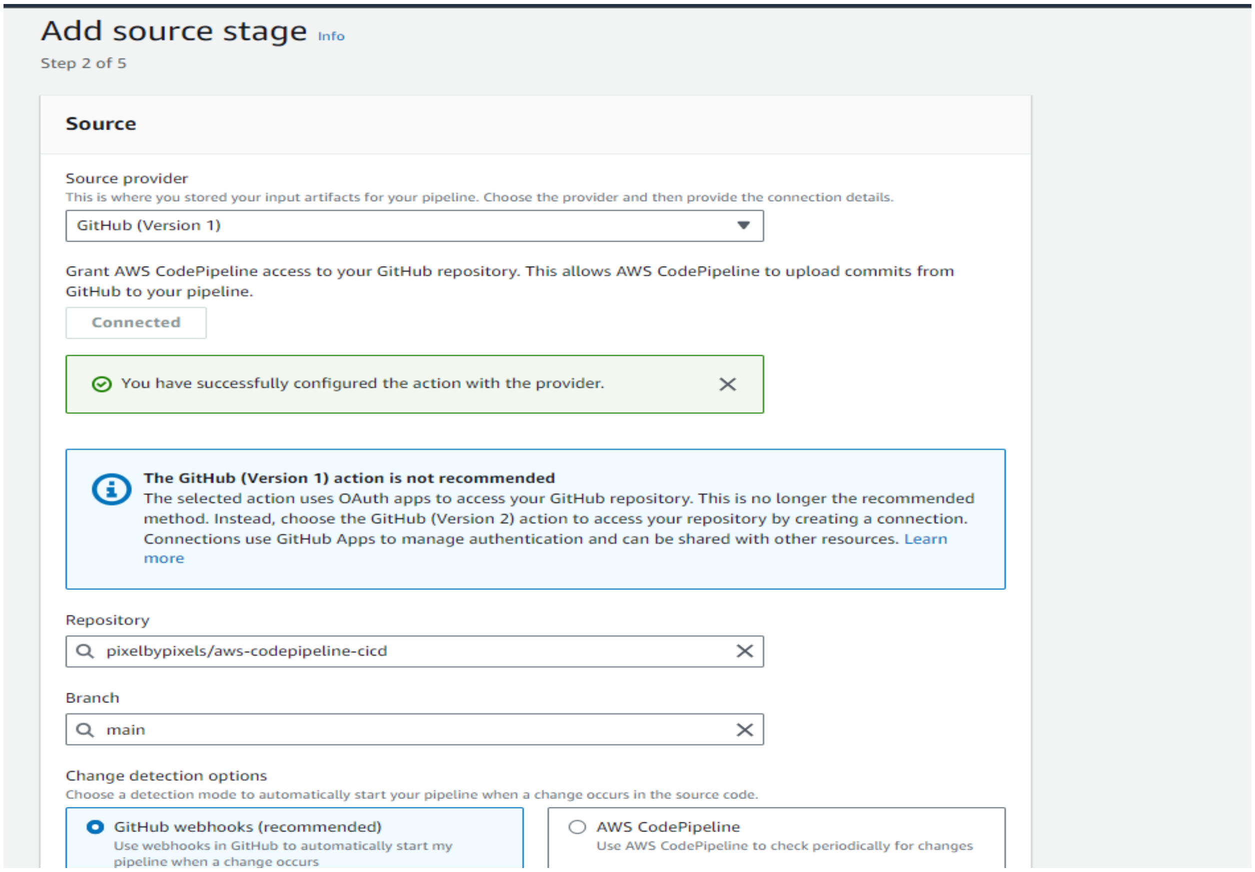
Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

● GitHub webhooks (recommended)
Use webhooks in GitHub to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

s



Add source stage

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 1)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

You have successfully configured the action with the provider.

The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more

Repository

pixelbypixels/aws-codepipeline-cicd

Branch

main

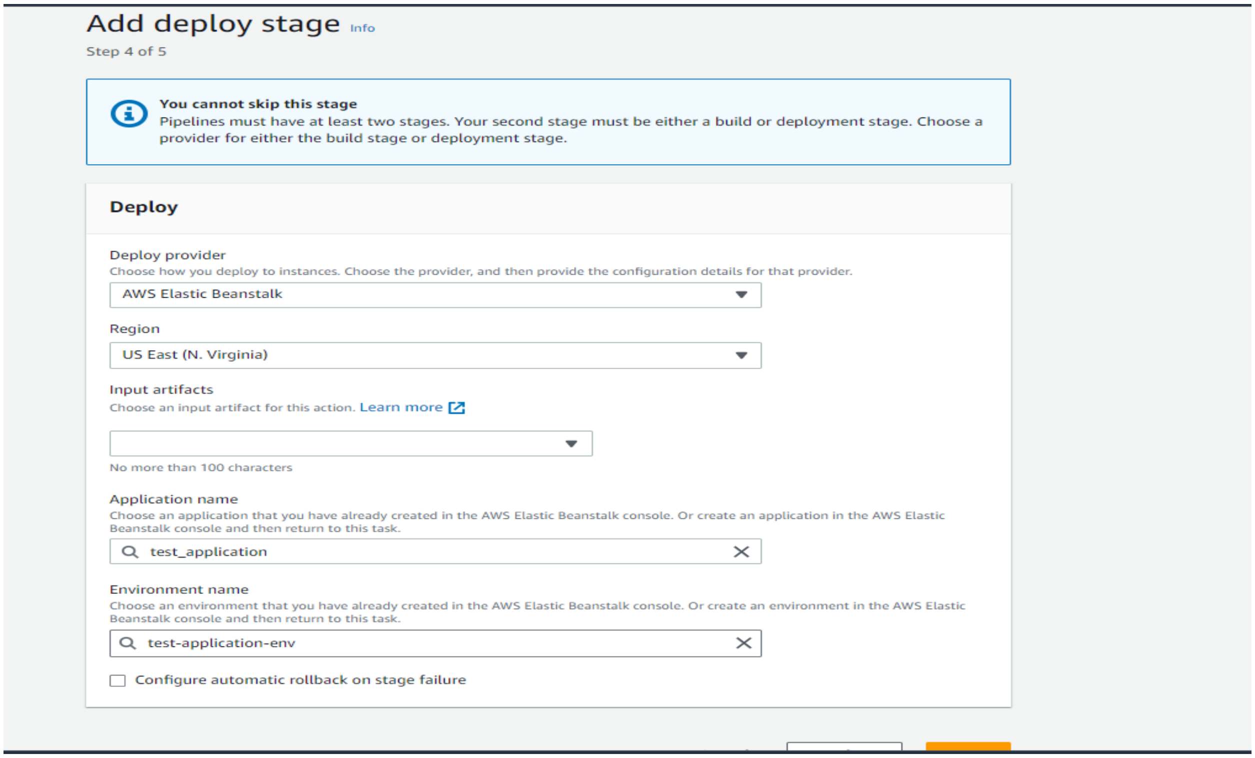
Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

● GitHub webhooks (recommended)
Use webhooks in GitHub to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

5) Go to the deploy stage and ensure the following settings.



Add deploy stage

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

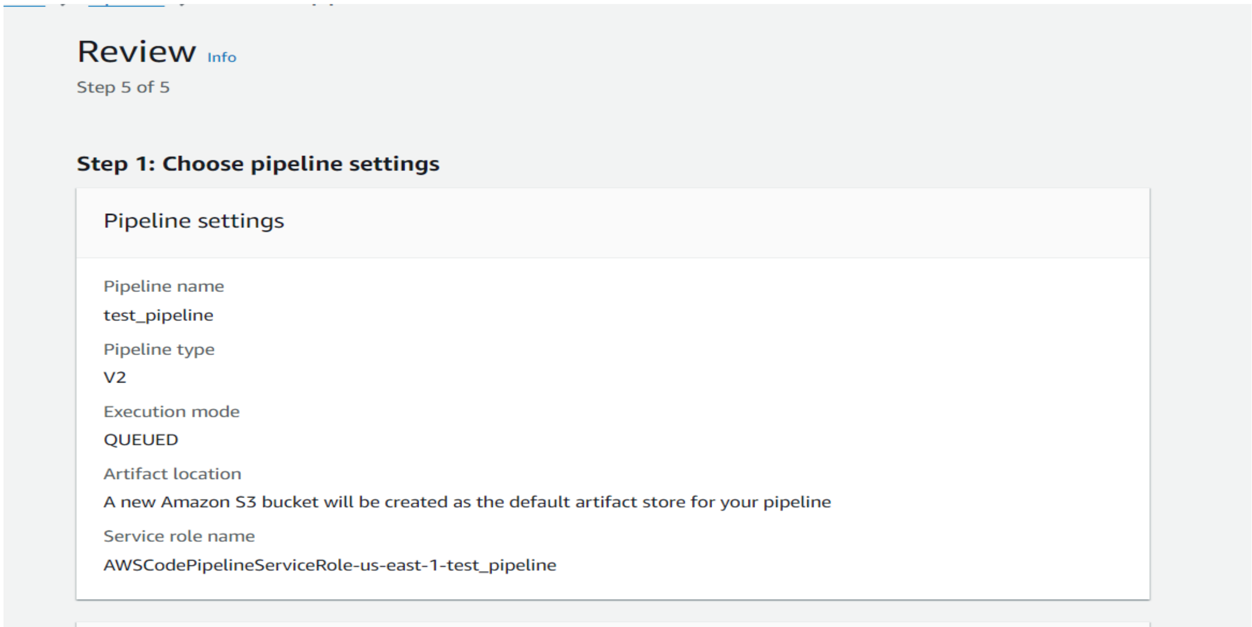
No more than 100 characters

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.
test_application

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.
test-application-env

□ Configure automatic rollback on stage failure

6) review the pipeline settings.



Review

Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
test_pipeline

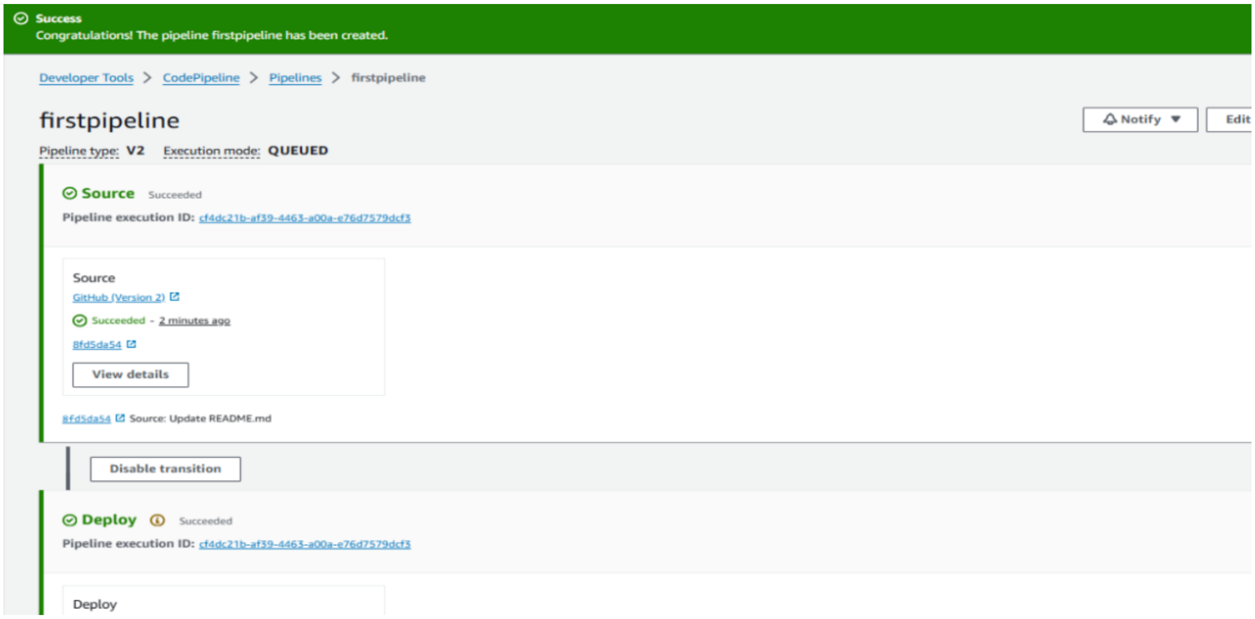
Pipeline type
V2

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-test_pipeline

7) Then go ahead and check the URL provided in the EBS environment.



Success
Congratulations! The pipeline firstpipeline has been created.

Developer Tools > CodePipeline > Pipelines > firstpipeline

firstpipeline

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: cf4dc21b-af39-4463-a00a-e76d7579dcf3

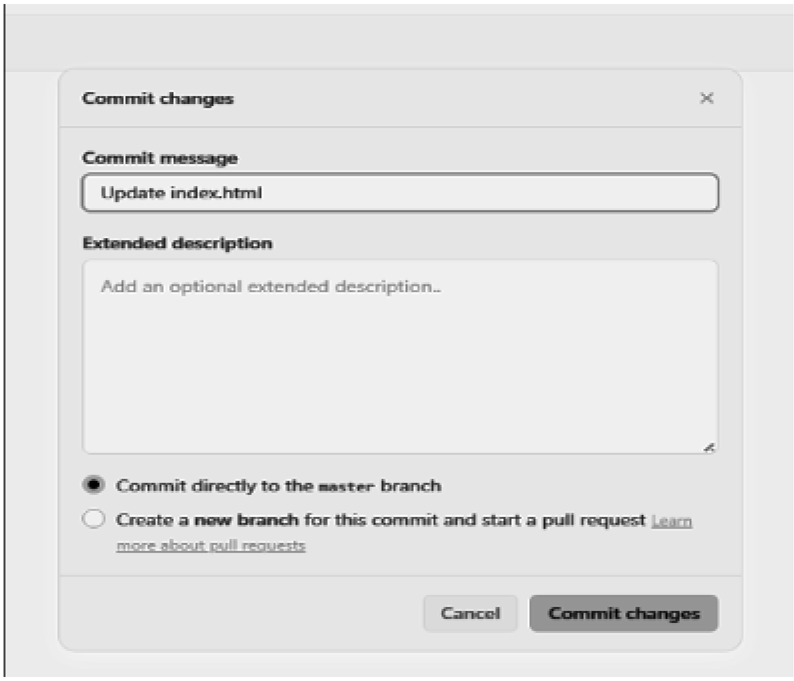
Source
GitHub (Version 2)
Succeeded - 2 minutes ago
8fd5da54

8fd5da54 Source: Update README.md

Deploy Succeeded
Pipeline execution ID: cf4dc21b-af39-4463-a00a-e76d7579dcf3

Deploy

8) Go to the repository and make the changes in the index.html file and commit them



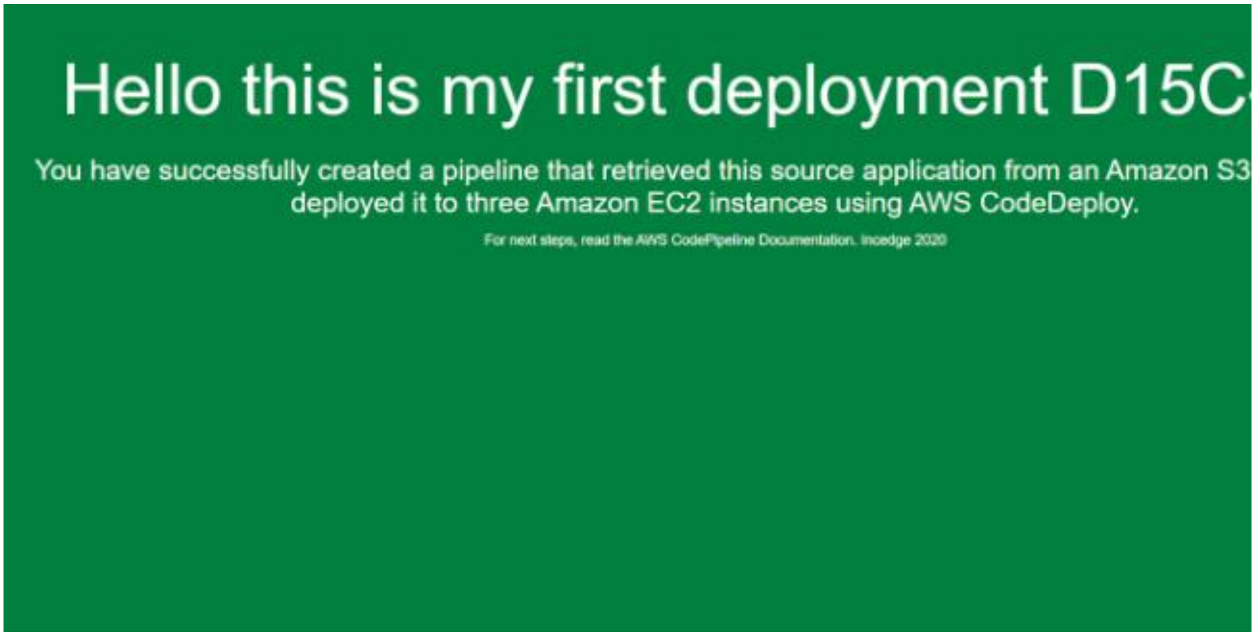
Commit changes

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

9) To view the changes made, ensure they are committed and visible in the source panel in real time. After confirming that the deployment section indicates success, refresh the URL to see the updates reflected on your site or application.



Hello this is my first deployment D15C

You have successfully created a pipeline that retrieved this source application from an Amazon S3
deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020