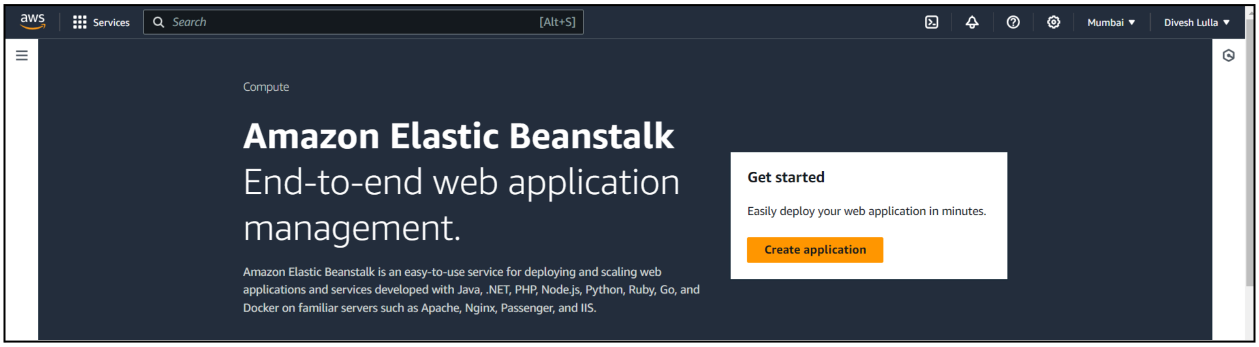
Name: Divesh Lulla Class: D15_C Roll No: 31

Experiment No 02

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.

Step 1: Go to AWS, search and open Elastic Beanstalk.



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

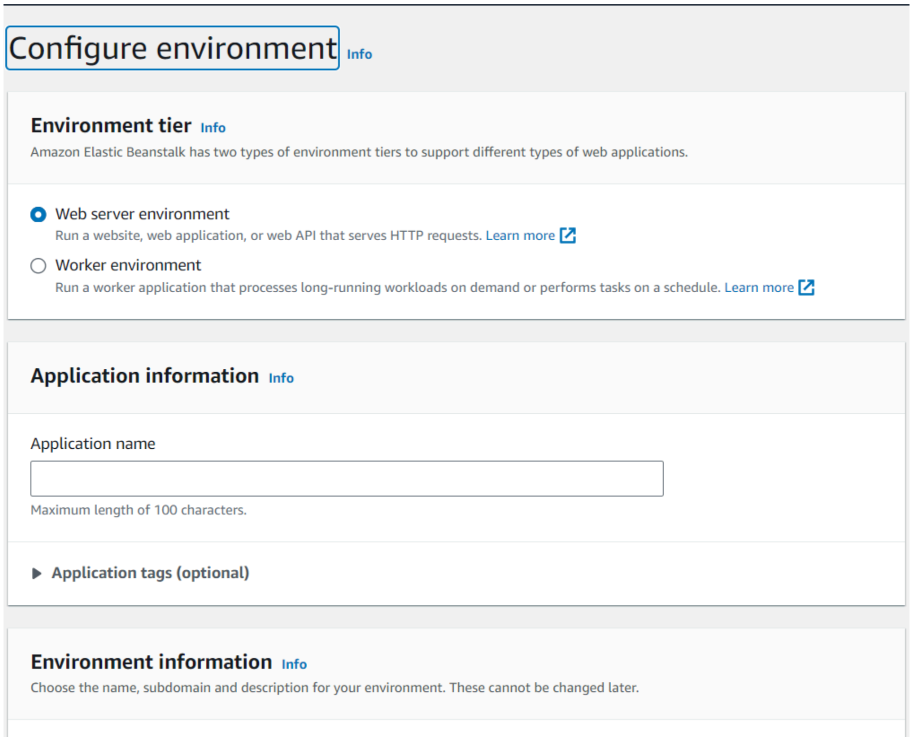
Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create application

Step 2: Click on Create application and configure the Environment.



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

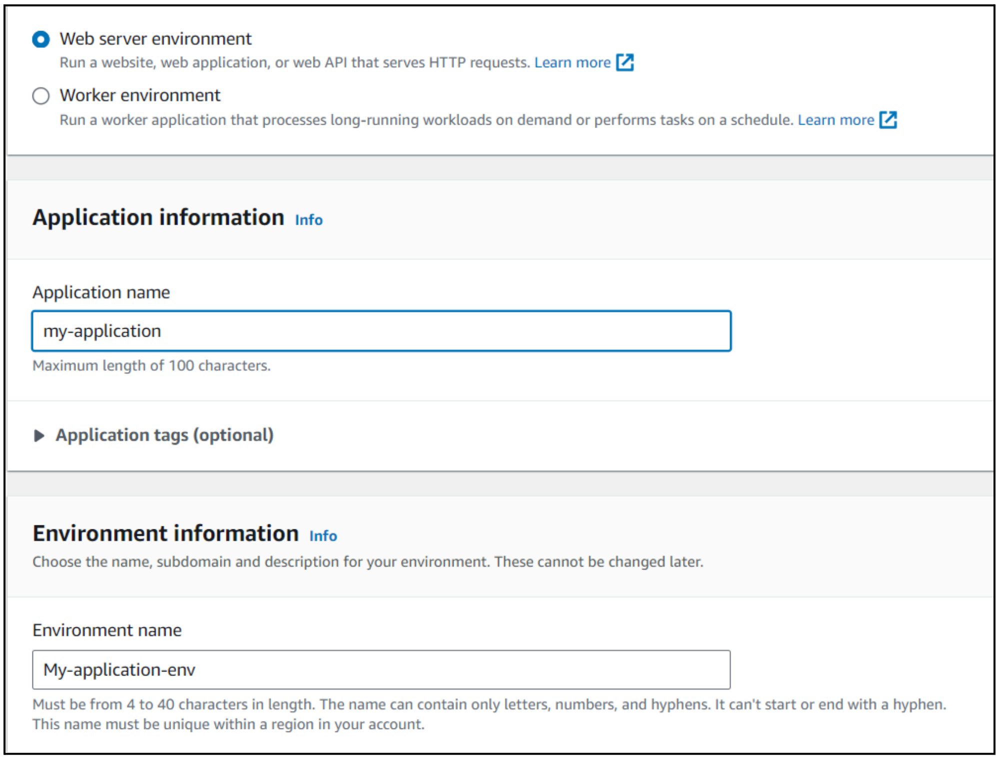
Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

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

Step 3: Fill out the details according to need.



● 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

my-application

Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

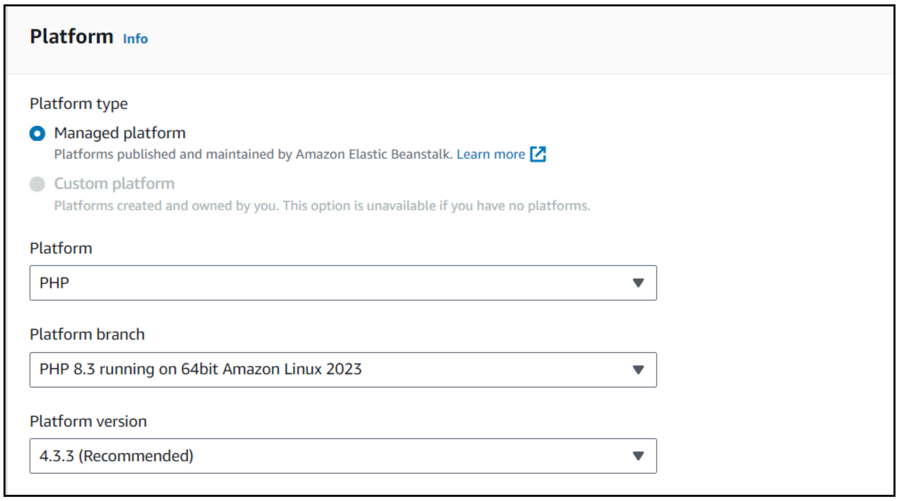
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

My-application-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.

Step 4: Scroll down to platform and fill the details.



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

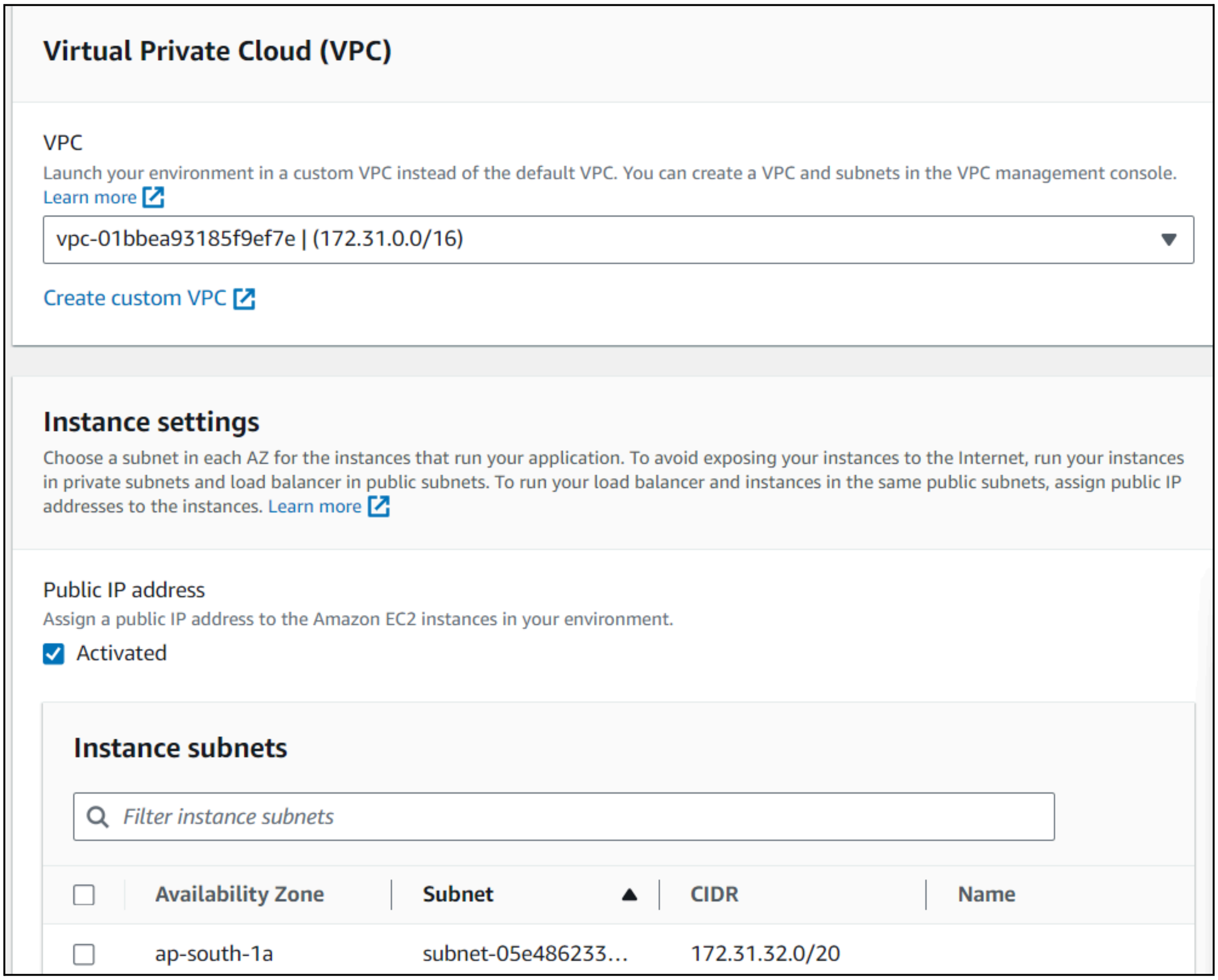
PHP 8.3 running on 64bit Amazon Linux 2023 ▼

Platform version

4.3.3 (Recommended) ▼

Step 5: Click on next, Keep all other settings as default.

Step 6: In the Setup Network section select a vpc and instance subnets then click on next



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-01bbea93185f9ef7e | (172.31.0.0/16)

Create custom VPC

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.

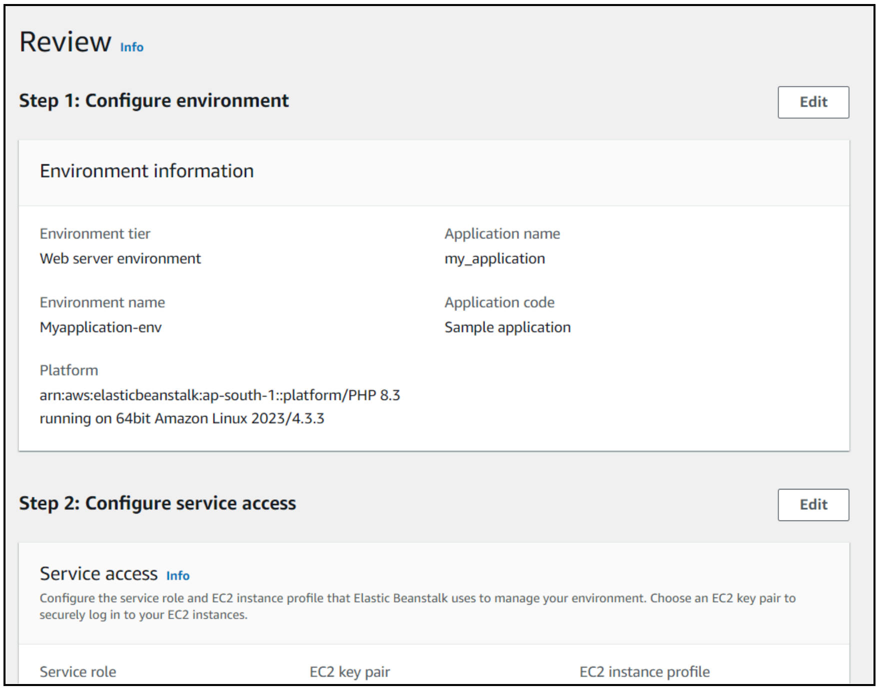
☑ Activated

Instance subnets

Filter instance subnets

□	Availability Zone	Subnet ▲	CIDR	Name
□	ap-south-1a	subnet-05e486233...	172.31.32.0/20	

Step 7: Next is the review page where you can check the configuration that have been set in the previous steps. Click on submit.



Review

Step 1: Configure environment

Environment information

Environment tier
Web server environment

Application name
my_application

Environment name
Myapplication-env

Application code
Sample application

Platform
arn:aws:elasticbeanstalk:ap-south-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.3

Step 2: Configure service access

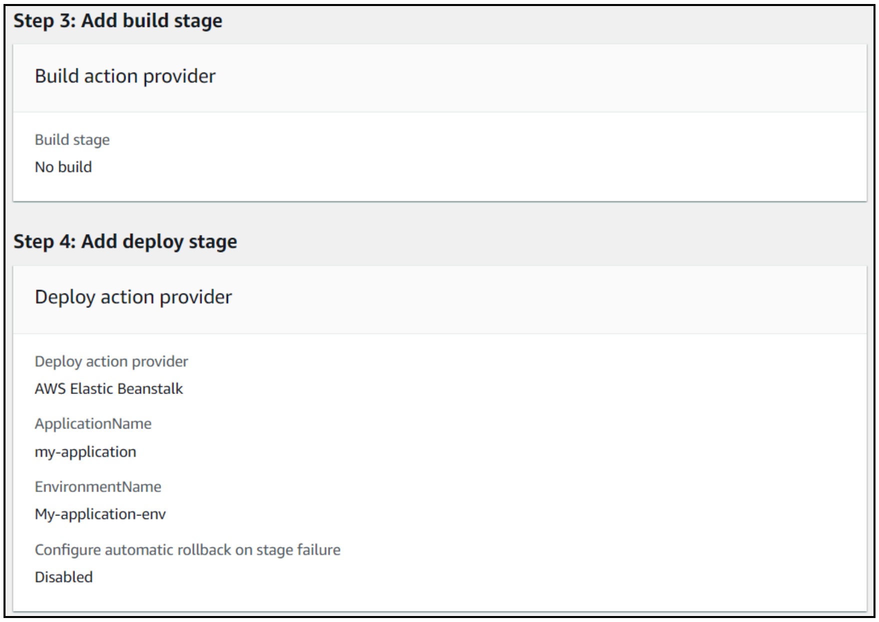
Service access

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



Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

Deploy action provider

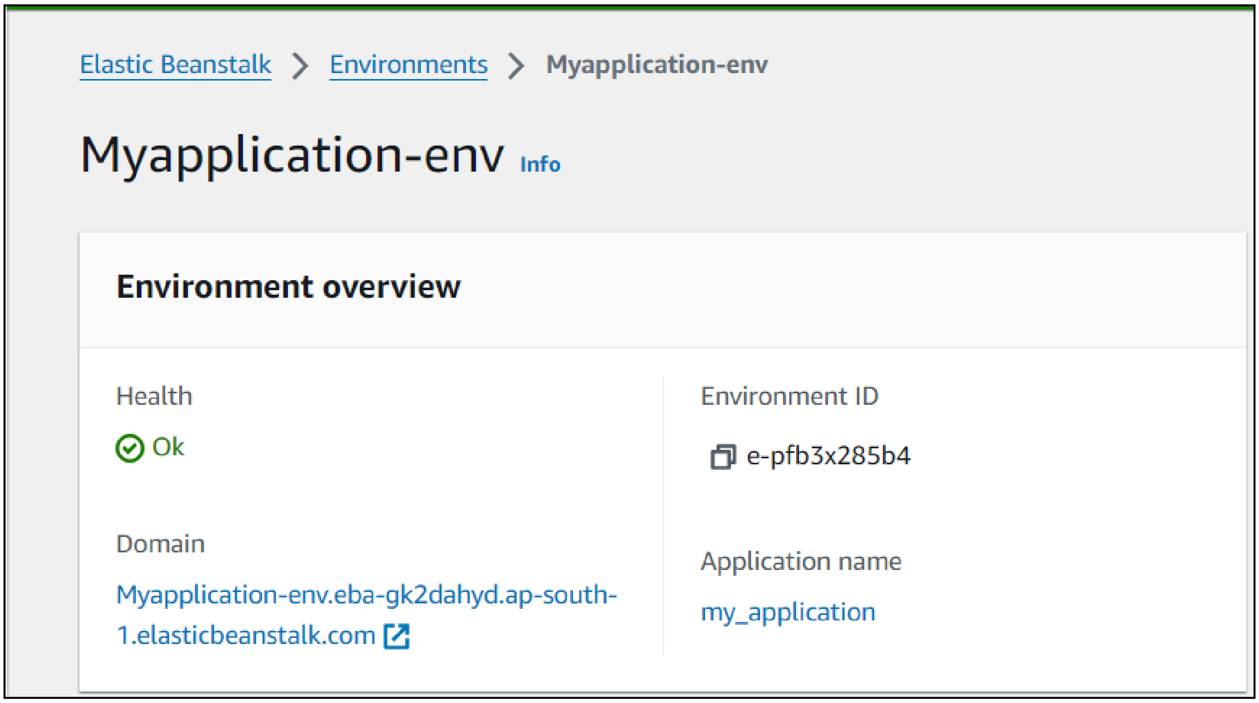
Deploy action provider
AWS Elastic Beanstalk

ApplicationName
my-application

EnvironmentName
My-application-env

Configure automatic rollback on stage failure
Disabled

Step 10: Environment will be successfully created.



Elastic Beanstalk > Environments > Myapplication-env

Myapplication-env Info

Environment overview

Health

Ok

Domain

Myapplication-env.eba-gk2dahyd.ap-south-1.elasticbeanstalk.com

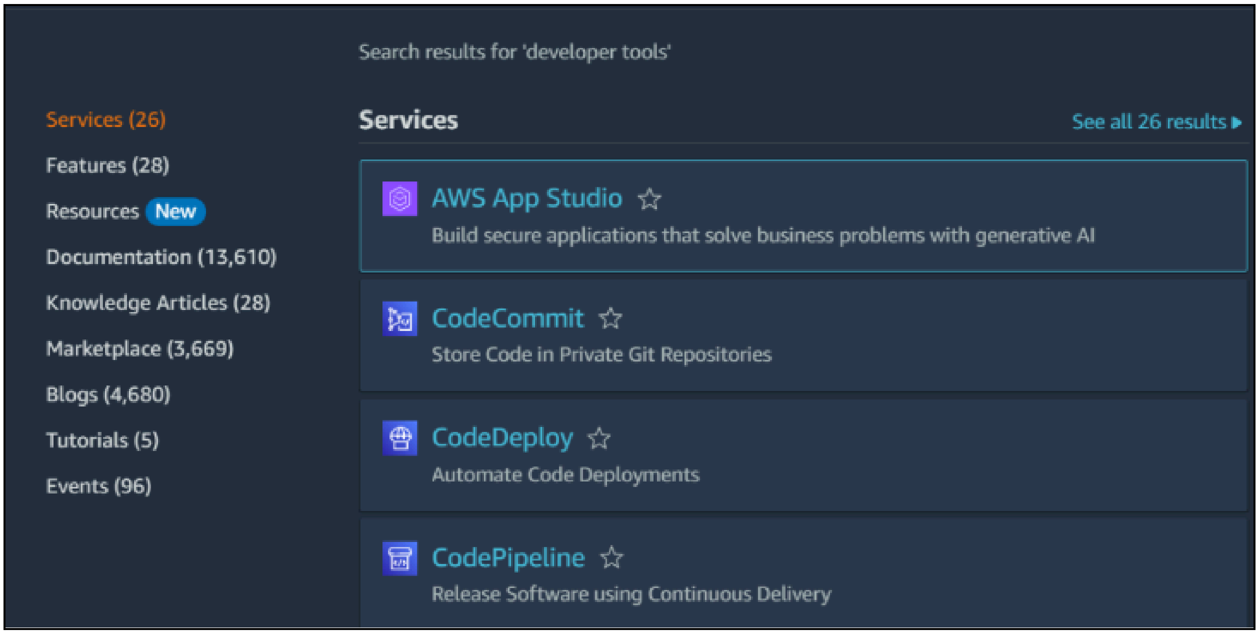
Environment ID

e-pfb3x285b4

Application name

my_application

Step 11: Now after this we are going to create a pipeline, for creating pipeline go to Services and Select CodePipeline.



Search results for 'developer tools'

Services

See all 26 results

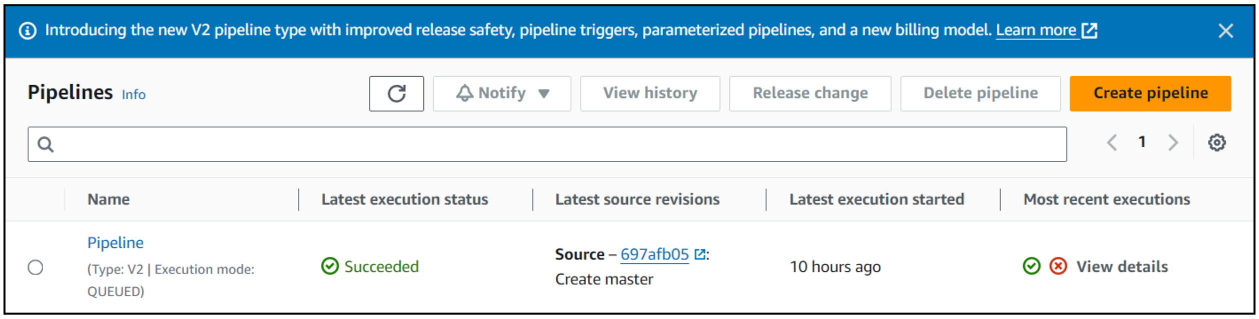
AWS App Studio ☆
Build secure applications that solve business problems with generative AI

CodeCommit ☆
Store Code in Private Git Repositories

CodeDeploy ☆
Automate Code Deployments

CodePipeline ☆
Release Software using Continuous Delivery

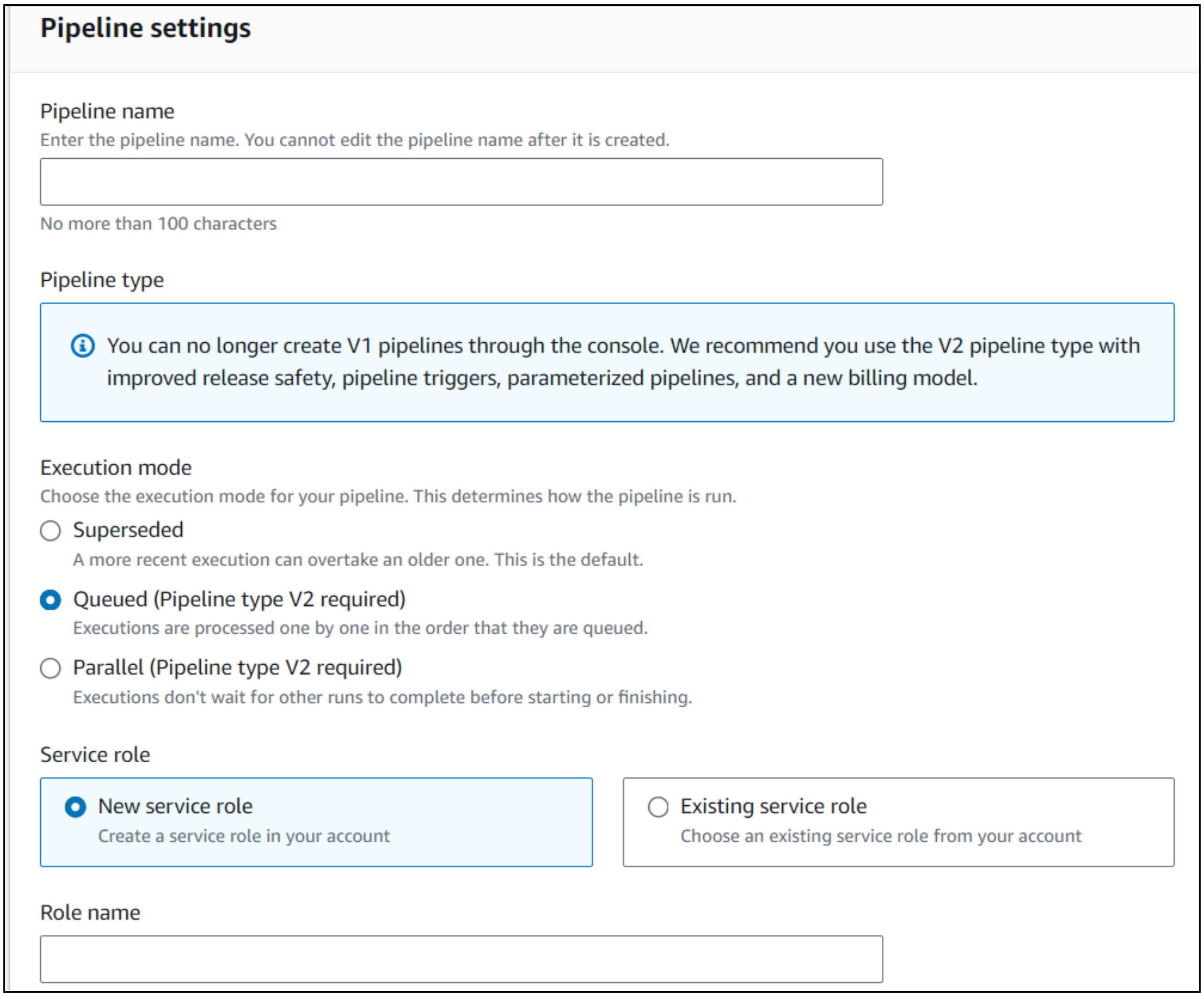
Step 12: Select Create a Pipeline



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

	Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
○	Pipeline (Type: V2 | Execution mode: QUEUED)	✓ Succeeded	**Source** – 697afb05 : Create master	10 hours ago	✓ ⊗ View details

Step 13: Fill the required fields for creating a pipeline



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.

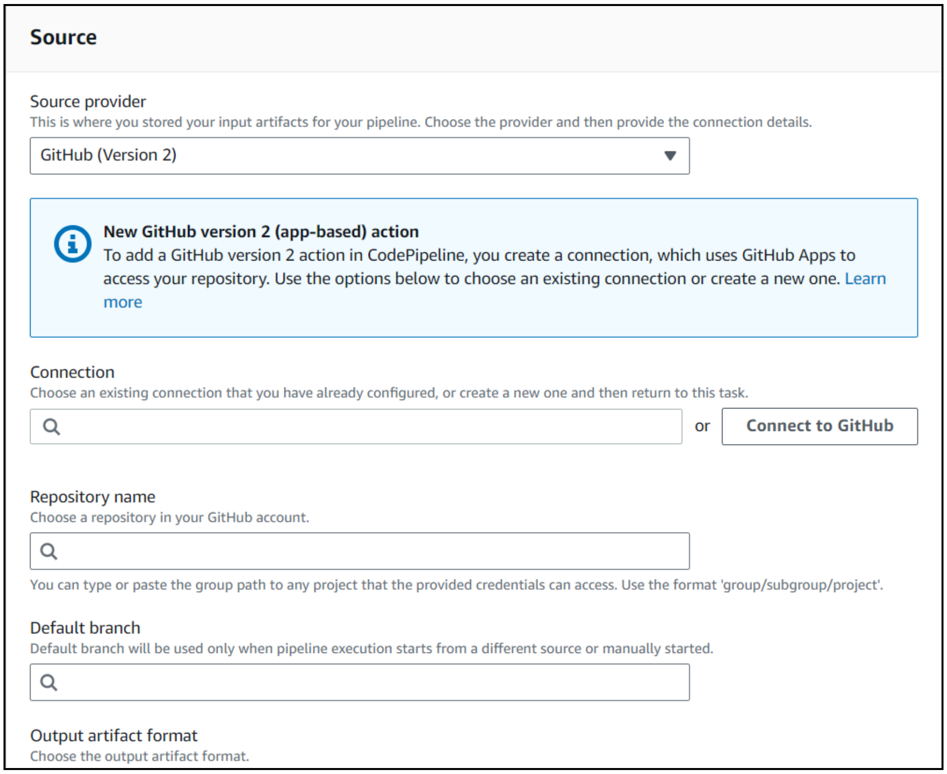
Service role

● New service role
Create a service role in your account

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

Role name

Step 14: In Add Source stage select Github version 2 as the source provider and then connect the aws to github account.



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.

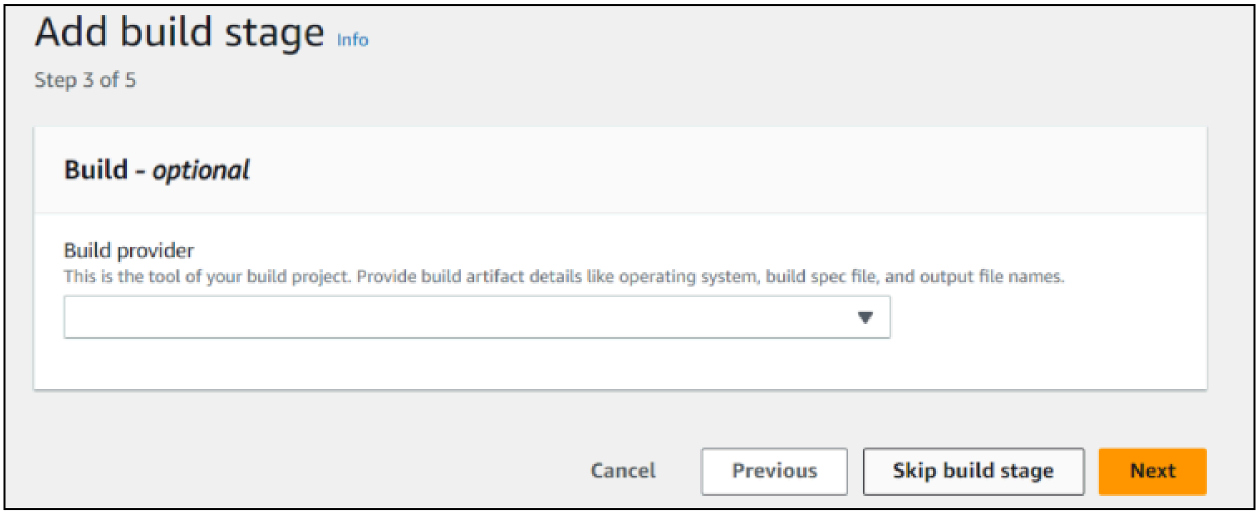
Repository name
Choose a repository in your GitHub account.

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.

Step 15: Skip the build stage and review stage.



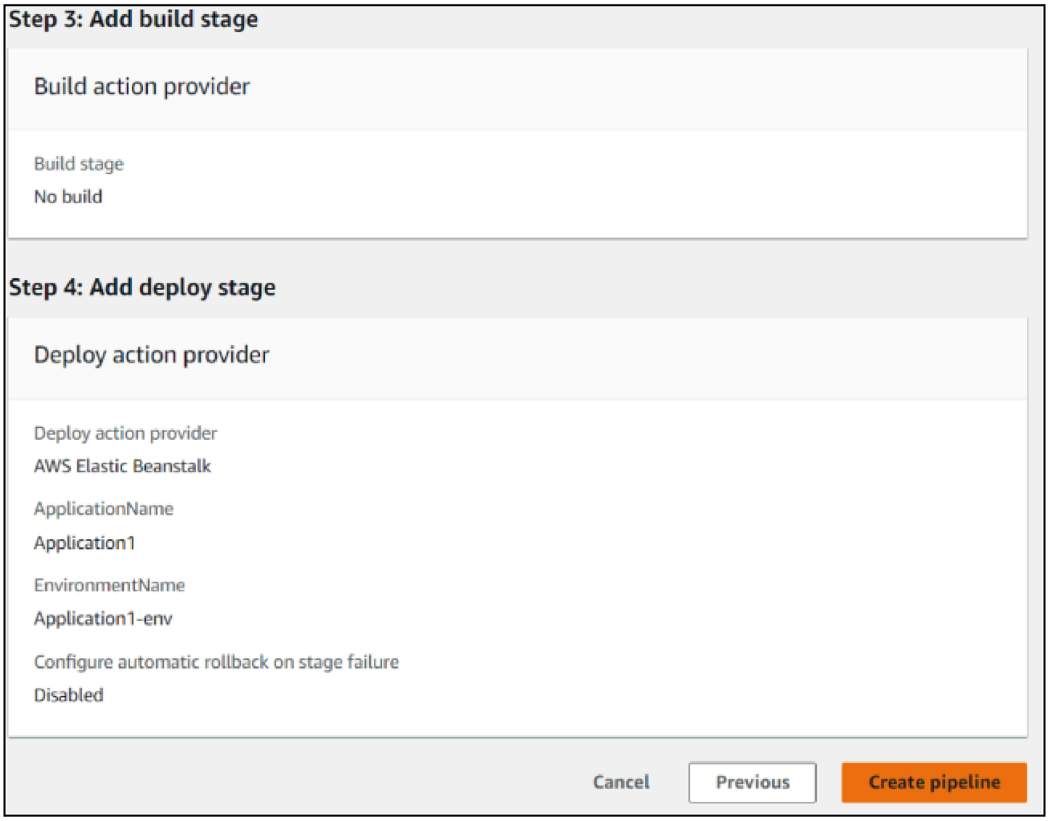
Add build stage Info

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.

Step 16: In review Stage check all the settings that have been done 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

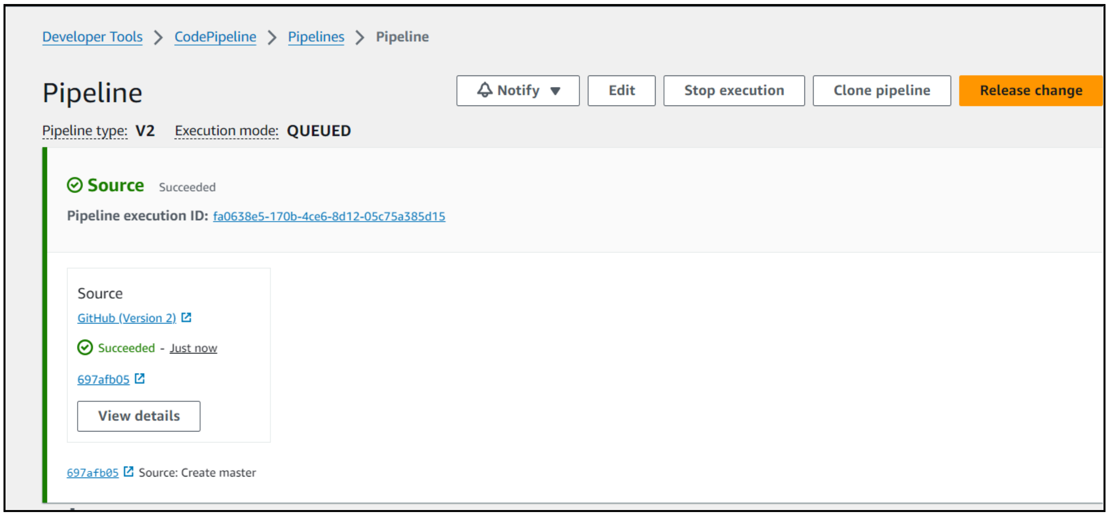
Deploy action provider
AWS Elastic Beanstalk

ApplicationName
Application1

EnvironmentName
Application1-env

Configure automatic rollback on stage failure
Disabled

Step 17: Pipeline is created successfully.



Developer Tools > CodePipeline > Pipelines > Pipeline

Pipeline

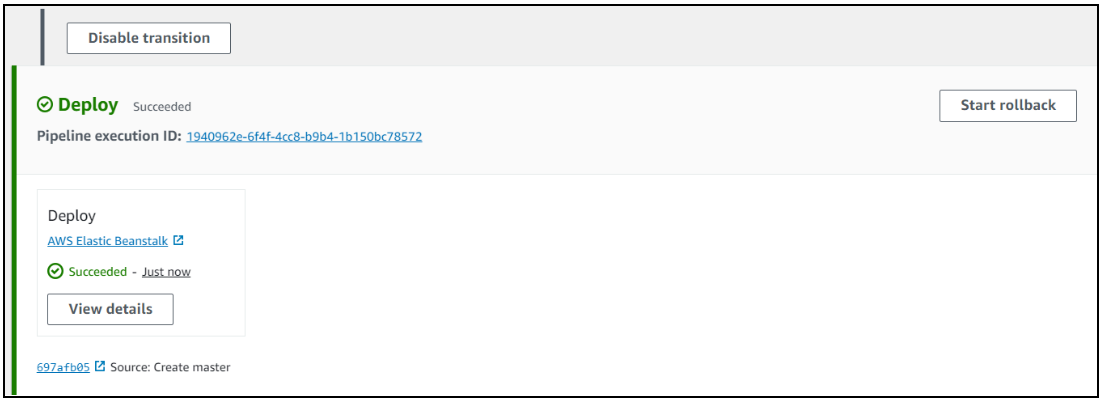
Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: fa0638e5-170b-4ce6-8d12-05c75a385d15

Source
GitHub (Version 2)
Succeeded - Just now
697afb05

697afb05 Source: Create master



Deploy Succeeded

Pipeline execution ID: 1940962e-6f4f-4cc8-b9b4-1b150bc78572

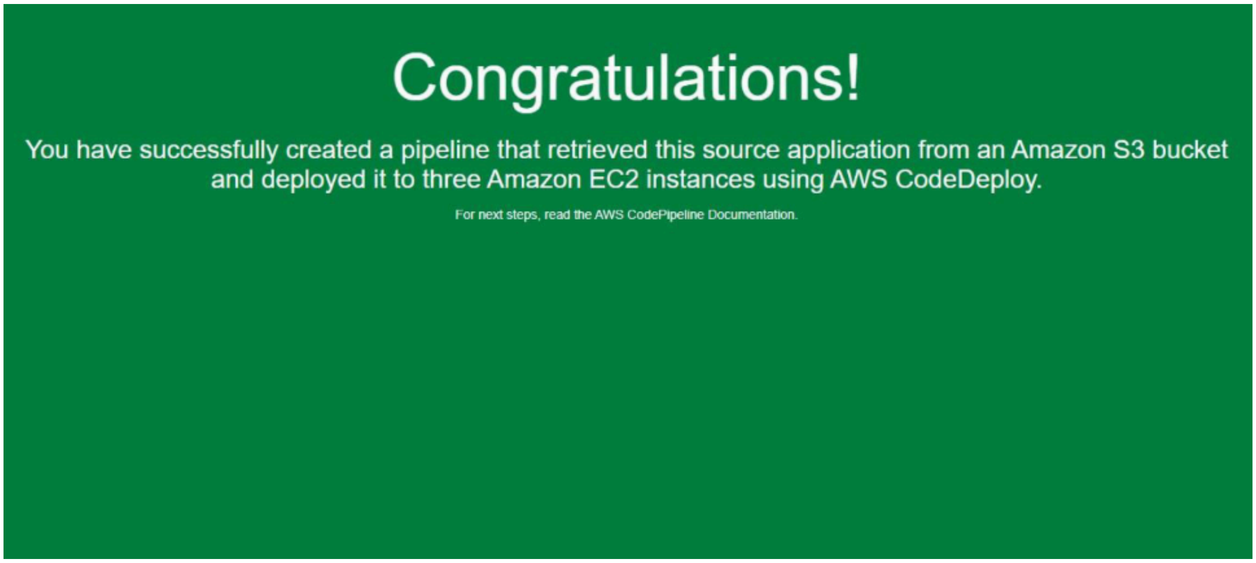
Deploy

AWS Elastic Beanstalk

Succeeded - Just now

697afb05 Source: Create master

Step 17: Now we will select the URL and it will redirect to sample website.



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.