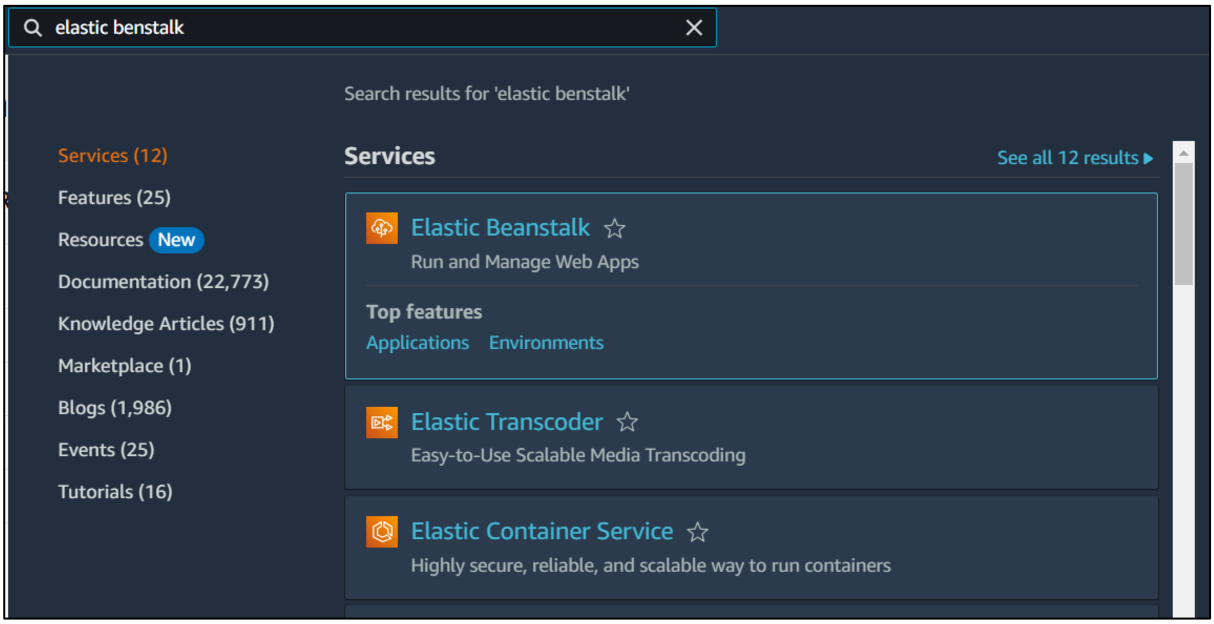
Advance DevOps

Experiment 2

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

Steps:

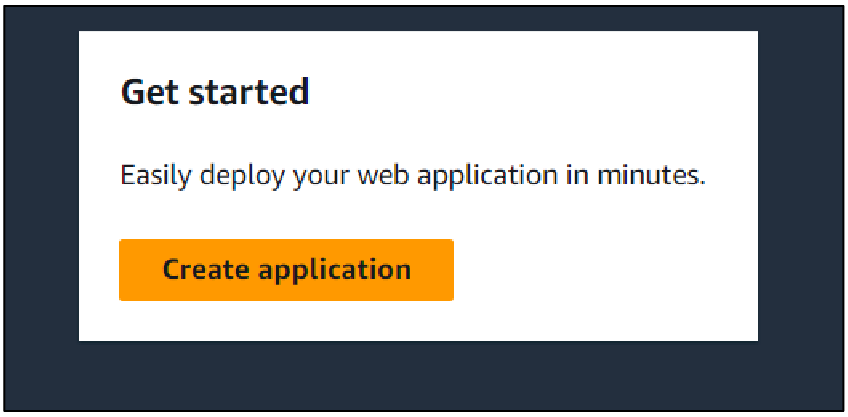
1. Firstly, we will create an environment and for that search elastic beanstalk in the services.



elastic benstalk

Search results for 'elastic benstalk'

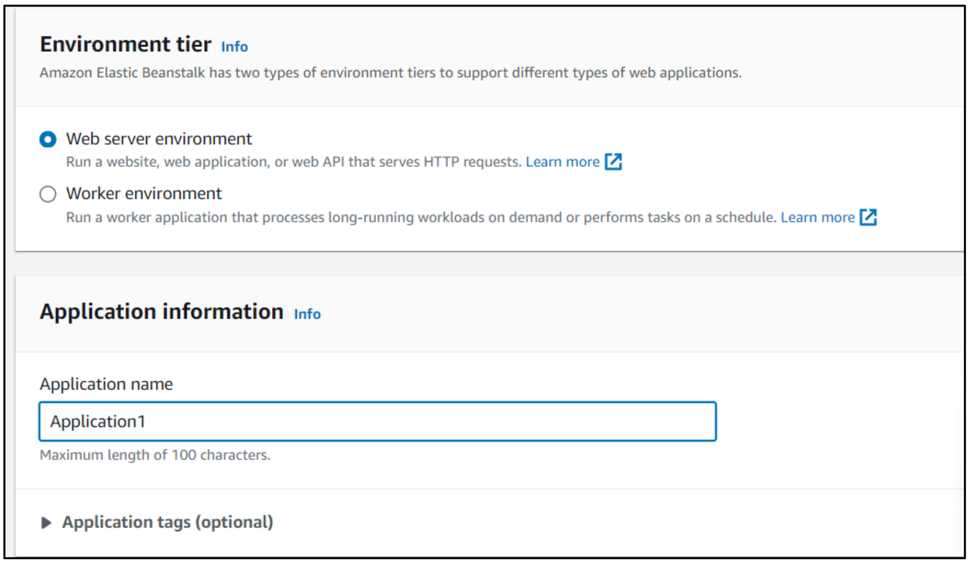
2. Select create application.



Get started

Easily deploy your web application in minutes.

3. Now we will configure the environment. In environment tier select web server environment and give application name.



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

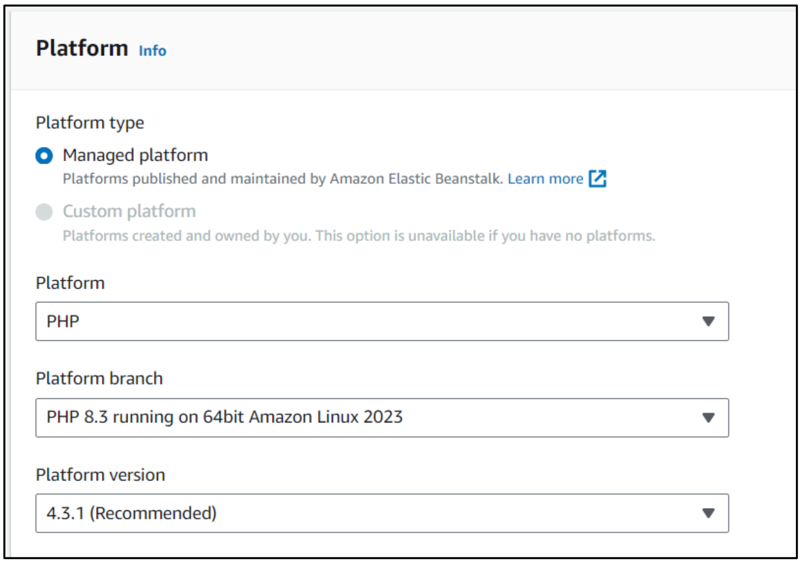
Application information Info

Application name: Application1

Maximum length of 100 characters.

▶ Application tags (optional)

4. Select a platform, here I have selected php.



Platform Info

Platform type

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

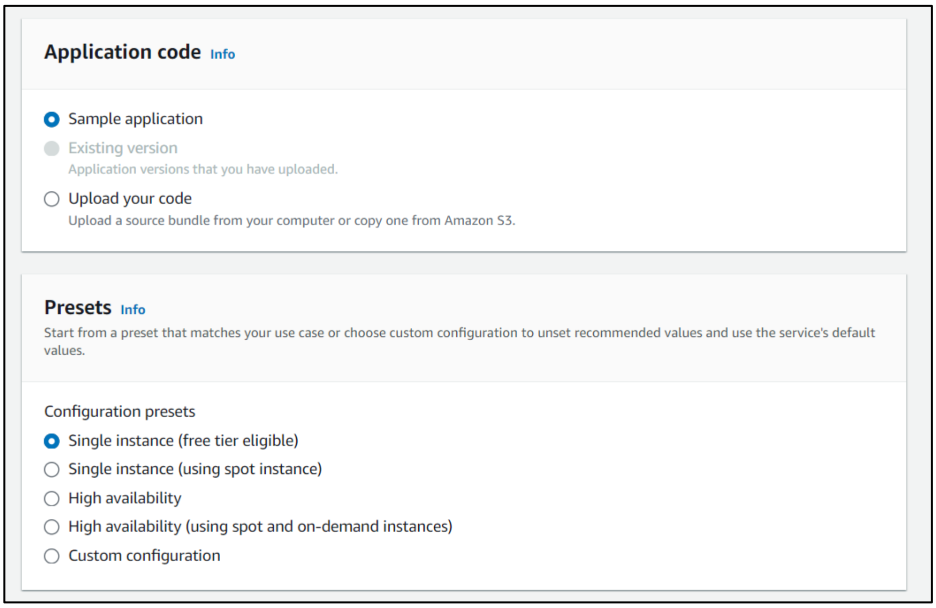
○ Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform: PHP

Platform branch: PHP 8.3 running on 64bit Amazon Linux 2023

Platform version: 4.3.1 (Recommended)

5. Keep all other settings at their default values



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)

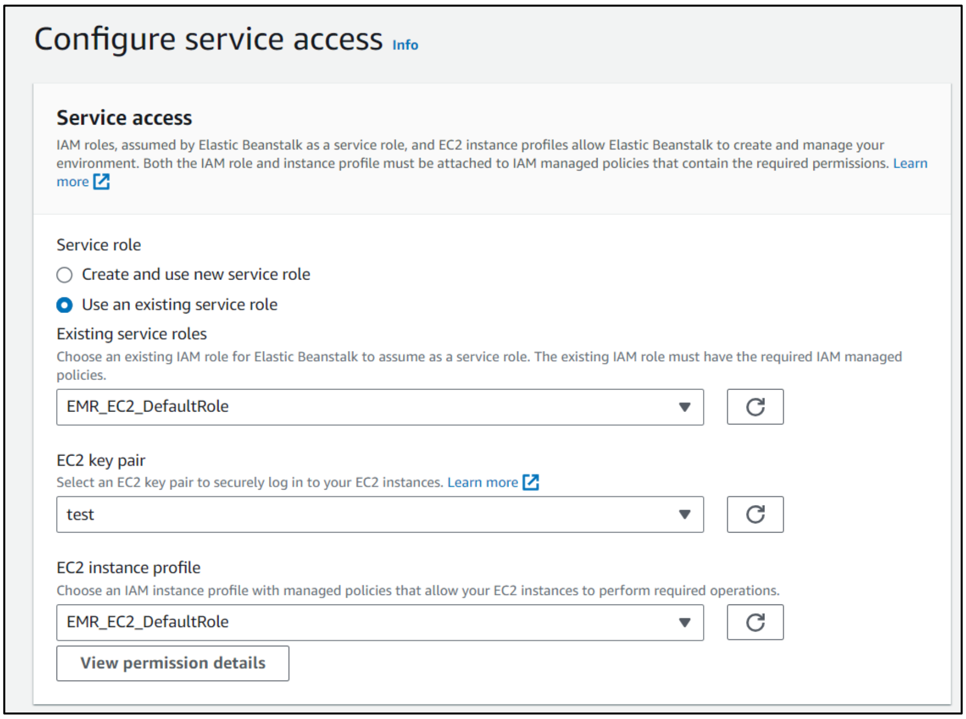
○ Single instance (using spot instance)

○ High availability

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

○ Custom configuration

6. Now we will configure service access. Select an existing service role, EC2 key pair and EC2 instance profile from the dropdowns given.



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

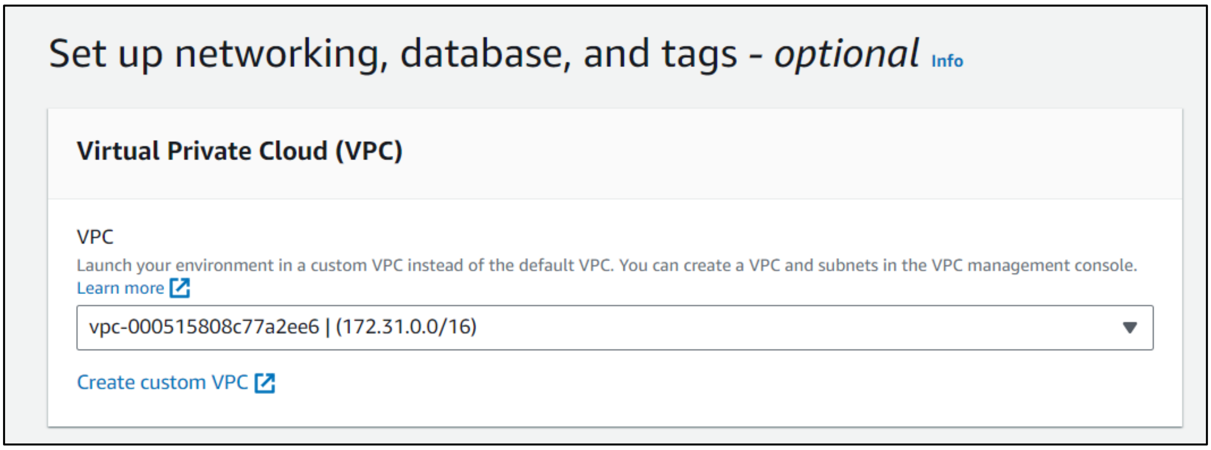
EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. Learn more

test

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

EMR_EC2_DefaultRole

7. In the setup network section select a vpc and instance subnets then click on next.



Set up networking, database, and tags - *optional* Info

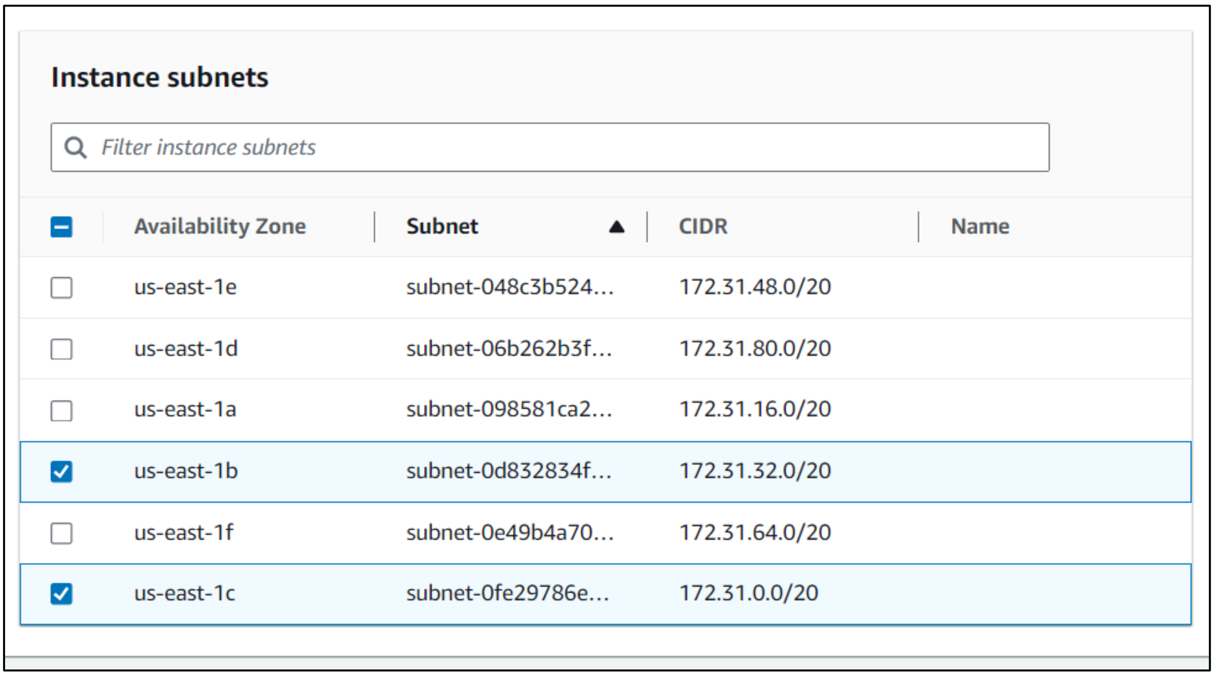
Virtual Private Cloud (VPC)

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-000515808c77a2ee6 | (172.31.0.0/16)

Create custom VPC

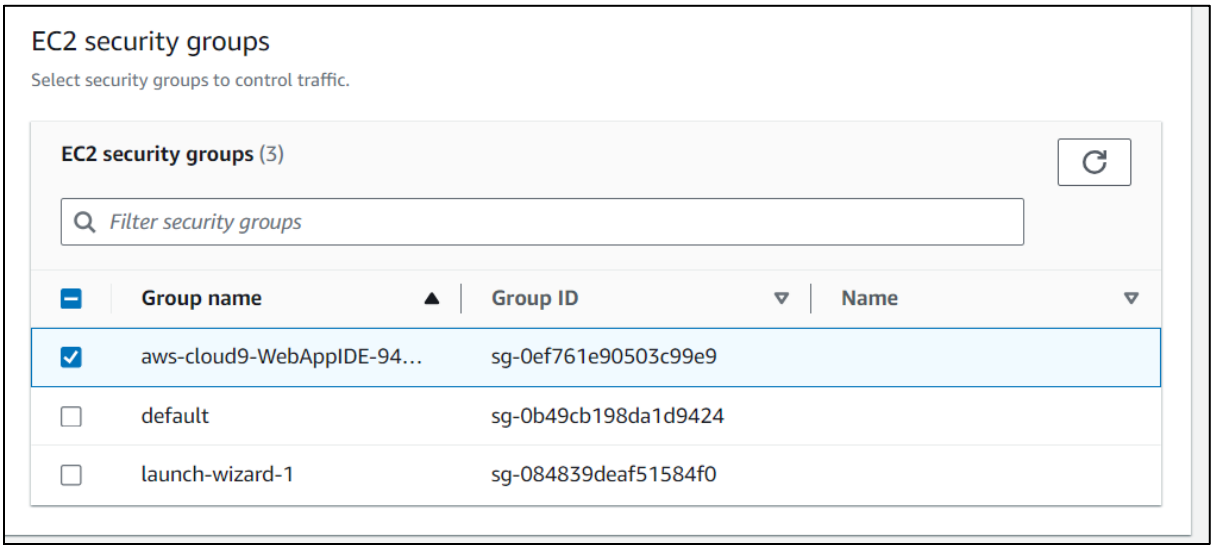


Instance subnets

Filter instance subnets

☐	Availability Zone	Subnet ▲	CIDR	Name
☐	us-east-1e	subnet-048c3b524...	172.31.48.0/20	
☐	us-east-1d	subnet-06b262b3f...	172.31.80.0/20	
☐	us-east-1a	subnet-098581ca2...	172.31.16.0/20	
☑	us-east-1b	subnet-0d832834f...	172.31.32.0/20	
☐	us-east-1f	subnet-0e49b4a70...	172.31.64.0/20	
☑	us-east-1c	subnet-0fe29786e...	172.31.0.0/20	

8. In the next step select EC2 security groups and set instance type as t2.micro



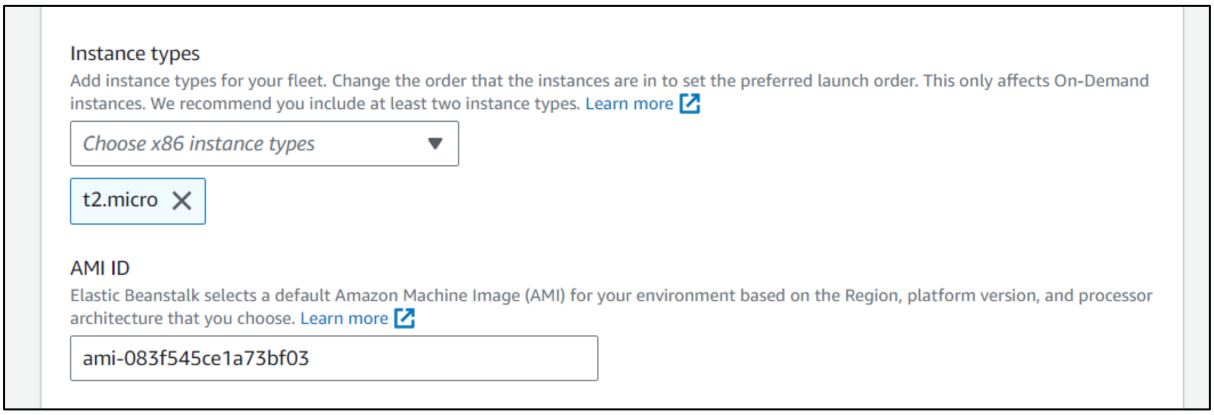
EC2 security groups

Select security groups to control traffic.

EC2 security groups (3)

Filter security groups

☐	Group name ▲	Group ID ▽	Name ▽
☑	aws-cloud9-WebAppIDE-94...	sg-0ef761e90503c99e9	
☐	default	sg-0b49cb198da1d9424	
☐	launch-wizard-1	sg-084839deaf51584f0	



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

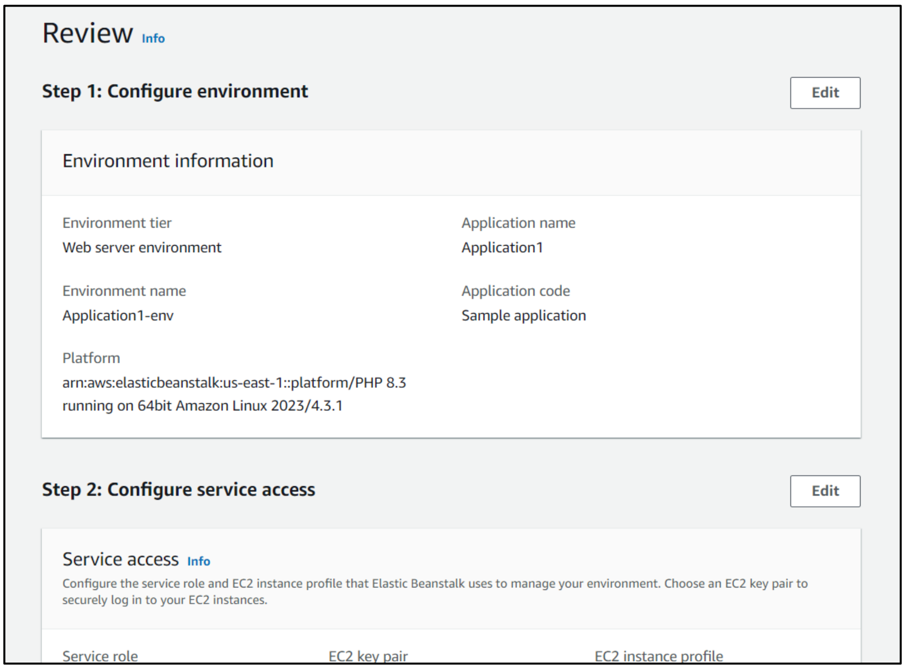
t2.micro

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

9. Next is the review page where you can check the configurations that have been set in the previous steps. Click on submit.



Review Info

Step 1: Configure environment

Environment information

Environment tier Web server environment	Application name Application1
Environment name Application1-env	Application code Sample application
Platform arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	

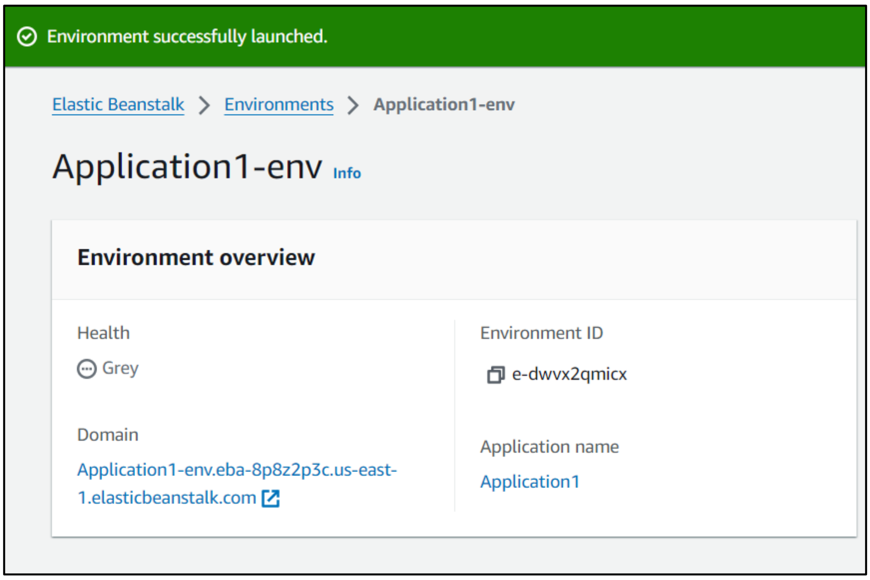
Step 2: Configure service access

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

10. Your environment will be successfully created.



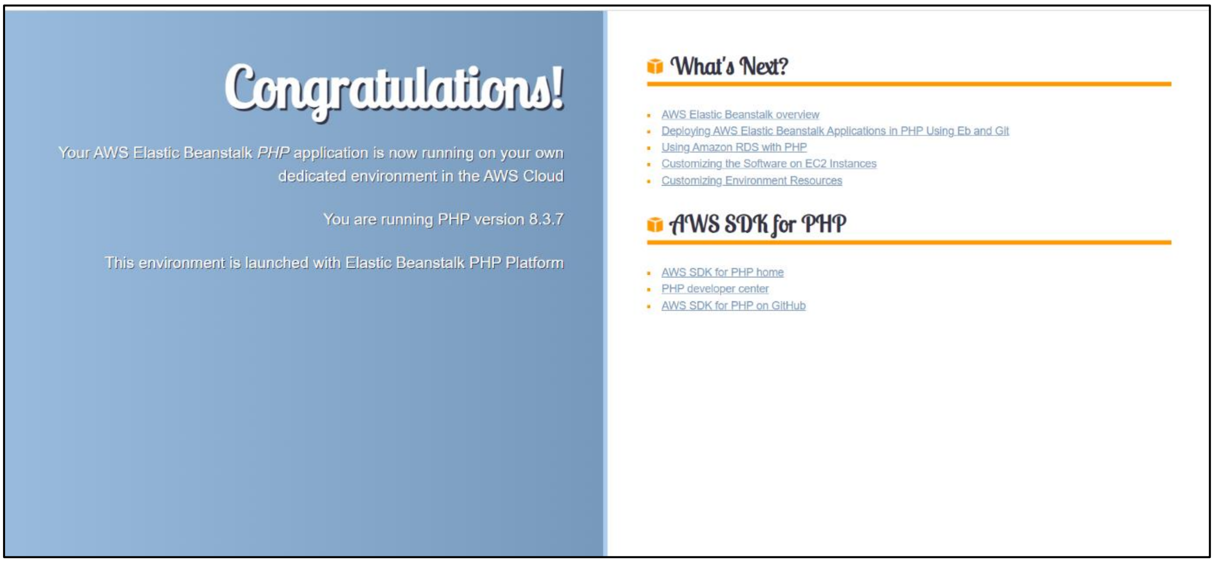
Environment successfully launched.

Application1-env Info

Environment overview

Health Grey	Environment ID e-dwvx2qmicx
Domain Application1-env.eba-8p8z2p3c.us-east-1.elasticbeanstalk.com	Application name Application1

11. After clicking on domain link the page given below will open:



Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 8.3.7

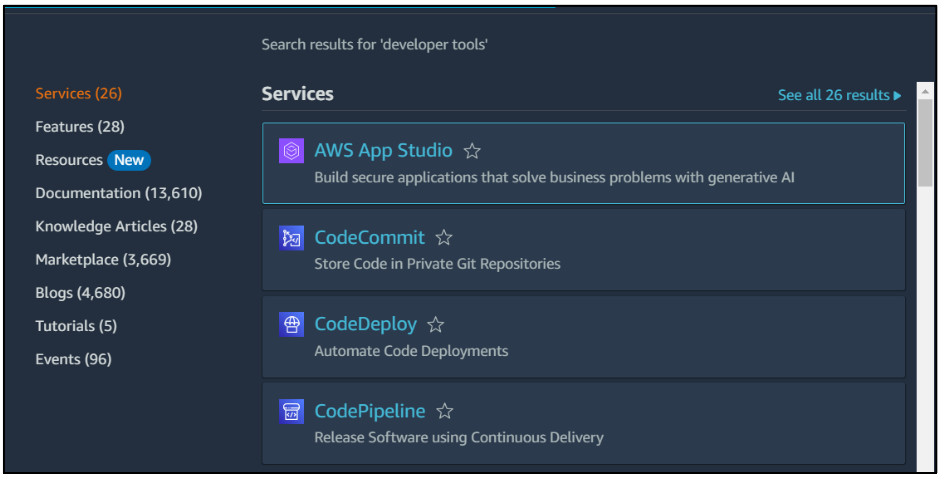
This environment is launched with Elastic Beanstalk PHP Platform

What's Next?

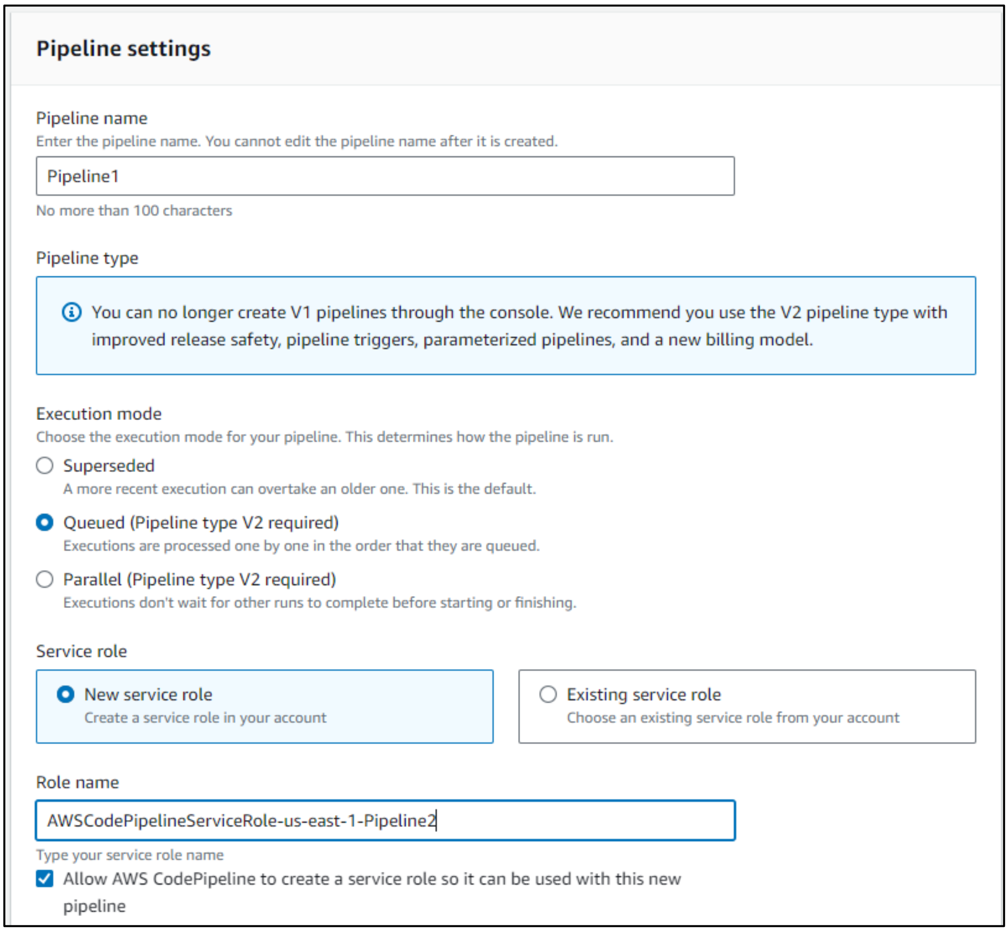
- AWS Elastic Beanstalk overview
- Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
- Using Amazon RDS with PHP
- Customizing the Software on EC2 Instances
- Customizing Environment Resources

AWS SDK for PHP

- AWS SDK for PHP home
- PHP developer center
- AWS SDK for PHP on GitHub



12. Now we will create a pipeline. Go to services and select CodePipeline.



Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
Pipeline1
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.

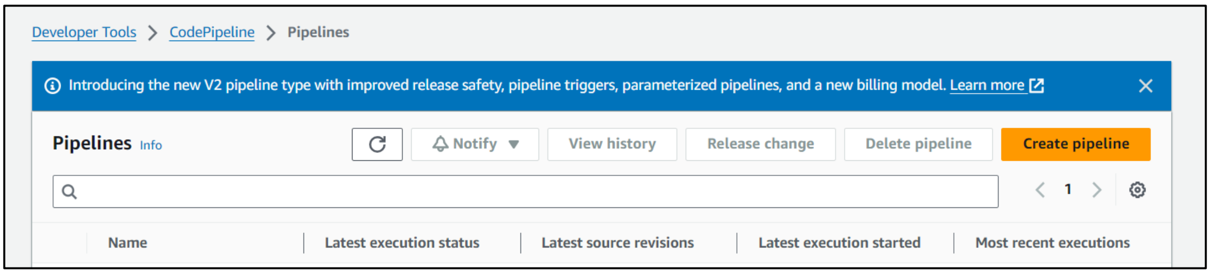
Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.
○ Superseded
A more recent execution can overtake an older one. This is the default.
● Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.
○ Parallel (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
AWSCodePipelineServiceRole-us-east-1-Pipeline2
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

13. Select create pipeline.

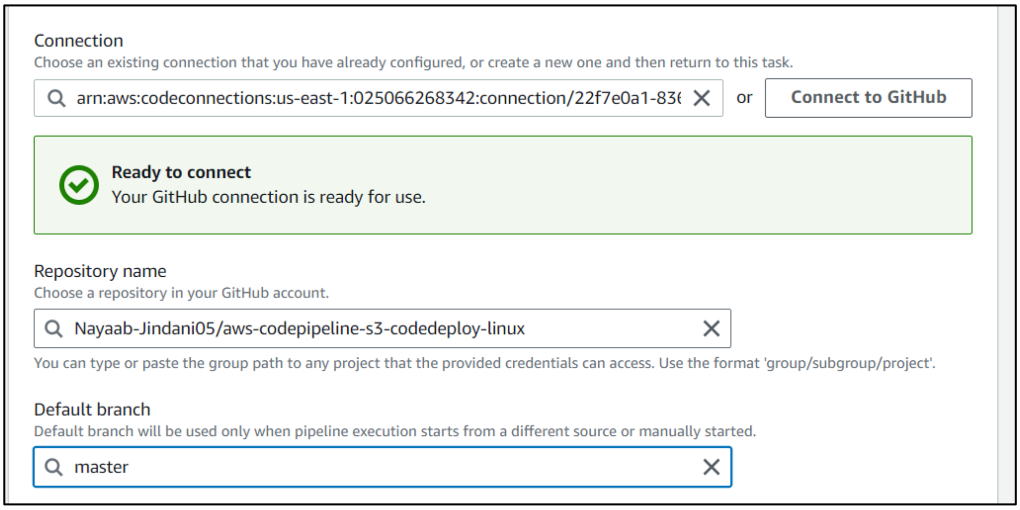
14. In step 1 give the pipeline name



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

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



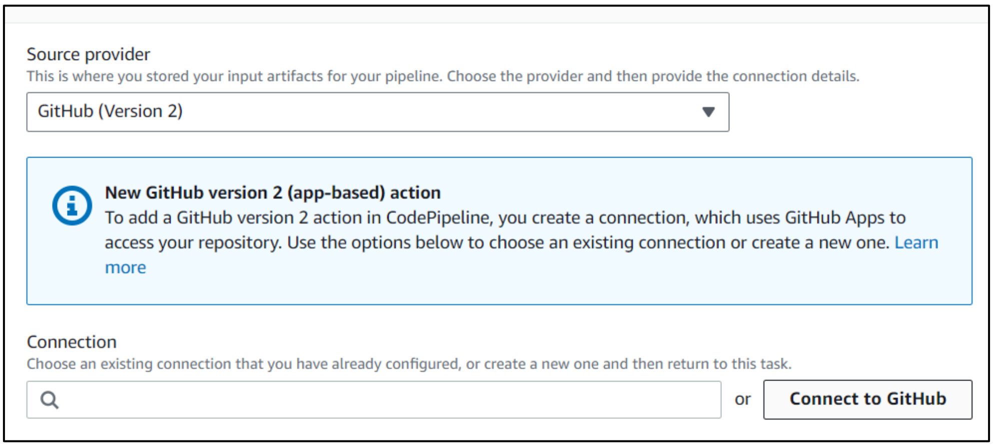
Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.
arn:aws:codeconnections:us-east-1:025066268342:connection/22f7e0a1-83(✕ or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.
Nayaab-Jindani05/aws-codepipeline-s3-codedeploy-linux ✕
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.
master ✕

15. In Add source stage select GitHub version2 as the source provider and then connect your AWS to GitHub account and if there is a connection which already exists then select that.



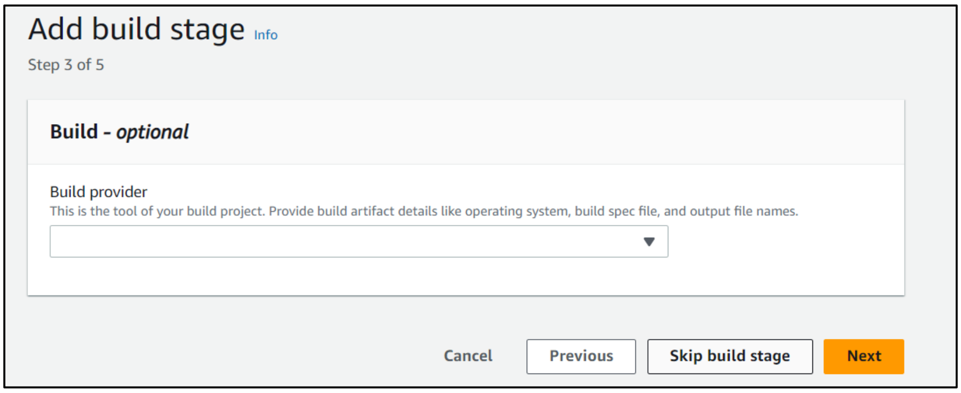
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 2) ▼

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 Connect to GitHub

16. After connecting to GitHub select the repository name and default branch.

17. Skip the build stage and go to deploy stage



Add build stage

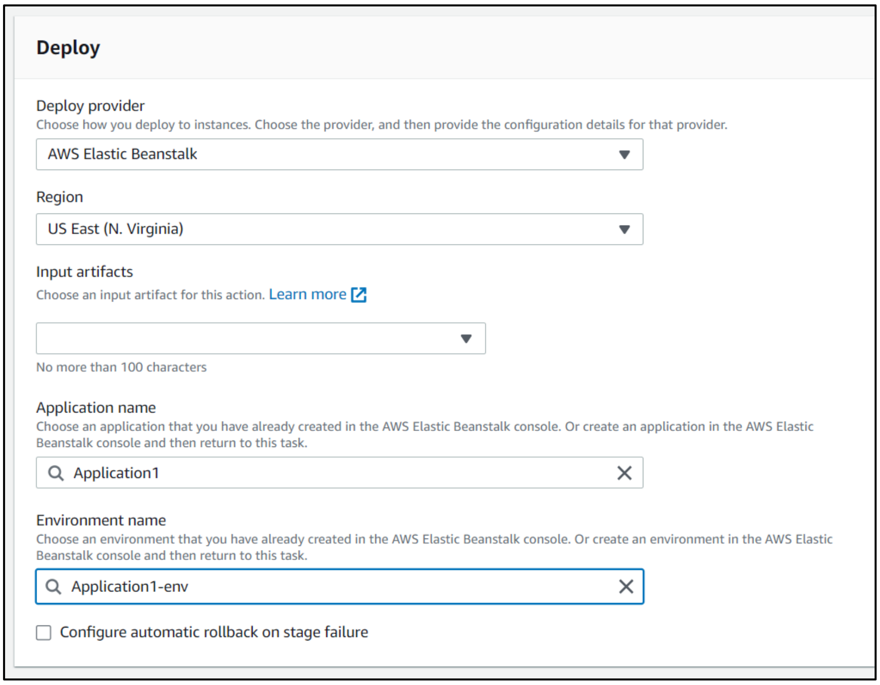
Step 3 of 5

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.

18. In the deploy stage select AWS Elastic Beanstalk as the deploy provider and also select the application name, environment name created earlier.



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.

Application1

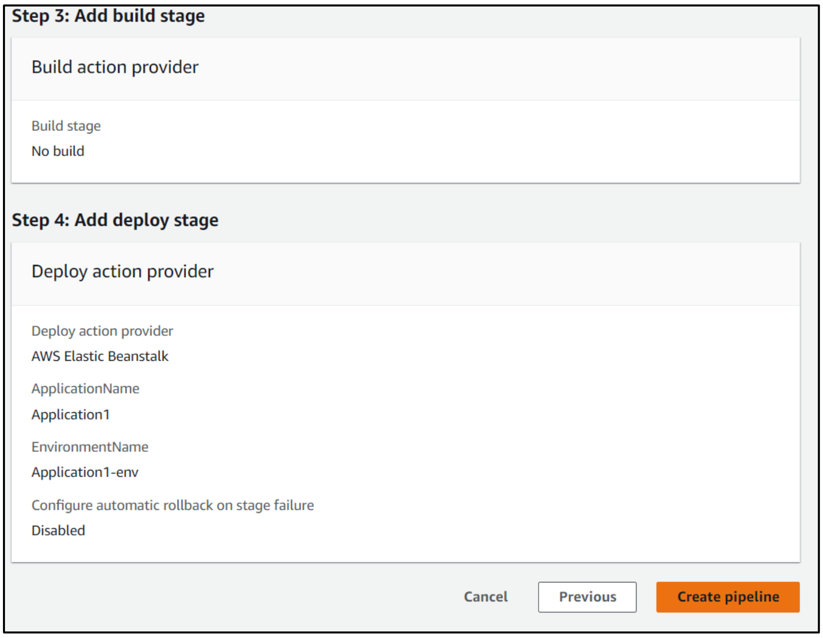
Environment name

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.

Application1-env

□ Configure automatic rollback on stage failure

19. In the review stage check all the settings that have been done and select 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

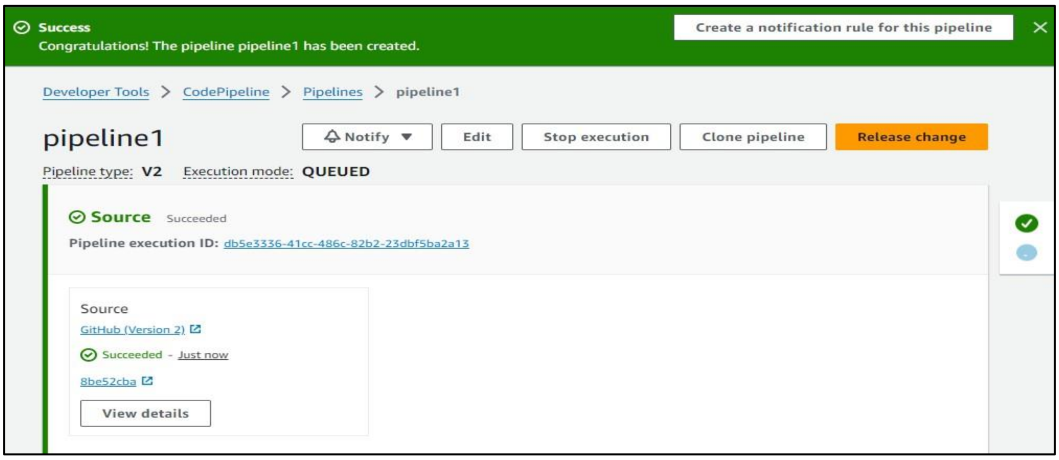
EnvironmentName

Application1-env

Configure automatic rollback on stage failure

Disabled

20. This screen means that pipeline creation is successful



Success
Congratulations! The pipeline pipeline1 has been created.

Create a notification rule for this pipeline
Developer Tools > CodePipeline > Pipelines > pipeline1

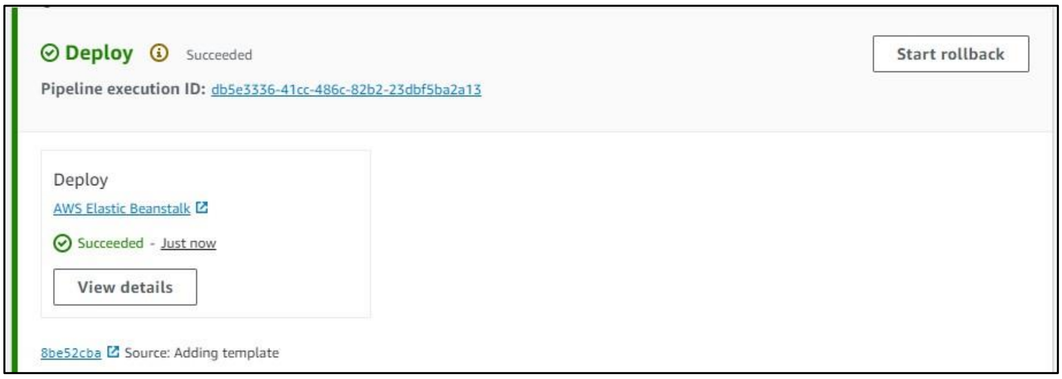
pipeline1

Notify
Edit
Stop execution
Clone pipeline
Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: db5e3336-41cc-486c-82b2-23dbf5ba2a13

Source
GitHub (Version 2)
Succeeded - Just now
8be52cba

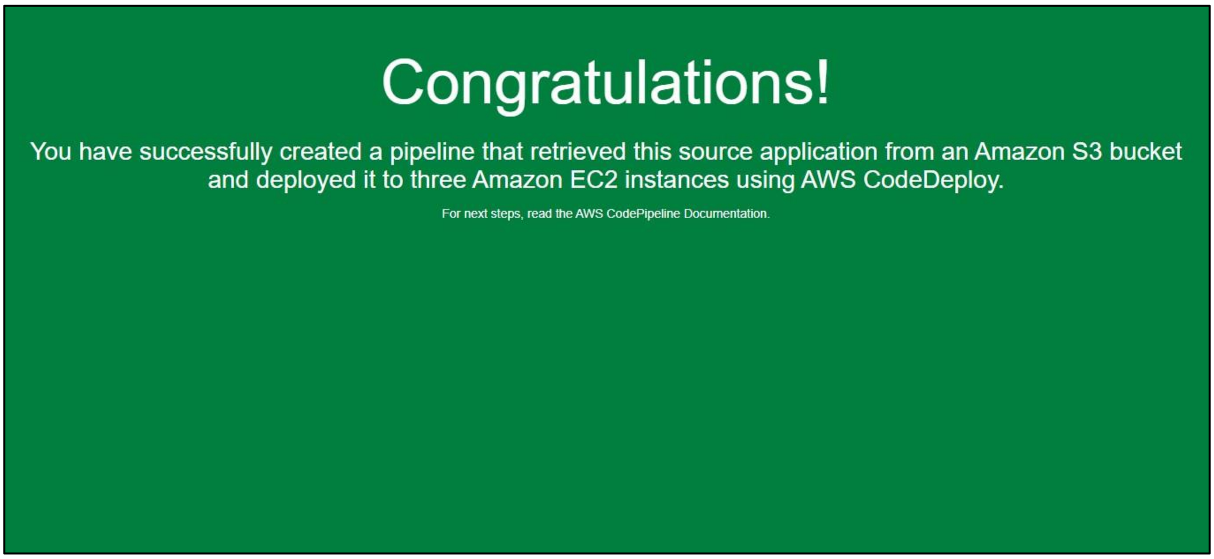


Deploy Succeeded
Pipeline execution ID: db5e3336-41cc-486c-82b2-23dbf5ba2a13

Deploy
AWS Elastic Beanstalk
Succeeded - Just now

8be52cba Source: Adding template

21. Now we can select the URL and it will open a sample website that we have created.



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.