

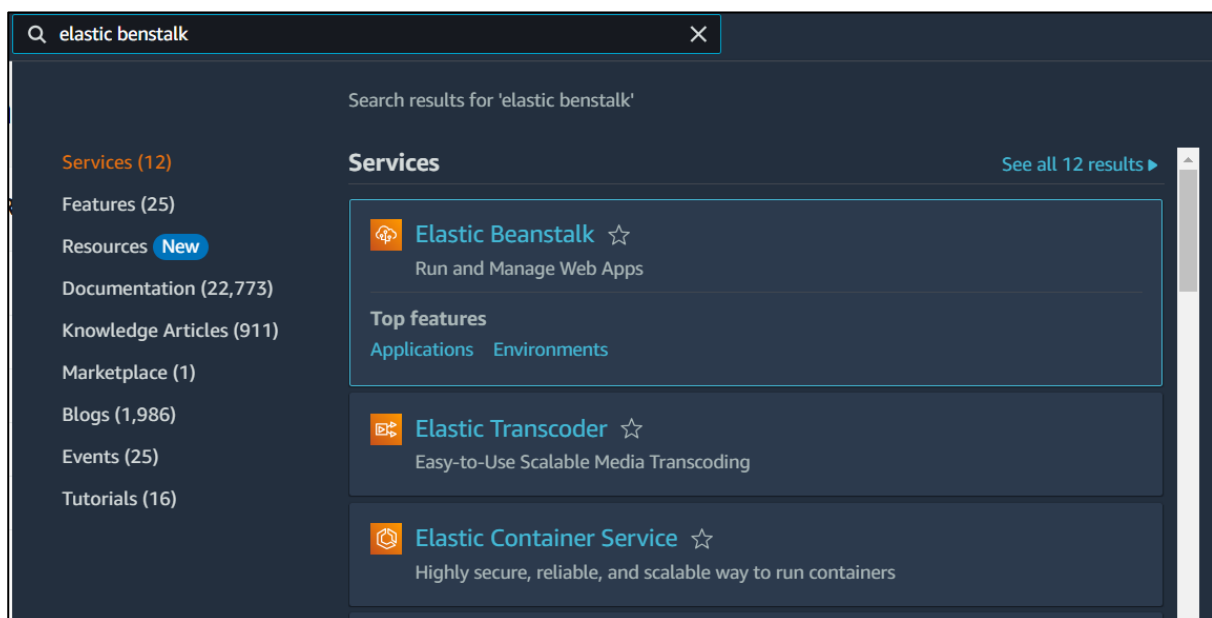
Advance DevOps

Experiment 2

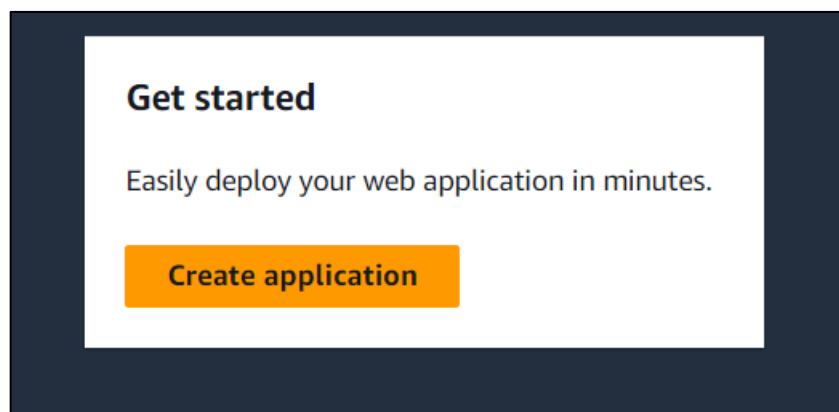
To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Steps:

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



2. Select create application.



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

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

Platform branch

Platform version

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

[View permission details](#)

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

	Availability Zone	Subnet ▲	CIDR	Name
<input type="checkbox"/>	us-east-1e	subnet-048c3b524...	172.31.48.0/20	
<input type="checkbox"/>	us-east-1d	subnet-06b262b3f...	172.31.80.0/20	
<input type="checkbox"/>	us-east-1a	subnet-098581ca2...	172.31.16.0/20	
<input checked="" type="checkbox"/>	us-east-1b	subnet-0d832834f...	172.31.32.0/20	
<input type="checkbox"/>	us-east-1f	subnet-0e49b4a70...	172.31.64.0/20	
<input checked="" type="checkbox"/>	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)

	Group name ▲	Group ID ▼	Name ▼
<input checked="" type="checkbox"/>	aws-cloud9-WebAppIDE-94...	sg-0ef761e90503c99e9	
<input type="checkbox"/>	default	sg-0b49cb198da1d9424	
<input type="checkbox"/>	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](#)

Choose x86 instance types ▼

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 [Edit](#)

Environment information

Environment tier	Application name
Web server environment	Application1
Environment name	Application code
Application1-env	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 [Edit](#)

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.

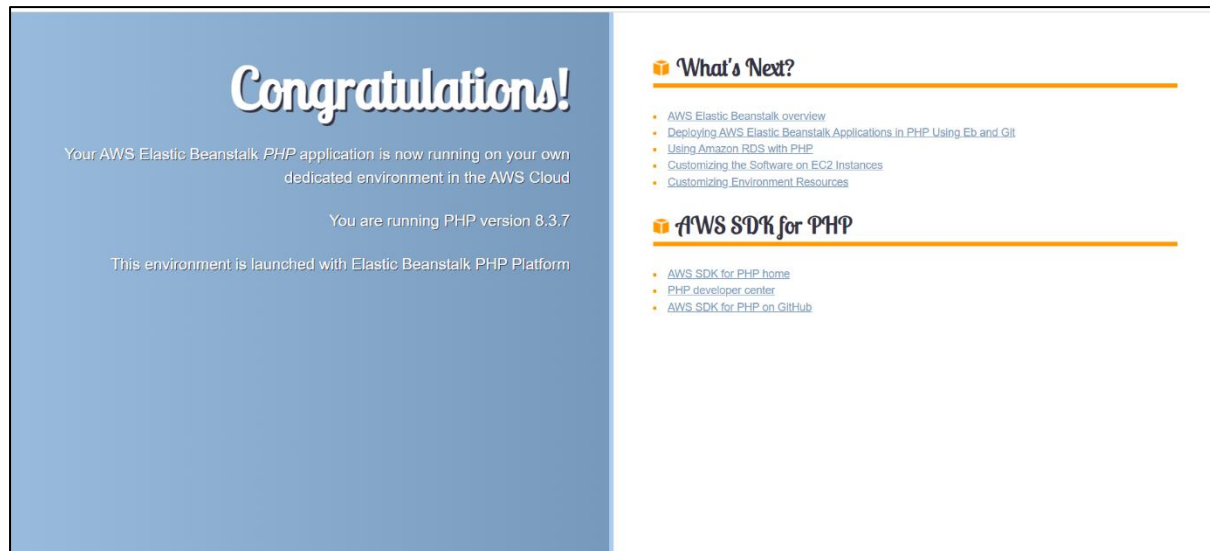
[Elastic Beanstalk](#) > [Environments](#) > Application1-env

Application1-env [Info](#)

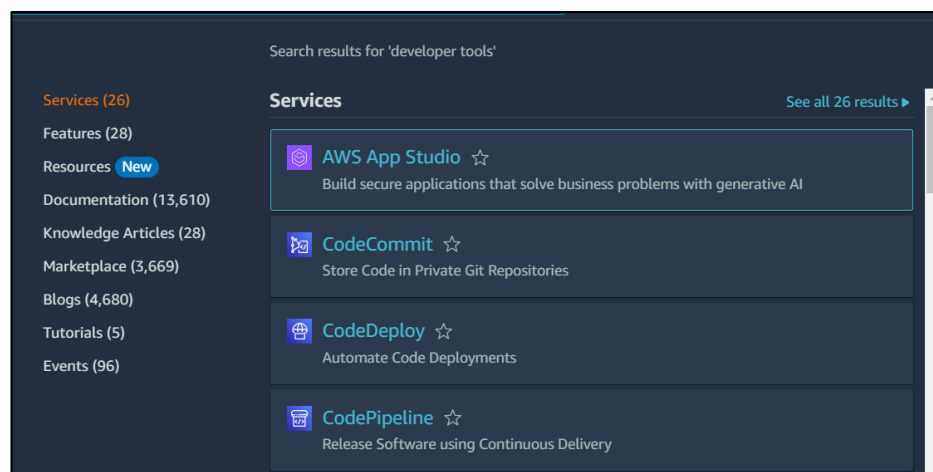
Environment overview

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

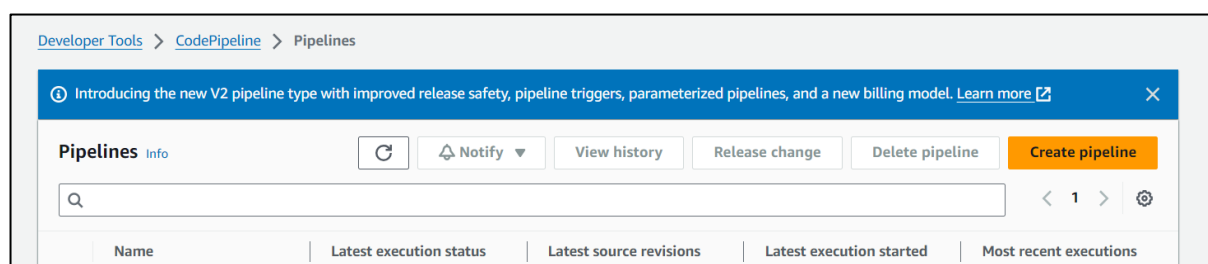
11. After clicking on domain link the page given below will open:



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



13. Select create pipeline.



14. In step 1 give the pipeline name

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

Type your service role name

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

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

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

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

or



Ready to connect
Your GitHub connection is ready for use.

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.

17. Skip the build stage and go to deploy 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.

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

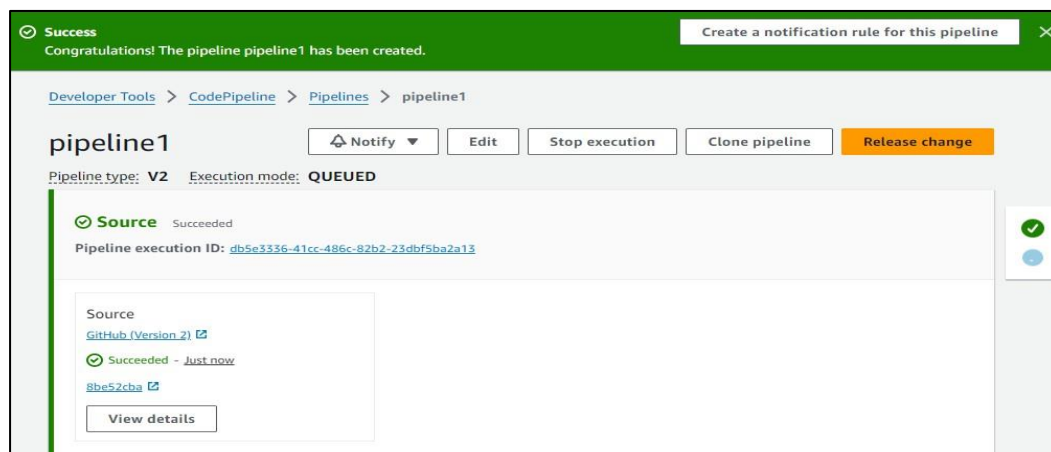
Application1-env

Configure automatic rollback on stage failure

Disabled

Cancel Previous Create pipeline

20. This screen means that pipeline creation is successful



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

