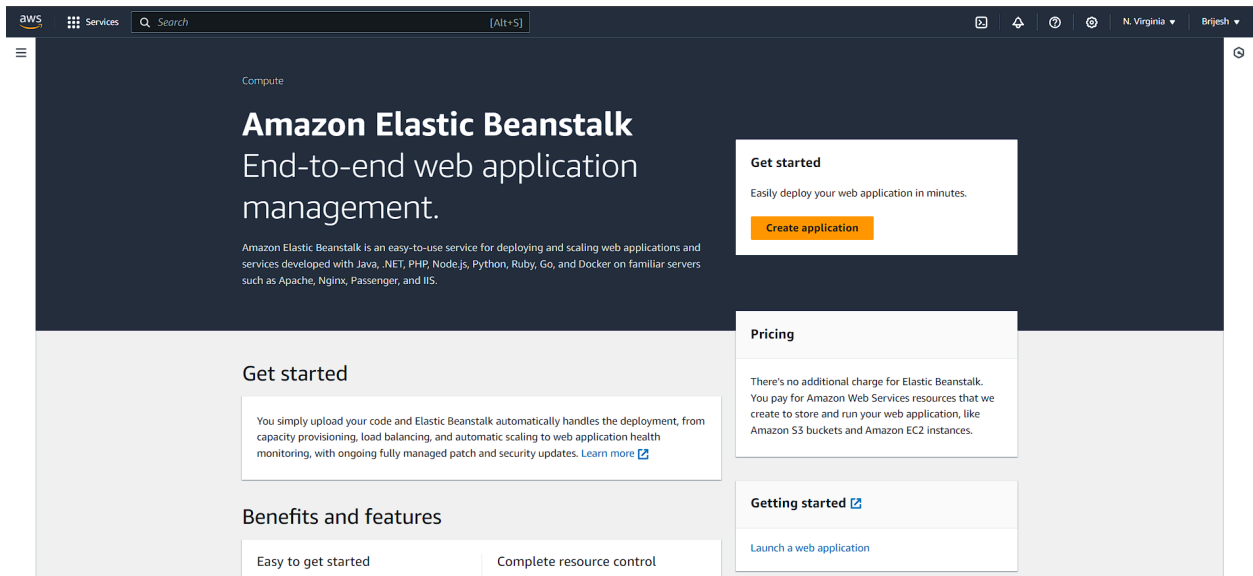
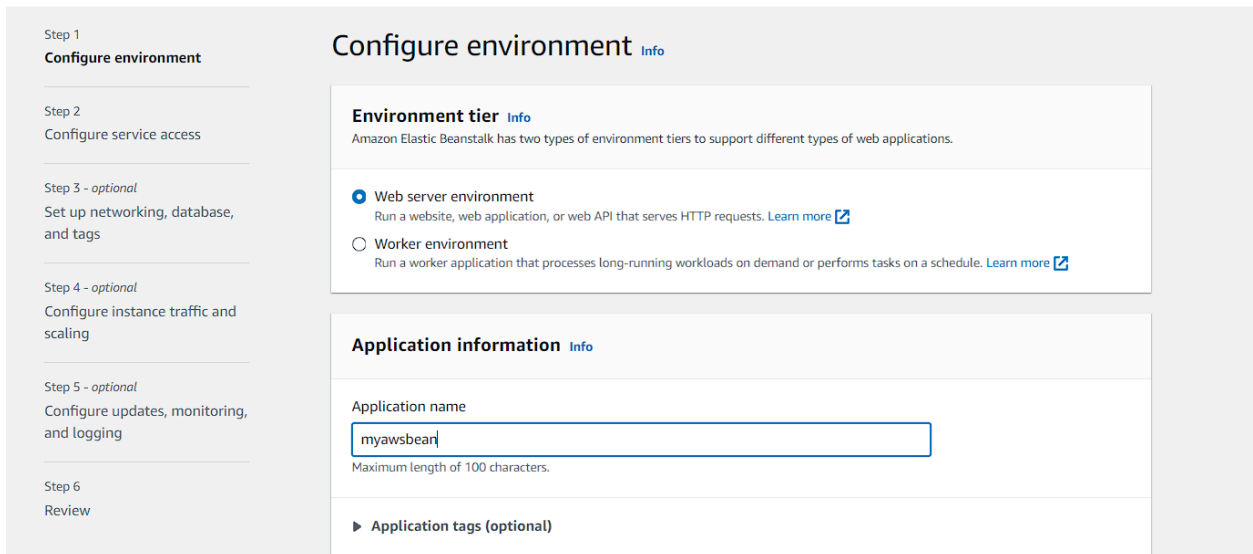


Aim : To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy

1) Open the aws console and then search Elastic Beanst



2) Click on create application and configure the environment by adding your application name.



### 3) Select the environment as PHP and other options as default and click on next.

#### Environment information [Info](#)

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

Environment name

Myawsbean-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.

Domain

Leave blank for autogenerated value

.us-east-1.elasticbeanstalk.com

Check availability

Environment description

#### 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

4.3.2 (Recommended)

#### 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

Cancel

Next

- 4) After clicking on Next for creating Elastic Beanstalk, we need key-pair that will be require for deployment. Go to EC2 Instance and click on Key Pairs.

The screenshot shows the AWS Management Console interface. The 'Resources' section at the top displays a grid of EC2 resources in the US East (N. Virginia) Region, including Instances (running), Elastic IPs, Load balancers, Snapshots, Auto Scaling Groups, Instances, Placement groups, Volumes, Dedicated Hosts, Key pairs, and Security groups. The 'Launch instance' section below it provides instructions on how to launch an instance, with a 'Launch instance' button and a 'Migrate a server' link. To the right, the 'Service health' section shows the status of the Region (US East (N. Virginia)) as 'operating normally'. Below that, a table lists the available Zones with their names and IDs. On the far right, the 'EC2 Free Tier' section shows that 0 offers are currently in use, and the 'Account attributes' section displays the Default VPC and various settings.

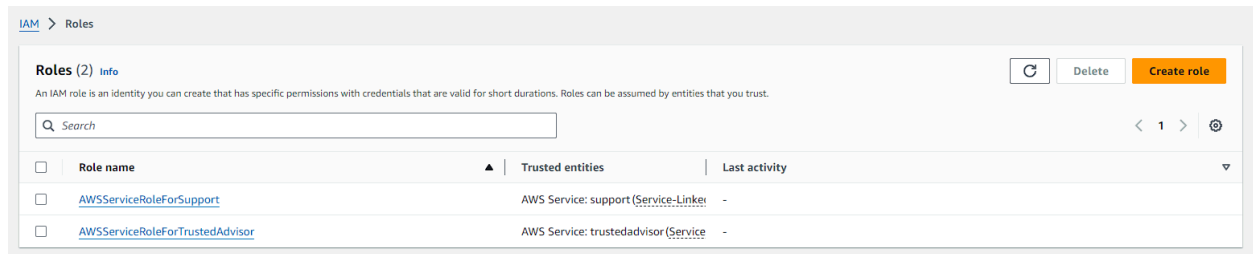
- 5) Then click on Create key pair

The screenshot shows the 'Key pairs' page in the AWS Management Console. At the top, there is a search bar and a 'Create key pair' button. Below the search bar, a table lists the key pairs, with columns for Name, Type, Created, Fingerprint, and ID. The table is currently empty, displaying 'No key pairs to display'.

- 6) Input the name of the key-pair and select pem as file format and click on Create key pair.

The screenshot shows the 'Create key pair' form in the AWS Management Console. The form has a title 'Create key pair' and a brief description of a key pair. It includes a 'Name' field with the value 'the\_key'. Below the name field, there is a 'Key pair type' section with two options: 'RSA' (selected) and 'ED25519'. The 'Private key file format' section has two options: '.pem' (selected) and '.ppk'. At the bottom, there is a 'Tags - optional' section with a note that no tags are currently associated with the resource and an 'Add new tag' button. The form concludes with 'Cancel' and 'Create key pair' buttons.

- 7) After creating key pair, open new tab and go to IAM to create a role that will be used to build Codepipeline. Click on Create role.



- 8) Select AWS service as Trusted Entity type and EC2 as service.

**Trusted entity type**

☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case  
EC2

Choose a use case for the specified service.  
Use case

☒ **EC2**  
Allows EC2 instances to call AWS services on your behalf.

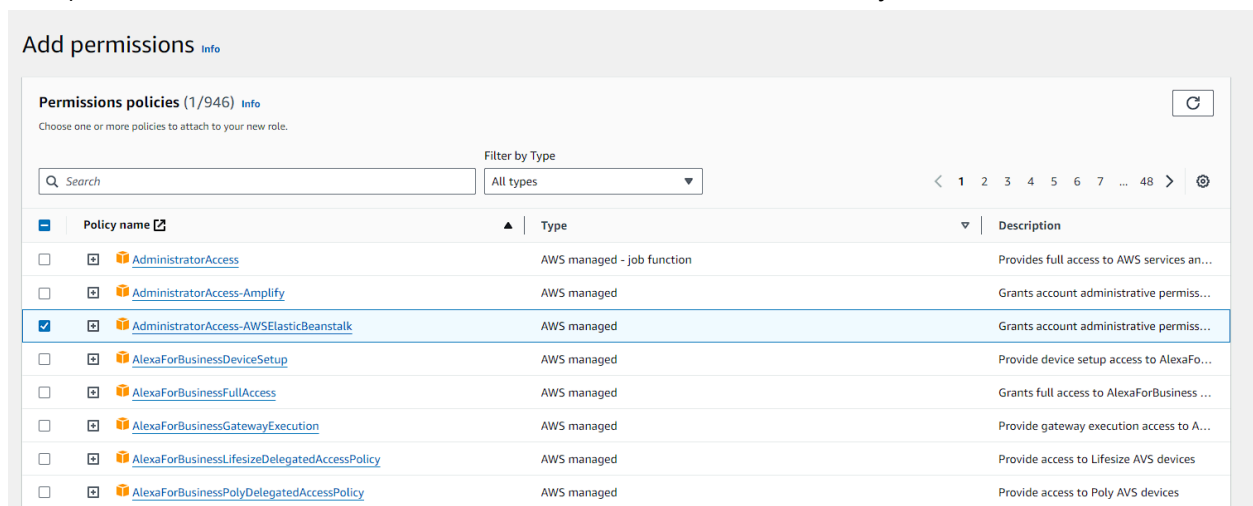
☐ **EC2 Role for AWS Systems Manager**  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

☐ **EC2 Spot Fleet Role**  
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

☐ **EC2 - Spot Fleet Auto Scaling**  
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

☐ **EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

- 9) Choose AdministratorAccess-AWSElasticBeanstalk as Policy and click on Next.



10) Name the role and keep other as default.

Role details

Role name

Enter a meaningful name to identify this role.

new-user

Maximum 64 characters. Use alphanumeric and "+-,@-\_" characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+-.@/[]{}#\$%'^\*~:"

Step 1: Select trusted entities

Trust policy

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": [
11          "ec2.amazonaws.com"
12        ]
13      }
14    ]
15  }
16 }
```

11) The role is created successfully.

IAM > Roles

Roles (3)

Info

Refresh Delete Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

< 1 > @

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service)	-
<input type="checkbox"/>	<a href="#">new-user</a>	AWS Service: ec2	-

12) Now move to the tab where Elastic Beanstalk was opened and from the drop down menu select the newly created key pair and instance profile. Now let everything be default.

Step 1  
Configure environment

Step 2  
Configure service access

Step 3 - optional  
Set up networking, database, and tags

Step 4 - optional  
Configure instance traffic and scaling

Step 5 - optional  
Configure updates, monitoring, and logging

Step 6  
Review

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

Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

View permission details

EC2 key pair

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

new-key

Refresh

EC2 instance profile

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

new-user

Refresh

View permission details

Cancel Skip to review Previous 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](#)

- ▼

[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

No instance subnets  
No instance subnets to display

## 13) Review the changes and click on Create.

## Review [Info](#)

### Step 1: Configure environment

Edit

#### Environment information

Environment tier	Application name
Web server environment	myawsbean
Environment name	Application code
Myawsbean-env	Sample application
Platform	
am:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2	

### Step 2: Configure service access

Edit

#### Service access [Info](#)

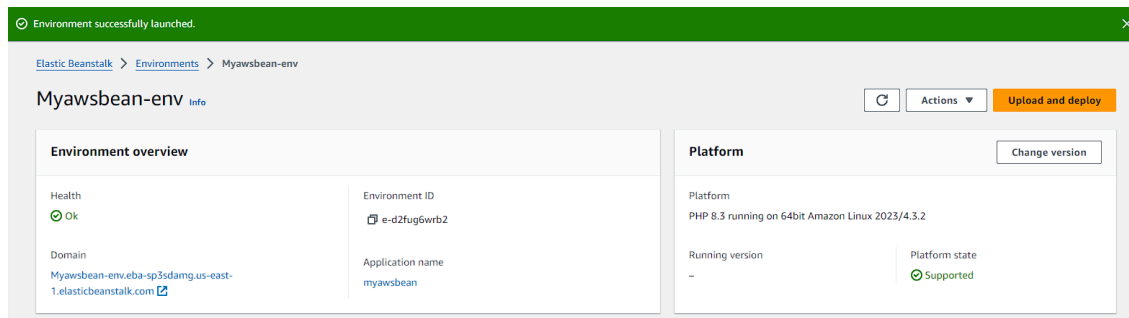
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
arn:aws:iam::017820672175:role/service-role/aws-elasticbeanstalk-service-role	the_key	new-user

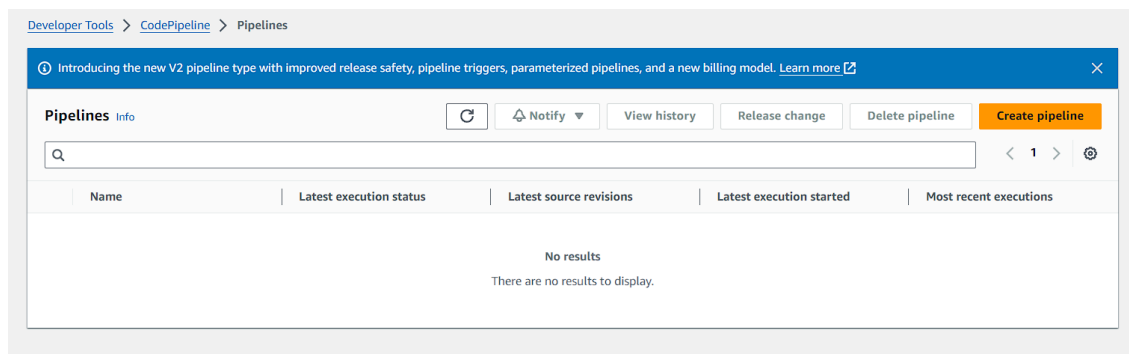
### Step 3: Set up networking, database, and tags

Edit

14) Your sample environment is created for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.



15) Now, we need to make a CodePipeline. Go to CodePipeline and click on Create Pipeline.



16) Name the pipeline and select the service role as below and click on Next.

## Choose pipeline settings Info

Step 1 of 5


### 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


### Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

Add variable

You can add up to 50 variables.

 The first pipeline execution will fail if variables have no default values.

► Advanced settings

Cancel

Next



17) In the source stage select Github v2 as the provider and then connect your github connect so that the pipeline can access the forked source code. Name the connection.

[Developer Tools](#) > [Connections](#) > Create connection

## Create a connection Info


### Create GitHub App connection Info

Connection name

► Tags - *optional*

[Connect to GitHub](#)

18) Signin to GitHub to connect with AWS.



Sign in to GitHub  
to continue to AWS Connector for  
GitHub

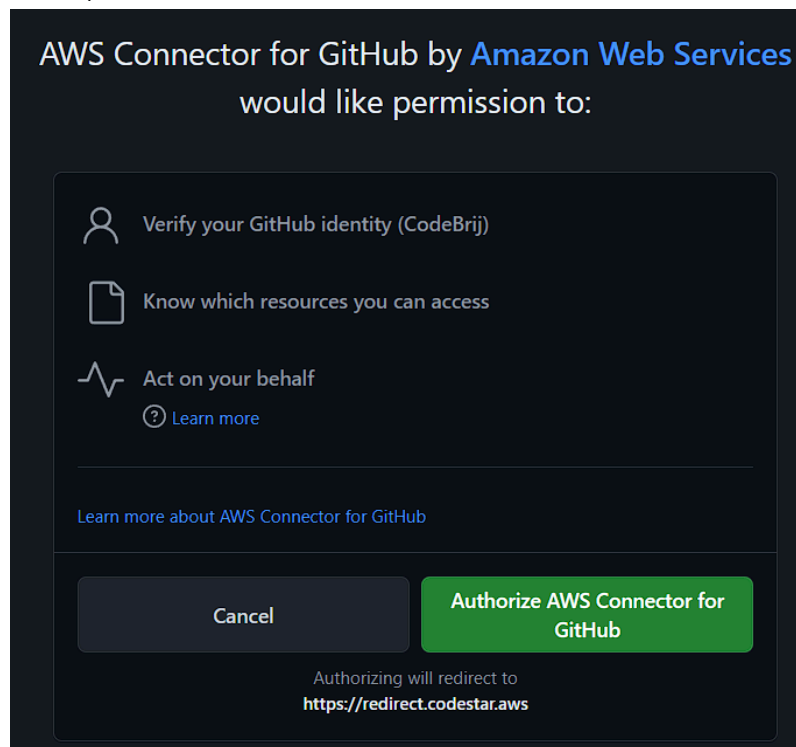
---

Username or email address

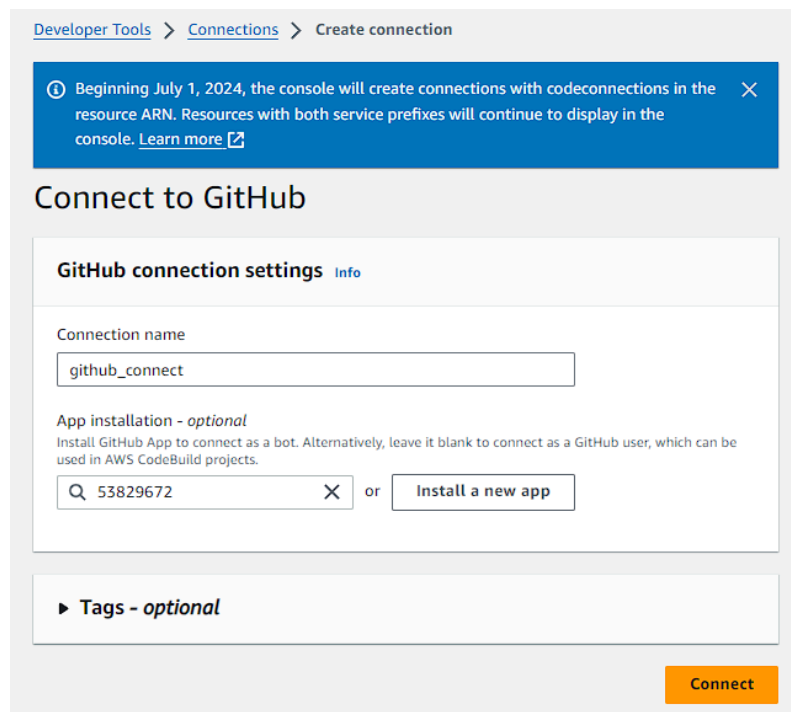
Password [Forgot password?](#)

[Sign in](#)

19) Authorize AWS Connector for GitHub.



20) We need to install the GitHub connector.




21) Now, select the repository and the branch to be deployed.

**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-1:017820672175:connection/ca9502e9-40...

X

or

Connect to GitHub

 **Ready to connect**  
Your GitHub connection is ready for use.

**Repository name**  
Choose a repository in your GitHub account.

Q

CodeBrij/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.

22) Select No filter in Trigger.


**Trigger**

**Trigger type**  
Choose the trigger type that starts your pipeline.

☒ **No filter**  
Starts your pipeline on any push and clones the HEAD.

☐ **Specify filter**  
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

☐ **Do not detect changes**  
Don't automatically trigger the pipeline.

 You can add additional sources and triggers by editing the pipeline after it is created.

23) In deploy stage add application name as environment name. Then review the settings and click on Create pipeline.

### 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](#)

▼

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.

Q myawsbean X

**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.

Q Myawsbean-env X

☐ Configure automatic rollback on stage failure

24) The pipeline is ready and the provided repo is deployed successfully.

**Success**  
Congratulations! The pipeline mypipeline has been created.

Create a notification rule for this pipeline X

Developer Tools > CodePipeline > Pipelines > mypipeline

### mypipeline

Pipeline type: V2 Execution mode: QUEUED

Notify Edit Stop execution Clone pipeline Release change

**Source** In progress

Pipeline execution ID: P907680e-bb64-4c49-a188-356bba8f99f

Source

In progress - Just now

View details

Disable transition

**Deploy** Didn't Run

Start rollback

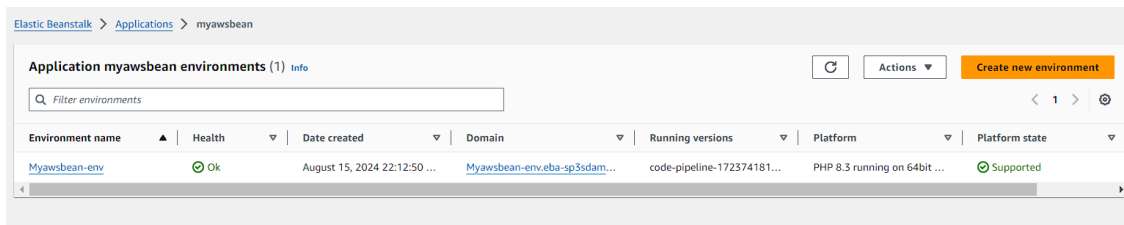
Deploy

AWS Elastic Beanstalk

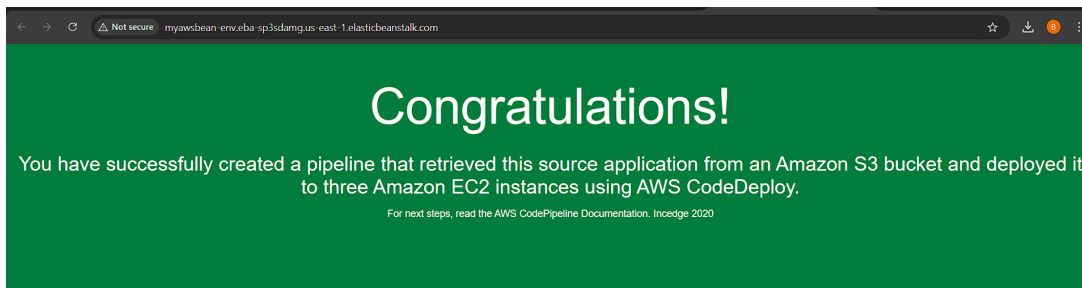
Didn't Run

No executions yet

25) Go to Elastic Beanstalk and from Domain open the hosted site.



26) Hosted site from the github repository.



27) Make some changes in the index.html and reload.

