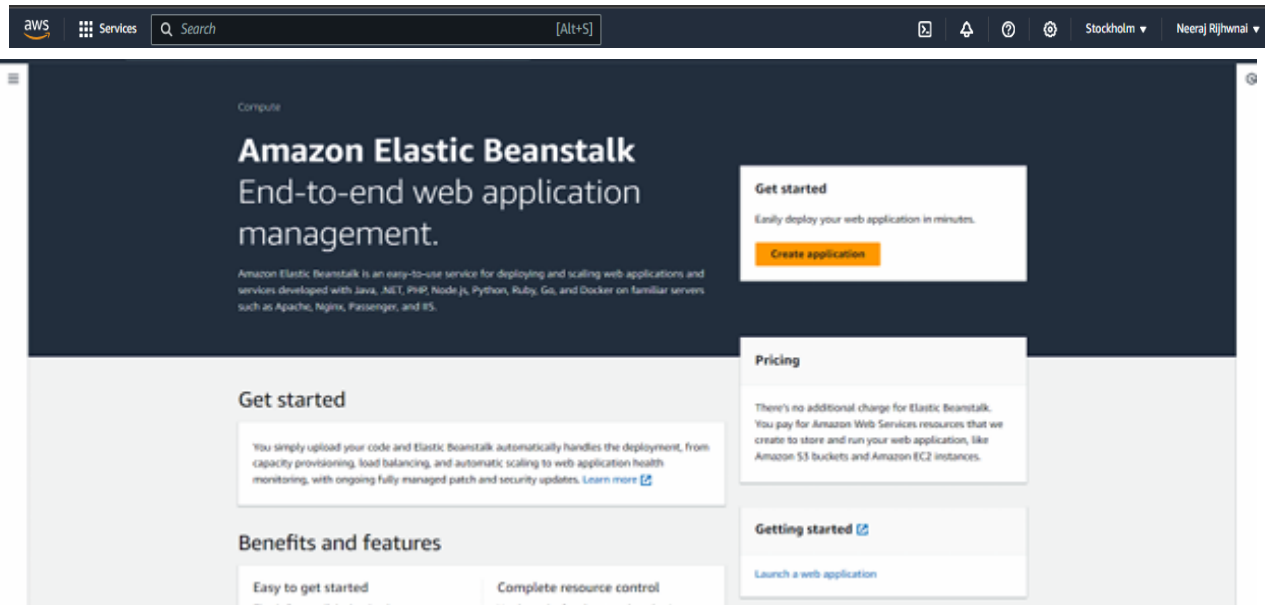


## EXPERIMENT 2

**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.

The screenshot shows the 'Configure environment' step in the AWS Elastic Beanstalk console. On the left, there's a sidebar with a list of steps: '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', and 'Step 6: Review'. The main content area is titled 'Configure environment' and contains two sections. The first section is 'Environment tier', which has two radio button options: 'Web server environment' (selected) and 'Worker environment'. The second section is 'Application information', which has a text input field for 'Application name' containing the text 'myawsbear'. Below the input field, there's a note: 'Maximum length of 100 characters.' At the bottom, there's a section for 'Application tags (optional)'.

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

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

.us-east-1.elasticbeanstalk.comCheck 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

Platform branch

Platform version

### 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.

**Resources** [EC2 Global View](#) [🔍](#) [🔔](#) [🔗](#)

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	1
Snapshots	0	Volumes	0		

**Launch instance**  
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#) [Migrate a server](#)

Note: Your instances will launch in the US East (N. Virginia) Region

**Instance alarms** [View in CloudWatch](#)

0 in alarm 0 OK 0 insufficient data

[Instances in alarm](#)

**Scheduled events** [🔗](#)

**Service health** [AWS Health Dashboard](#) [🔗](#) [🔔](#)

Region: US East (N. Virginia) Status: ✔ This service is operating normally.

**Zones**

Zone name	Zone ID
us-east-1a	use1-az4
us-east-1b	use1-az6
us-east-1c	use1-az1
us-east-1d	use1-az2
us-east-1e	use1-az3
us-east-1f	use1-az5

**EC2 Free Tier** [Info](#)  
Offers for all AWS Regions.

0 EC2 free tier offers in use

End of month forecast  
⚠ 0 offers forecasted to exceed free tier limit.

Exceeds free tier  
⚠ 0 offers exceeded and is now pay-as-you-go pricing.

[View Global EC2 resources](#)

[View all AWS Free Tier offers](#)

**Account attributes** [🔗](#)

**Default VPC** [🔗](#)  
vpc-07b4b6571f0c46e01

**Settings**  
[Data protection and security](#)  
[Zones](#)  
[EC2 Serial Console](#)  
[Default credit specification](#)  
[EC2 console preferences](#)

**Additional information** [🔗](#)

5) Then click on Create key pair

**Key pairs** [Info](#)

[🔗](#) [Actions](#) [Create key pair](#)

Name	Type	Created	Fingerprint	ID
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.

EC2 > Key pairs > Create key pair

## Create key pair Info

**Key pair**  
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

**Name**  
  
 The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type** Info  
☒ RSA ☐ ED25519

**Private key file format**  
☒ .pem  
 For use with OpenSSH  
☐ .ppk  
 For use with PuTTY

**Tags - optional**  
 No tags associated with the resource.  
  
 You can add up to 50 more tags.

- 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.

IAM > Roles

**Roles (2)** Info

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.

<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-Linker)	-

- 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

Choose a use case for the specified service.

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

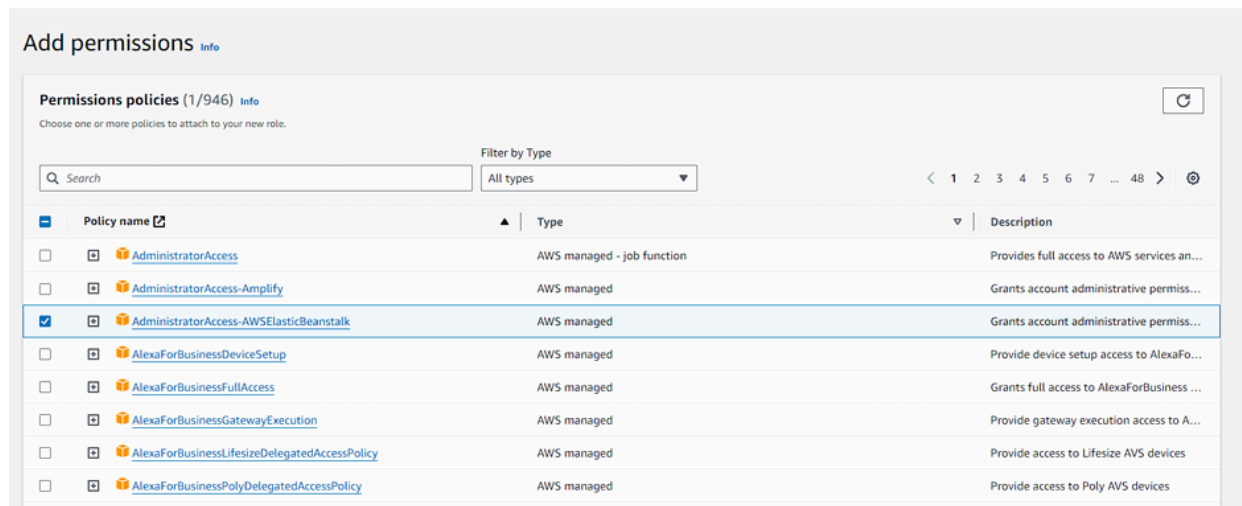
☐ **EC2 Role for AWS Systems Manager**  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

☐ **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: '\_', '+', '@', '/', '!', '#', '\$', '%', '^', '&', '\*', '~', '(', ')', '[', ']', '{', '}', '&#34;', '&#39;', '&#124;', '&#92;', '&#58;', '&#44;', '&#5B;', '&#5D;', '&#5C;', '&#5E;', '&#5F;', '&#60;', '&#62;', '&#63;', '&#64;', '&#65;', '&#66;', '&#67;', '&#68;', '&#69;', '&#70;', '&#71;', '&#72;', '&#73;', '&#74;', '&#75;', '&#76;', '&#77;', '&#78;', '&#79;', '&#80;', '&#81;', '&#82;', '&#83;', '&#84;', '&#85;', '&#86;', '&#87;', '&#88;', '&#89;', '&#90;', '&#91;', '&#92;', '&#93;', '&#94;', '&#95;', '&#96;', '&#97;', '&#98;', '&#99;', '&#100;', '&#101;', '&#102;', '&#103;', '&#104;', '&#105;', '&#106;', '&#107;', '&#108;', '&#109;', '&#110;', '&#111;', '&#112;', '&#113;', '&#114;', '&#115;', '&#116;', '&#117;', '&#118;', '&#119;', '&#120;', '&#121;', '&#122;', '&#123;', '&#124;', '&#125;', '&#126;', '&#127;', '&#128;', '&#129;', '&#130;', '&#131;', '&#132;', '&#133;', '&#134;', '&#135;', '&#136;', '&#137;', '&#138;', '&#139;', '&#140;', '&#141;', '&#142;', '&#143;', '&#144;', '&#145;', '&#146;', '&#147;', '&#148;', '&#149;', '&#150;', '&#151;', '&#152;', '&#153;', '&#154;', '&#155;', '&#156;', '&#157;', '&#158;', '&#159;', '&#160;', '&#161;', '&#162;', '&#163;', '&#164;', '&#165;', '&#166;', '&#167;', '&#168;', '&#169;', '&#170;', '&#171;', '&#172;', '&#173;', '&#174;', '&#175;', '&#176;', '&#177;', '&#178;', '&#179;', '&#180;', '&#181;', '&#182;', '&#183;', '&#184;', '&#185;', '&#186;', '&#187;', '&#188;', '&#189;', '&#190;', '&#191;', '&#192;', '&#193;', '&#194;', '&#195;', '&#196;', '&#197;', '&#198;', '&#199;', '&#200;', '&#201;', '&#202;', '&#203;', '&#204;', '&#205;', '&#206;', '&#207;', '&#208;', '&#209;', '&#210;', '&#211;', '&#212;', '&#213;', '&#214;', '&#215;', '&#216;', '&#217;', '&#218;', '&#219;', '&#220;', '&#221;', '&#222;', '&#223;', '&#224;', '&#225;', '&#226;', '&#227;', '&#228;', '&#229;', '&#230;', '&#231;', '&#232;', '&#233;', '&#234;', '&#235;', '&#236;', '&#237;', '&#238;', '&#239;', '&#240;', '&#241;', '&#242;', '&#243;', '&#244;', '&#245;', '&#246;', '&#247;', '&#248;', '&#249;', '&#250;', '&#251;', '&#252;', '&#253;', '&#254;', '&#255;'

**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.

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-Linker)	-
<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

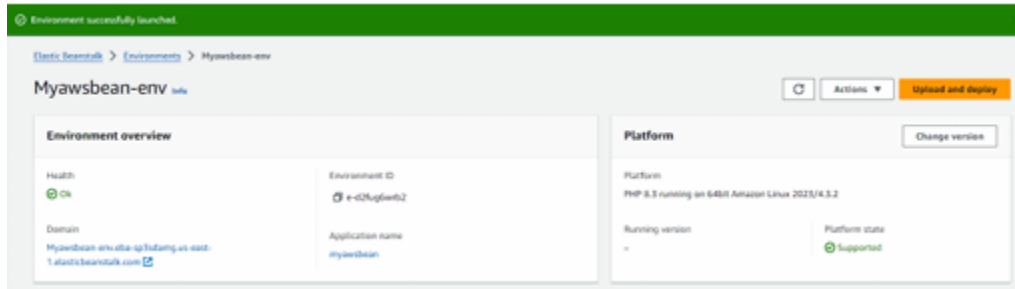
select the newly created key pair and instance profile. Now let everything be default.

The screenshot shows the 'Configure service access' step in the AWS Elastic Beanstalk console. On the left, a sidebar lists steps: 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), and Step 6 (Review). The main content area is titled 'Configure service access' and includes an 'Info' link. It contains three sections: 'Service access' with radio buttons for 'Create and use new service role' (selected) and 'Use an existing service role'; 'Service role name' with a text input field containing 'aws-elasticbeanstalk-service-role' and a 'View permission details' button; 'EC2 key pair' with a dropdown menu showing 'new-key' and a refresh button; and 'EC2 instance profile' with a dropdown menu showing 'new-user' and a refresh button. At the bottom are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

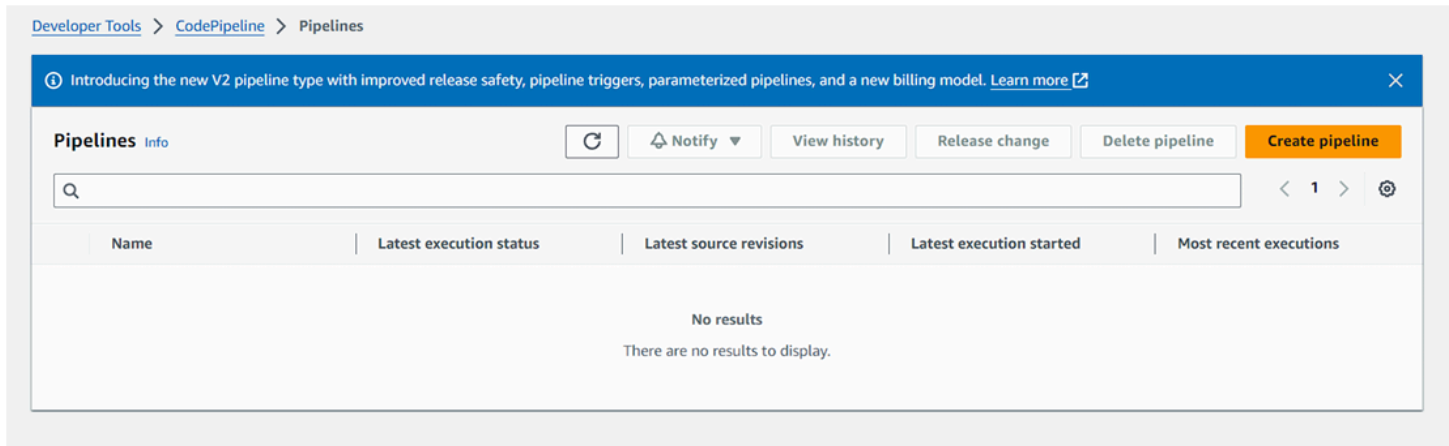
13) Review the changes and click on Create.

The screenshot shows the 'Review' step in the AWS Elastic Beanstalk console. It displays a summary of the configuration across three steps. Step 1: 'Configure environment' shows environment details like 'Environment tier: Web server environment', 'Application name: myjavabean', 'Environment name: Myjavabean-env', 'Application code: Sample application', and 'Platform: aws-elasticbeanstalk-us-east-1:platform/PHP 8.3 running on 64bit Amazon Linux 2021/4.3.2'. Step 2: 'Configure service access' shows 'Service role: arn:aws:iam::017820672175:role/ber-vice-role/aws-elasticbeanstalk-service-role', 'EC2 key pair: the\_key', and 'EC2 instance profile: new-user'. Step 3: 'Set up networking, database, and tags' is partially visible. Each step has an 'Edit' button.

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

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.



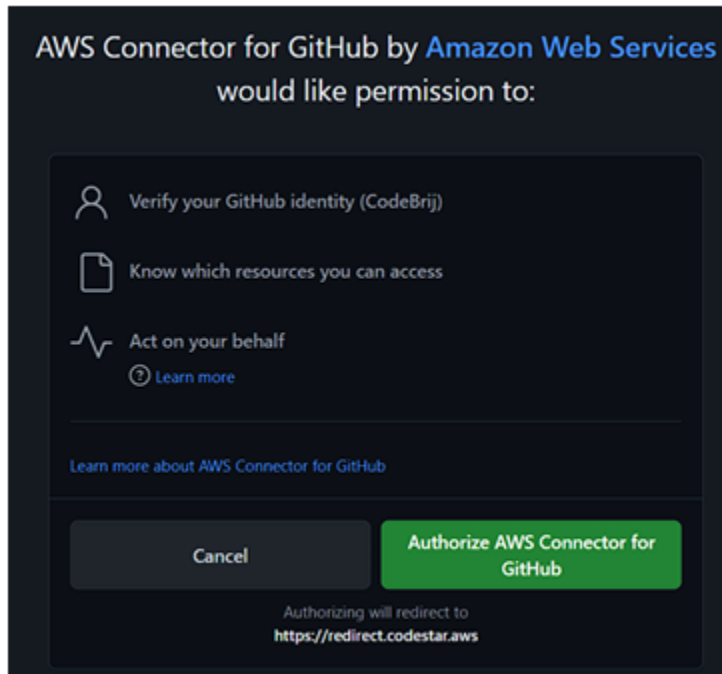
**Sign in to GitHub**  
to continue to **AWS Connector for GitHub**

Username or email address

Password [Forgot password?](#)

[Sign in](#)

19) Authorize AWS Connector for GitHub.



20) We need to install the GitHub connector.

Developer Tools > Connections > Create connection

Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. [Learn more](#)

### Connect to GitHub

**GitHub connection settings** [Info](#)

Connection name

App installation - optional

Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.

or [Install a new app](#)

Tags - optional

[Connect](#)

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 amaws: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-c3-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.

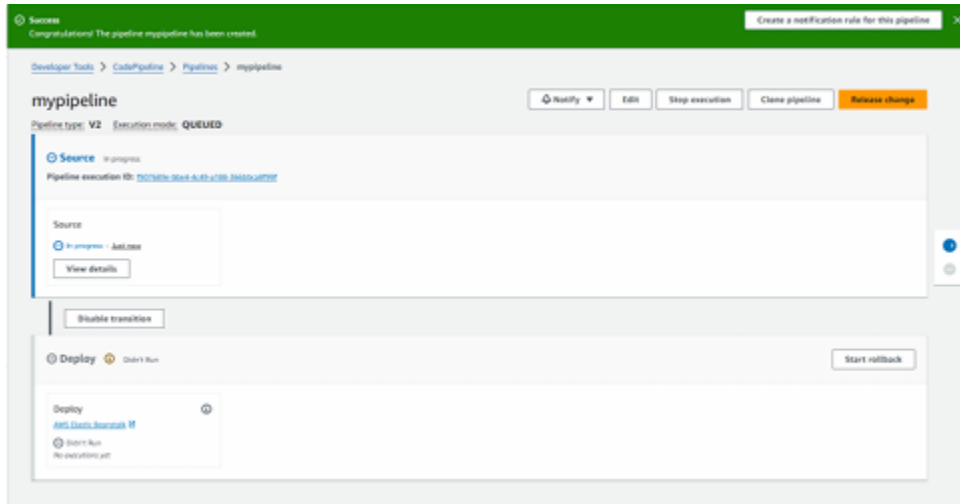
☒ **No filter**  
Starts your pipeline on any push and clones the HEAD.

☐ **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.



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

### 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.

myawsbean

**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.

Myawsbean-env

☐ Configure automatic rollback on stage failure

25) Go to Elastic Beanstalk and from D0main open the hosted site.

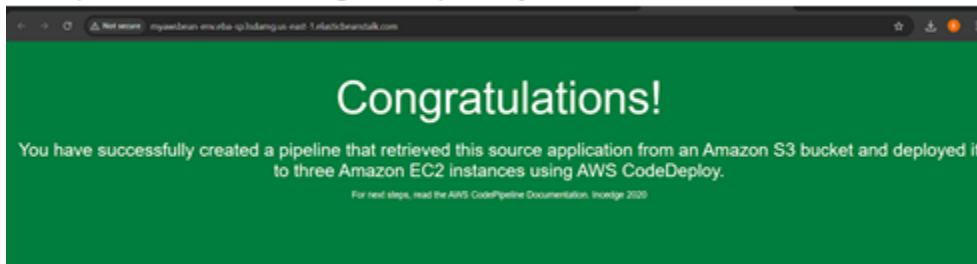
Elastic Beanstalk > Applications > myawsbean

Application myawsbean environments (1) info

Filter environments

Environment name	Health	Date created	Domain	Running versions	Platform	Platform state
Myawsbean-env	Ok	August 15, 2024 22:12:50 ...	Myawsbean-env.elfe-qg3tdam...	code-pipeline-172374181...	PHP 8.3 running on 64bit ...	Supported

26) Hosted site from the github repository.



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

