St. Francis Institute of Technology, Mumbai-400 103

Department of Information Technology

A.Y. 2023-2024

Class: TE-ITA/B, Semester: V

Subject: **Advanced DevOps Lab**

Experiment – 4: 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. **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.

2. **Objectives:** After study of this experiment, the student will be able to
   - Build an application using AWS CodeBuild and Deploy on S3 using CodePipeline.
   - Deploy an application on EC2 using AWS CodeDeploy.

3. **Outcomes:** After study of this experiment, the student will be able to
   - Build an application using AWS CodeBuild and Deploy on S3 using CodePipeline.
   - Deploy an application on EC2 using AWS CodeDeploy.

4. **Prerequisite:** Fundamentals of cloud computing

5. **Requirements:** PC and Internet

6. **Pre-Experiment Exercise:**

Brief Theory:

AWS CodePipeline:

AWS CodePipeline is a continuous delivery service you can use to model, visualize, and automate the steps required to release your software. You can quickly model and configure the different stages of a software release process. CodePipeline automates the steps required to release your software changes continuously.

CICD:

codePipeline is a *continuous delivery* service that automates the building, testing, and deployment of your software into production.

Continuous delivery is a software development methodology where the release process is automated. Every software change is automatically built, tested, and deployed to production. Before the final push to production, a person, an automated test, or a business rule decides when the final push should occur. Although every successful software change can be immediately released to production with continuous delivery, not all changes need to be released right away.

Continuous integration is a software development practice where members of a team use a version control system and frequently integrate their work to the same location, such as a main

branch. Each change is built and verified to detect integration errors as quickly as possible. Continuous integration is focused on automatically building and testing code, as compared to *continuous delivery*, which automates the entire software release process up to production.

Features:

You can use CodePipeline to help you automatically build, test, and deploy your applications in the cloud. Specifically, you can:

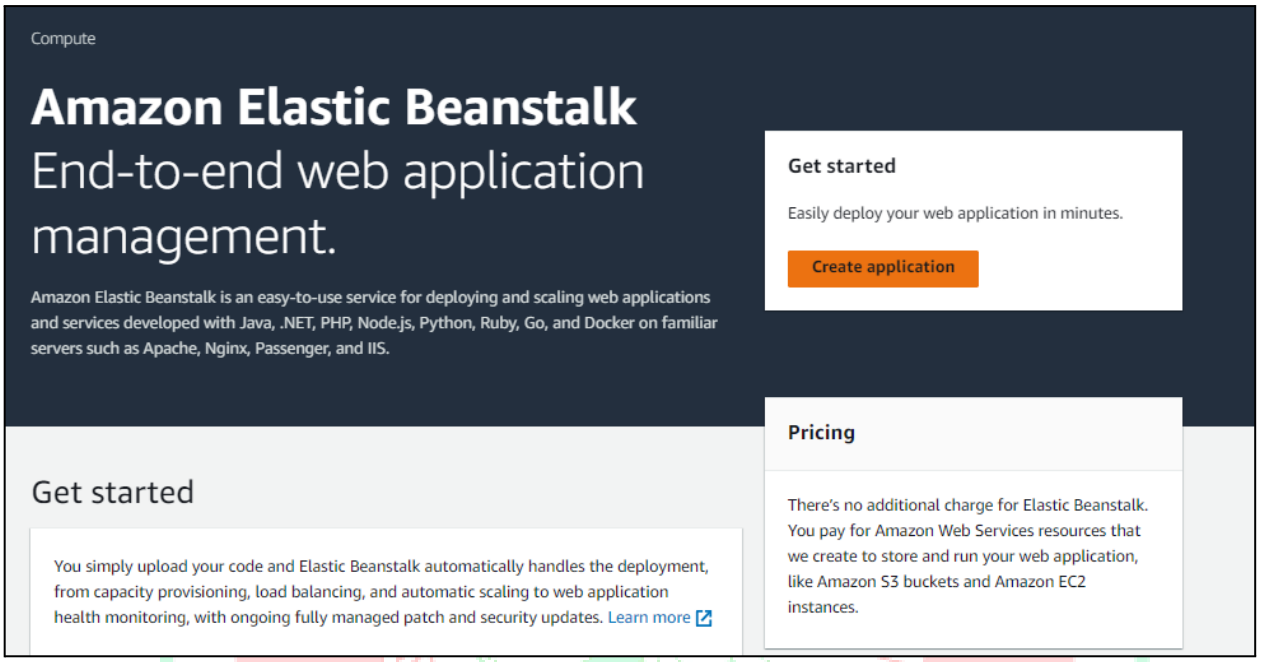
1. **Automate your release processes**: CodePipeline fully automates your release process from end to end, starting from your source repository through build, test, and deployment. You can prevent changes from moving through a pipeline by including a manual approval action in any stage except a Source stage. You can release when you want, in the way you want, on the systems of your choice, across one instance or multiple instances.
2. **Establish a consistent release process**: Define a consistent set of steps for every code change. CodePipeline runs each stage of your release according to your criteria. 3. **Speed up delivery while improving quality**: You can automate your release process to allow your developers to test and release code incrementally and speed up the release of new features to your customers.
4. **Use your favorite tools**: You can incorporate your existing source, build, and deployment tools into your pipeline. For a full list of AWS services and third-party tools currently supported by CodePipeline..
5. **View progress at a glance**: You can review real-time status of your pipelines, check the details of any alerts, retry failed actions, view details about the source revisions used in the latest pipeline execution in each stage, and manually rerun any pipeline.
6. **View pipeline history details**: You can view details about executions of a pipeline, including start and end times, run duration, and execution IDs.

You will create the pipeline using AWS CodePipeline, a service that builds, tests, and deploys your code every time there is a code change. You will use your GitHub account, an Amazon Simple Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the sample app. Your completed pipeline will be able to detect changes made to the source repository containing the sample app and then automatically update your live sample app.

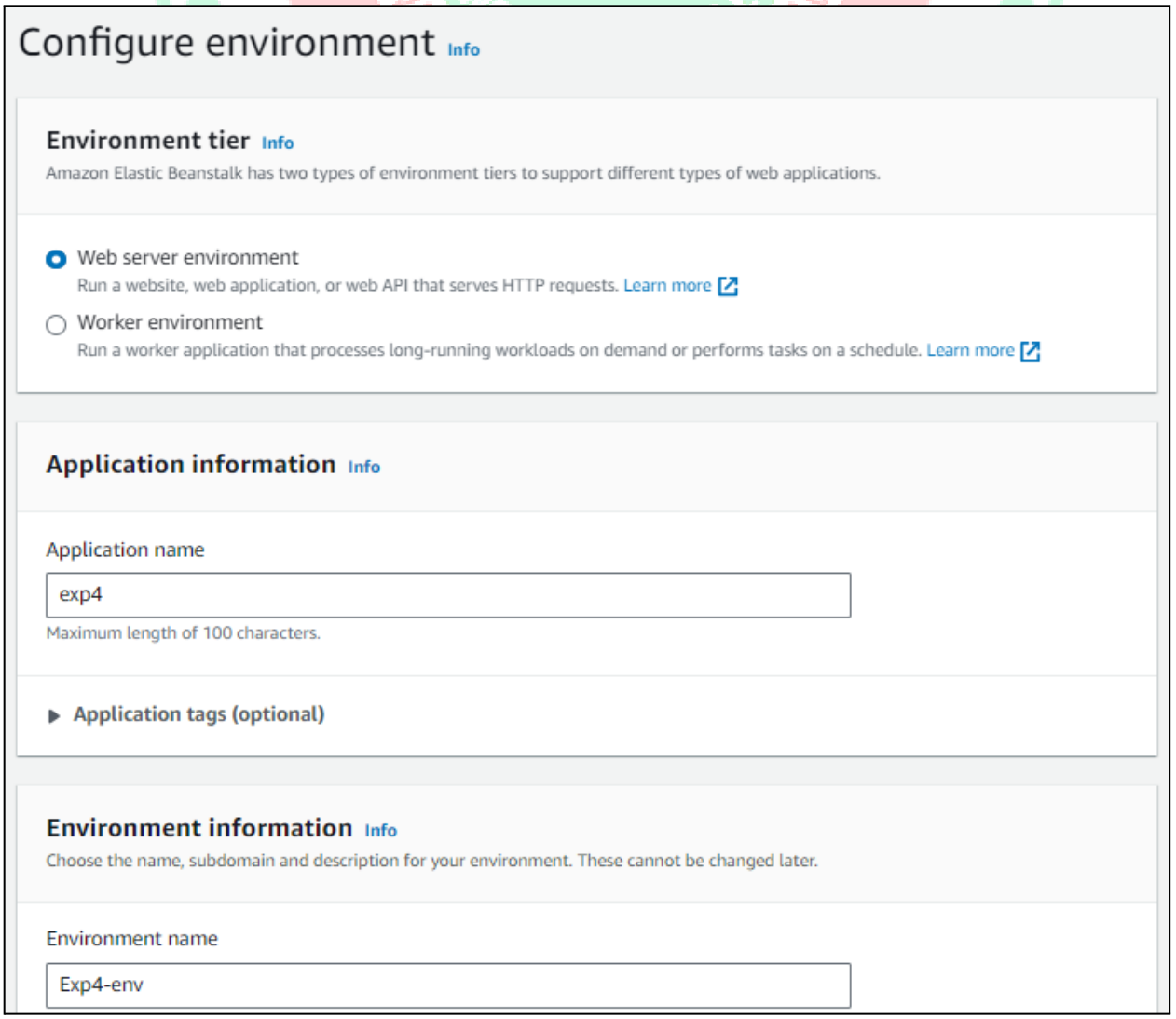
7. Laboratory Exercise

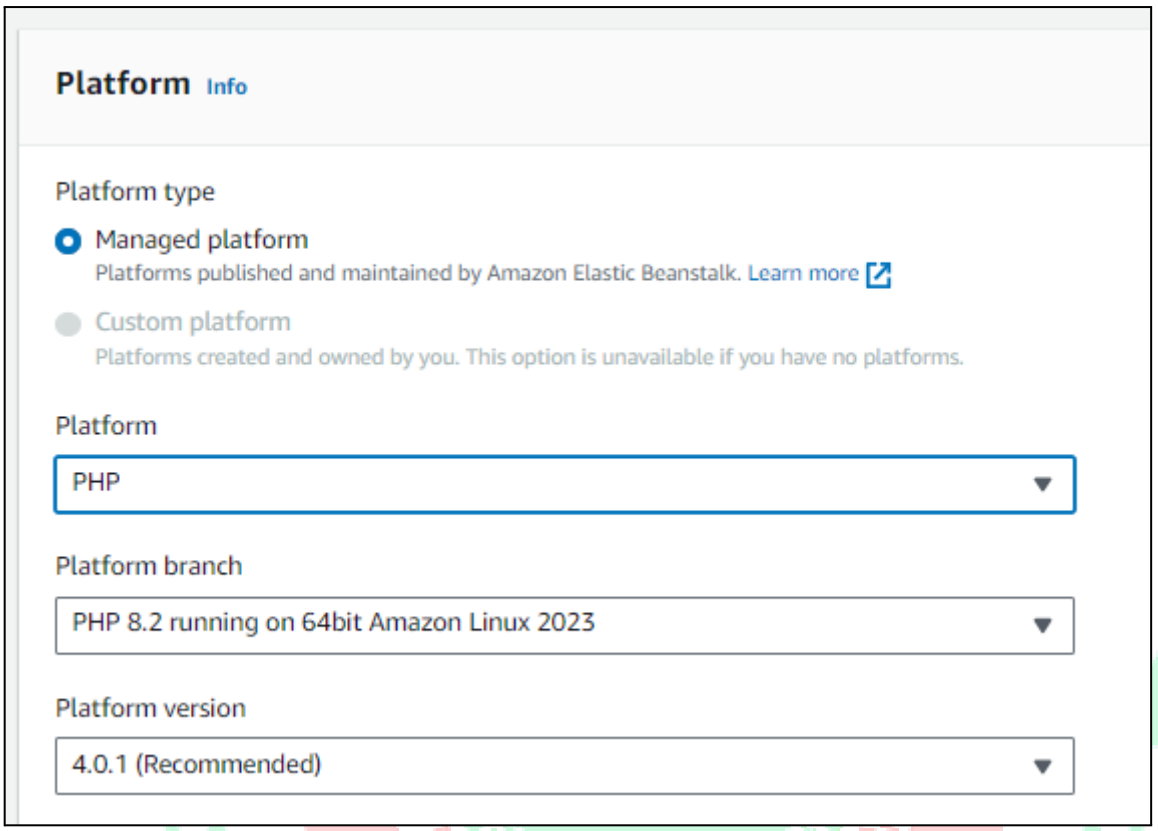
1) To simplify the process of setting up and configuring EC2 instances, you will spin up a sample environment using AWS Elastic Beanstalk. Elastic Beanstalk lets you easily host web applications without needing to launch, configure, or operate virtual servers on your own. It automatically provisions and operates the infrastructure (e.g. virtual servers, load balancers, etc.) and provides the application stack (e.g. OS, language and framework, web and application server, etc.) for you. Your continuous deployment pipeline will need a target environment containing virtual servers, or Amazon EC2 instances, where it will deploy sample code. You will prepare this environment before
creating the pipeline.

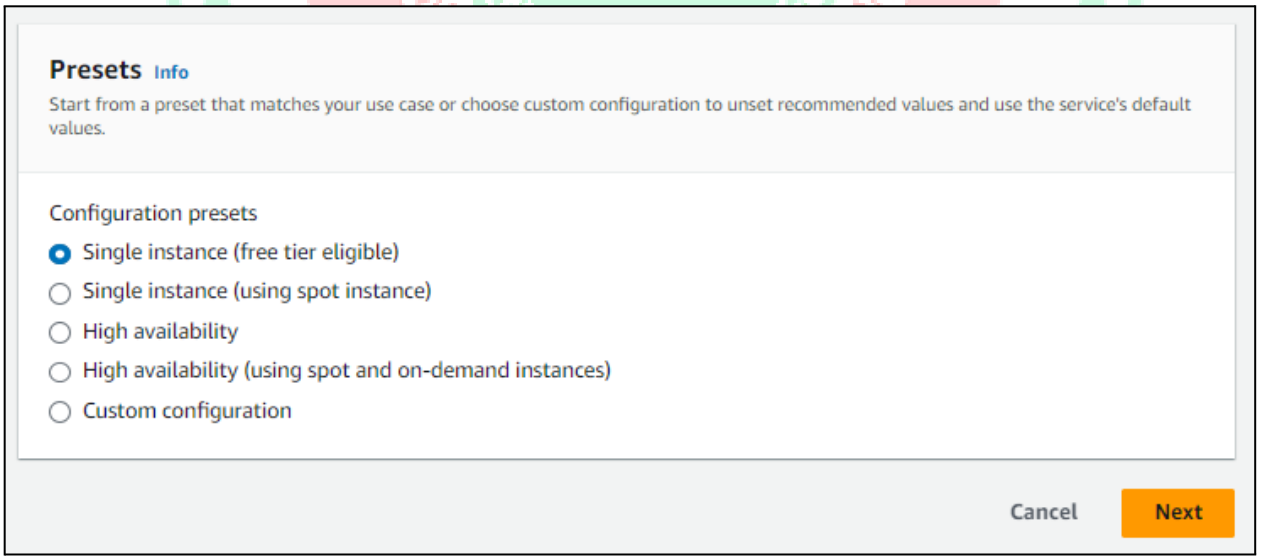
Step 1: Create a deployment environment



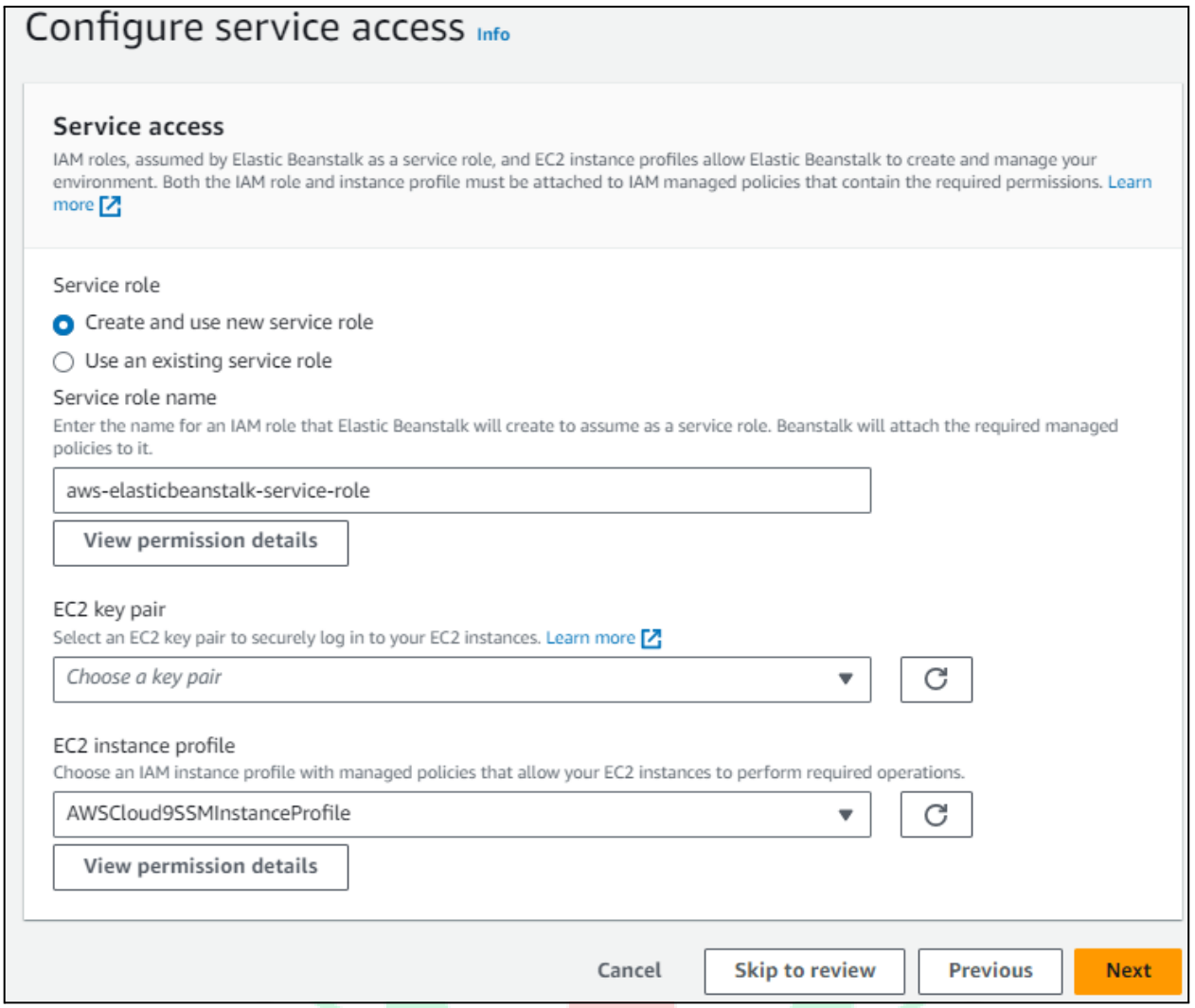
Configure environment

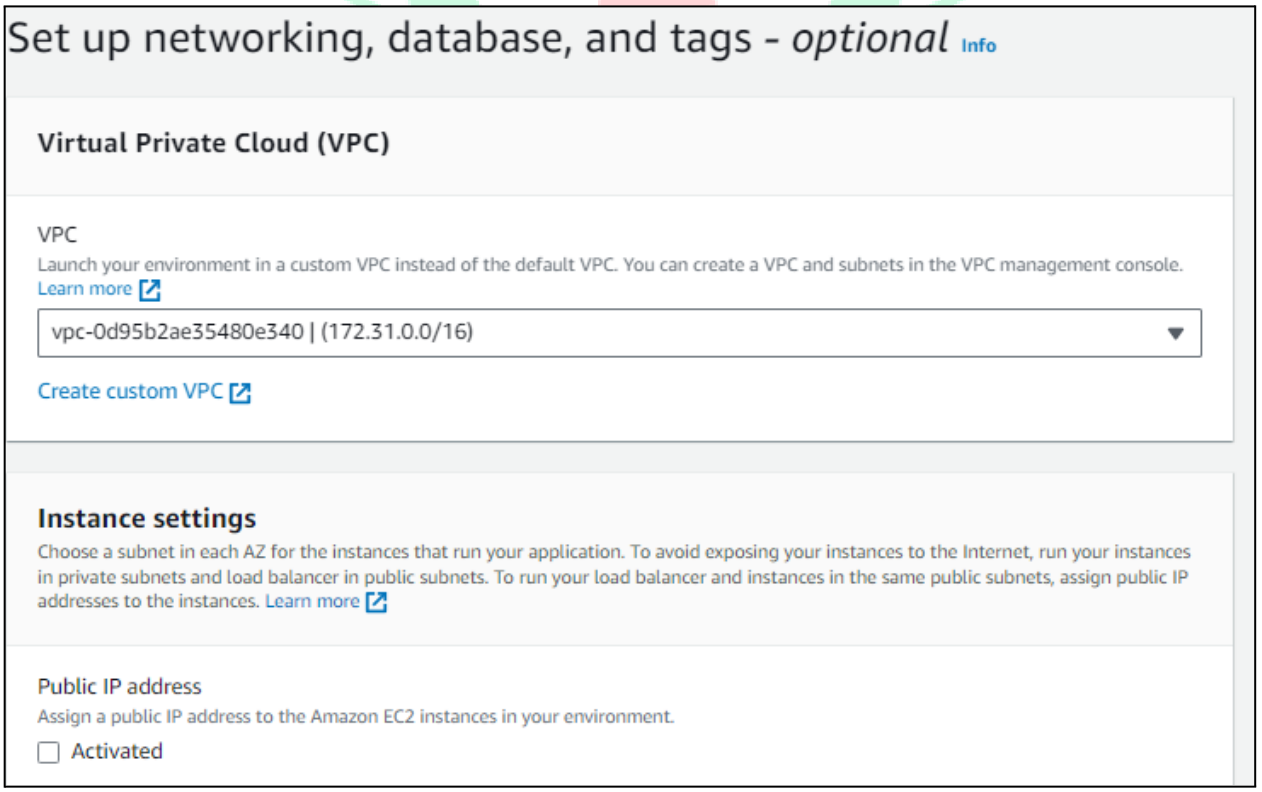


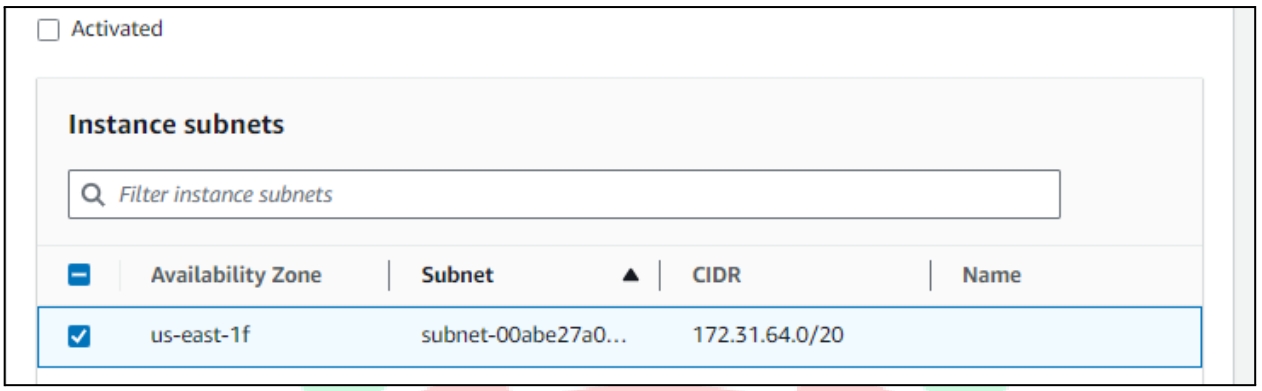

Platform Info
Platform type
Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more
Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.
Platform
PHP
Platform branch
PHP 8.2 running on 64bit Amazon Linux 2023
Platform version
4.0.1 (Recommended)

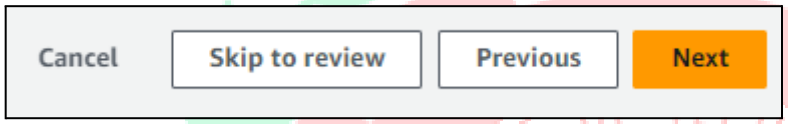

Presets Info
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
Configuration presets
Single instance (free tier eligible)
Single instance (using spot instance)
High availability
High availability (using spot and on-demand instances)
Custom configuration
Cancel
Next

Configure service access

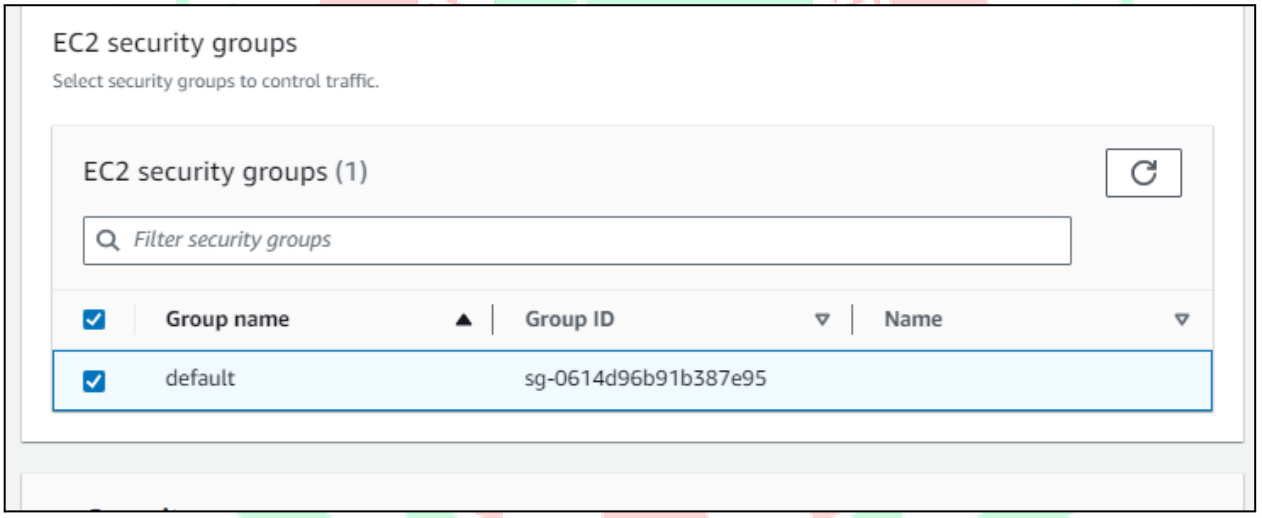


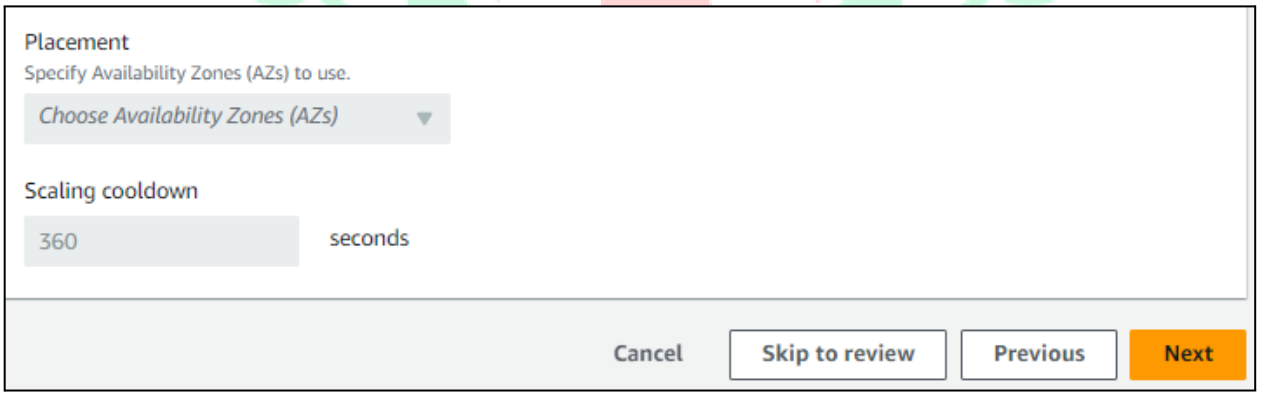


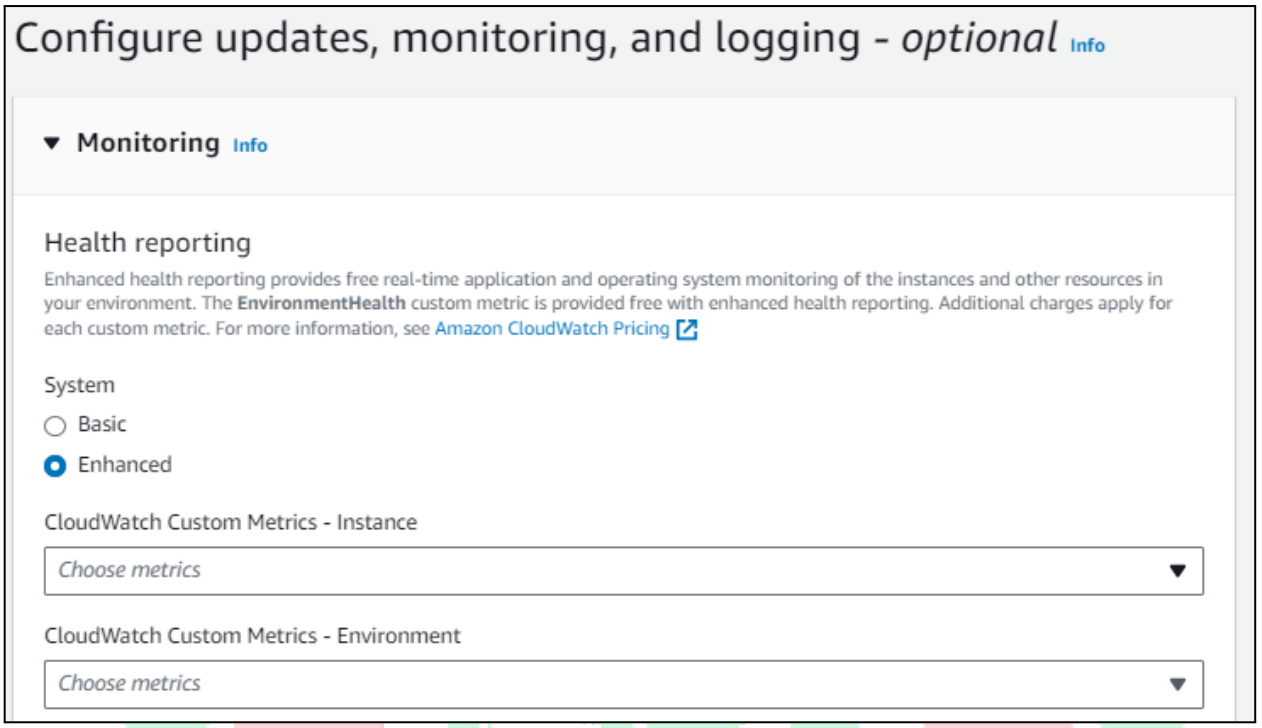


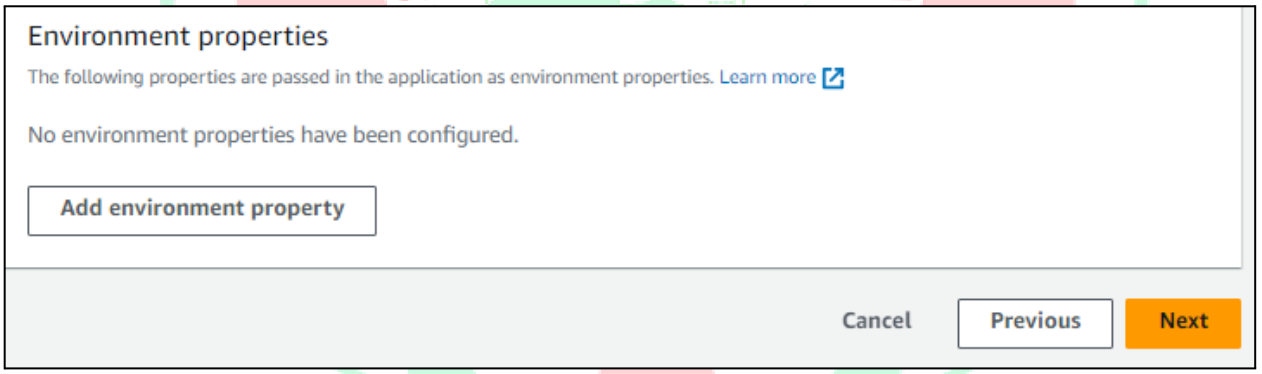


Configure instance traffic and scaling

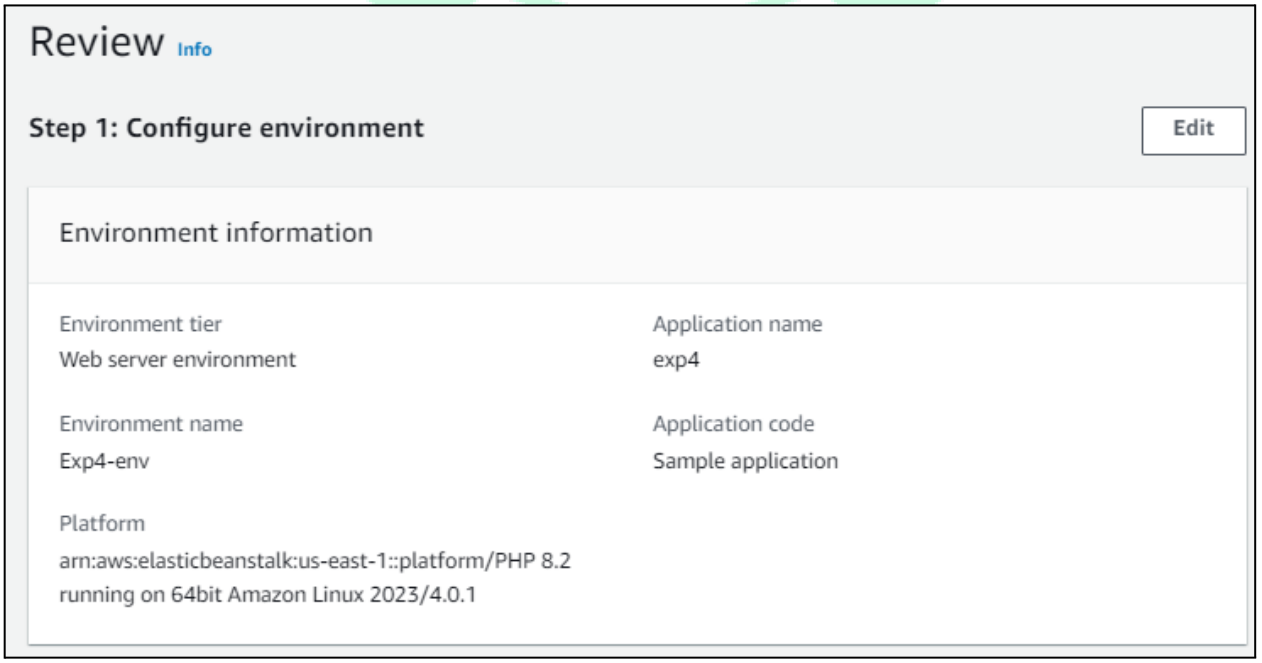


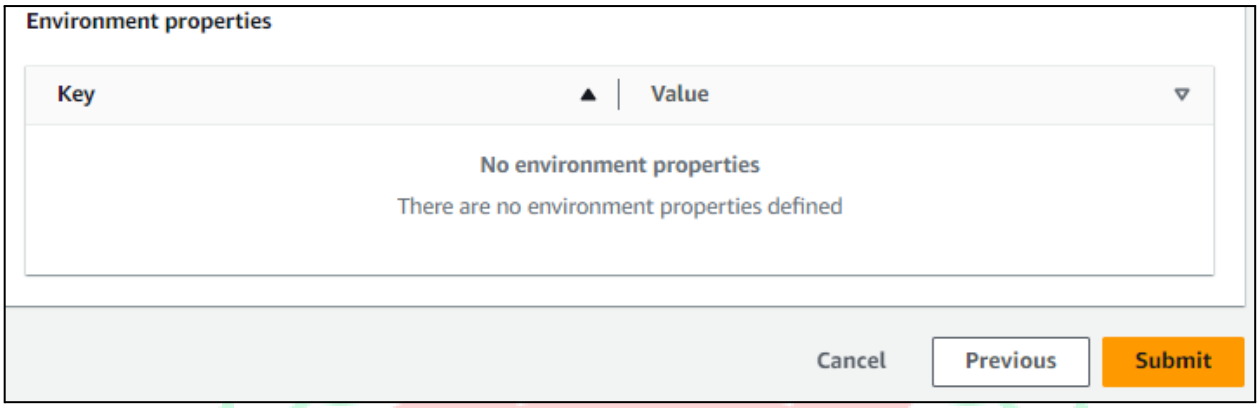


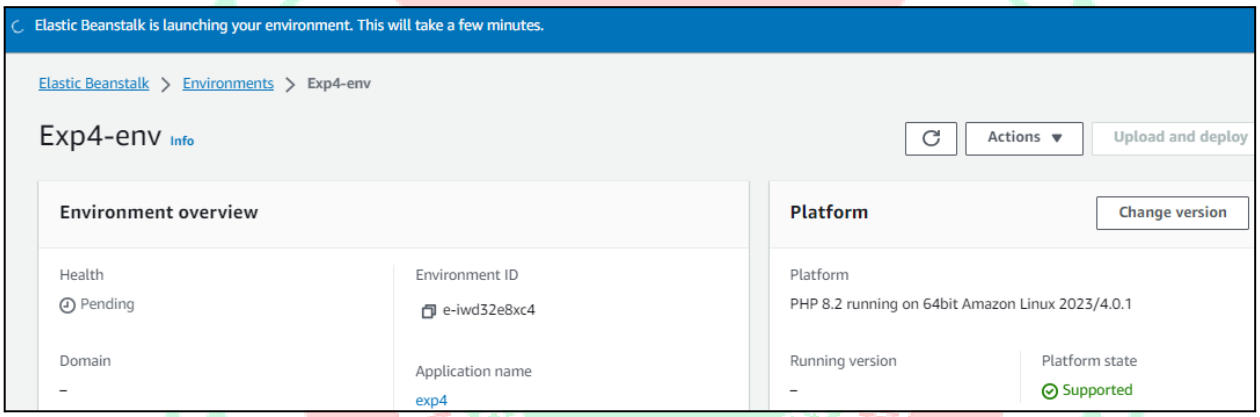

Configure updates, monitoring, and logging - optional Info
Monitoring Info
Health reporting
Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The EnvironmentHealth custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see Amazon CloudWatch Pricing
System
Basic
Enhanced
CloudWatch Custom Metrics - Instance
Choose metrics
CloudWatch Custom Metrics - Environment
Choose metrics


Environment properties
The following properties are passed in the application as environment properties. Learn more
No environment properties have been configured.
Add environment property
Cancel
Previous
Next

Review


Review Info
Step 1: Configure environment
Edit
Environment information
Environment tier
Web server environment
Application name
exp4
Environment name
Exp4-env
Application code
Sample application
Platform
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.2 running on 64bit Amazon Linux 2023/4.0.1



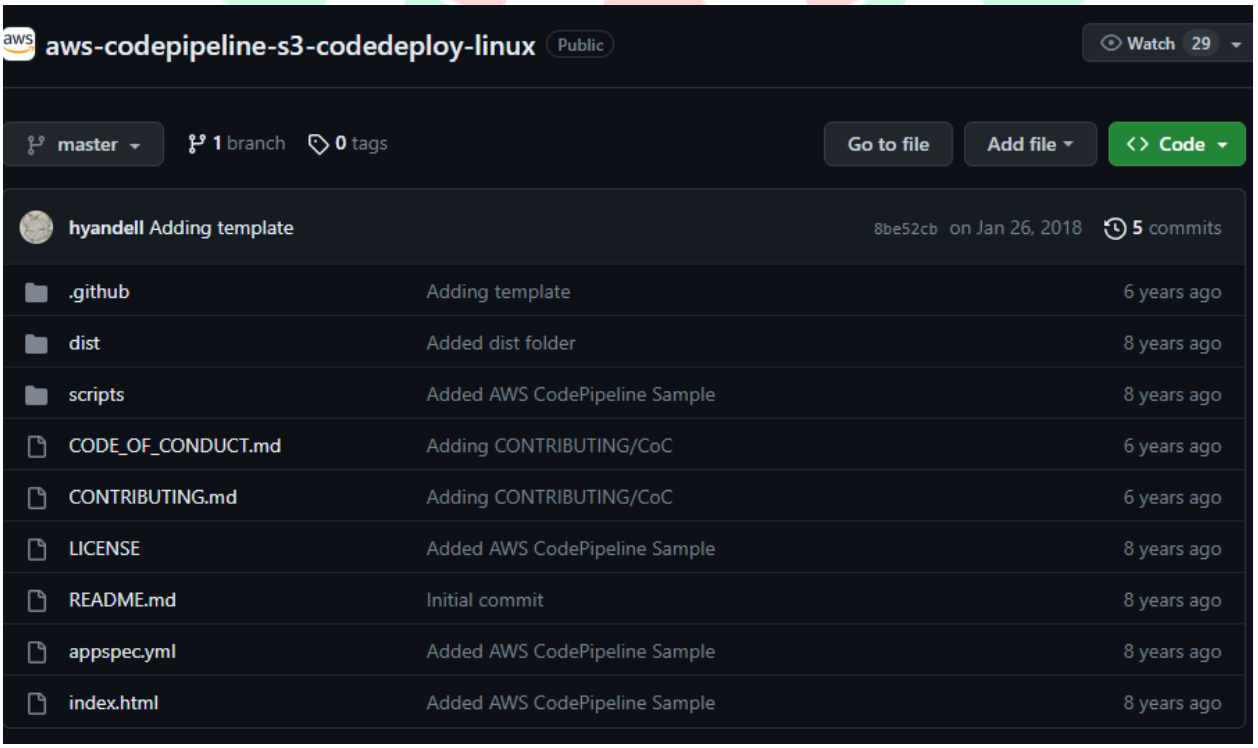


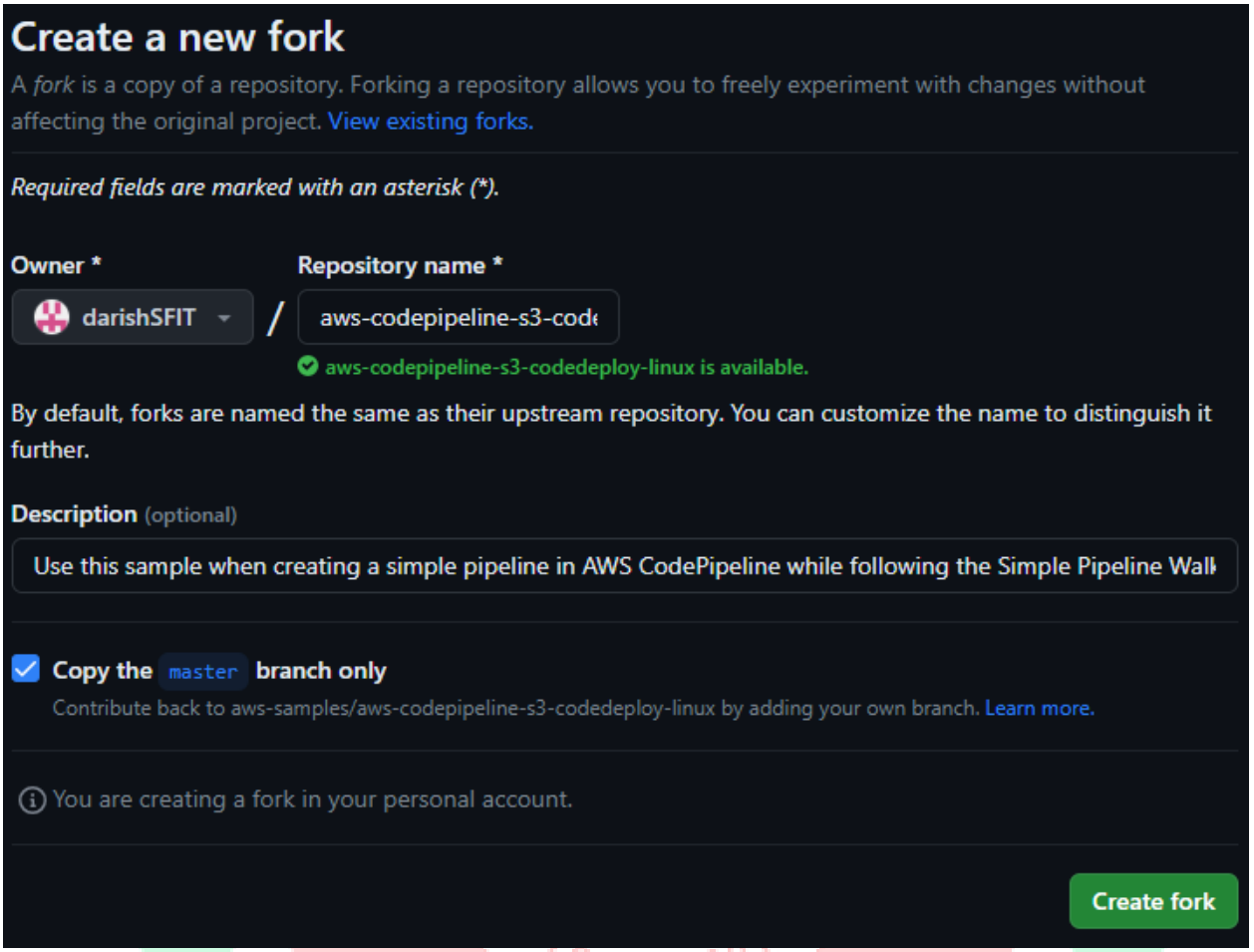
Step 2: Get the copy of Sample Code

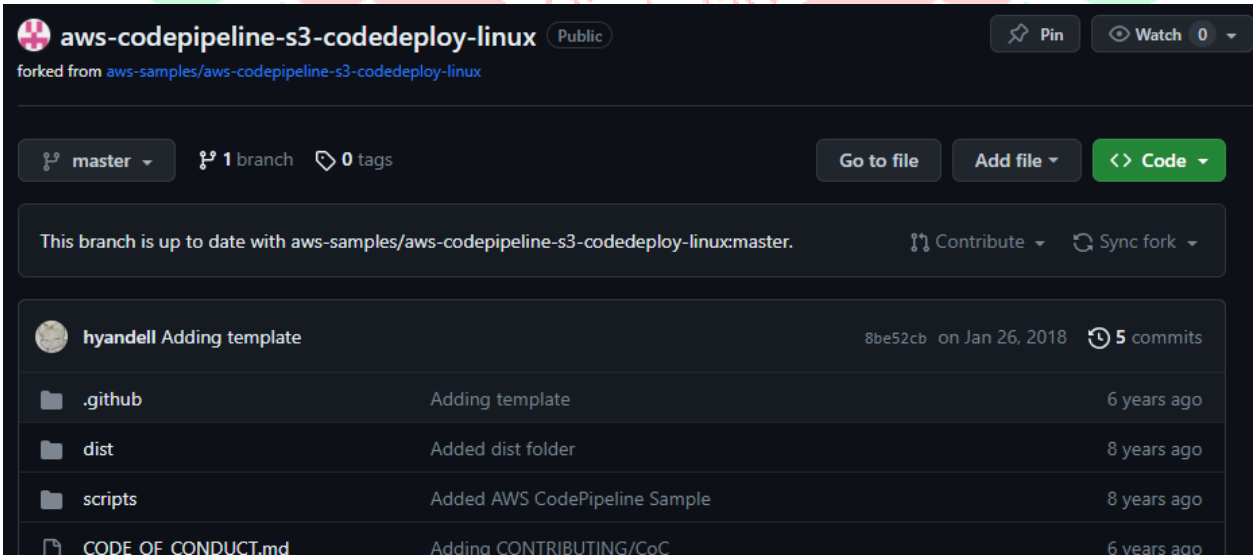
In this step you will retrieve the sample app's code and choose a source to host the code. Pipeline takes the code and performs actions on it.

You can use three options.

- GitHub Repository







- Amazon S3 Bucket
- AWS CodeCommit Repository

Open Amazon S3 console and Create your S3 Bucket:

Create Bucket

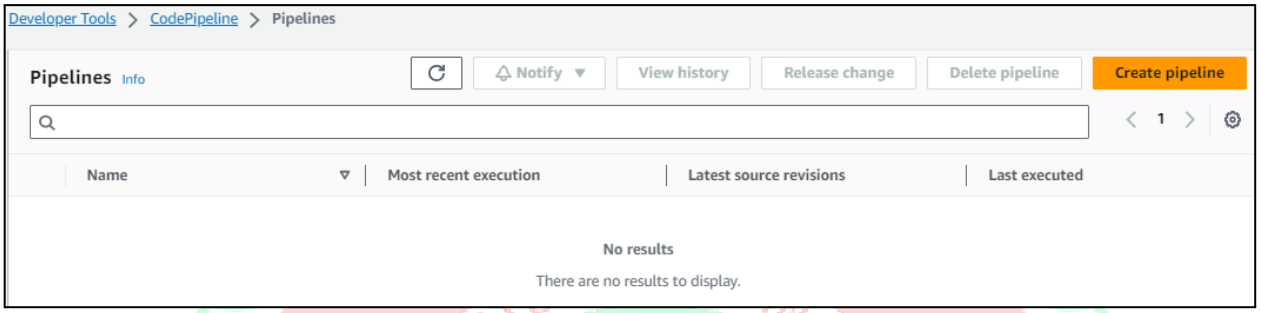
Upload the code to Bucket

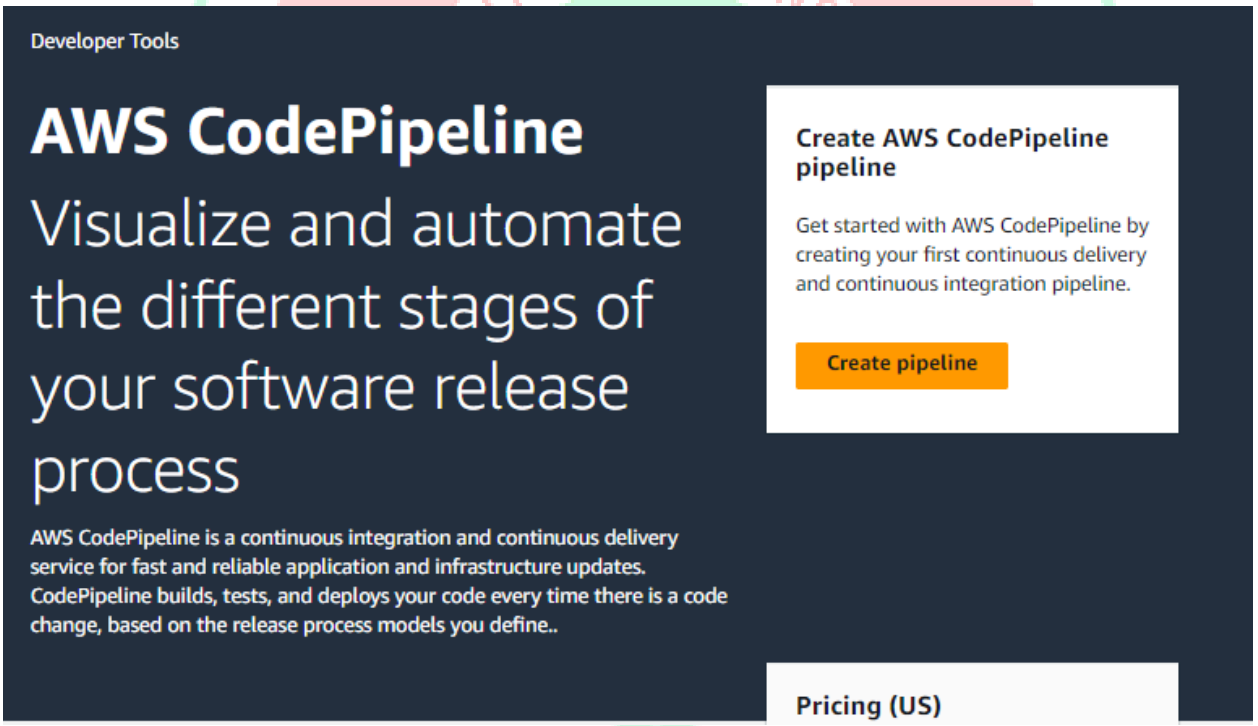
Select Bucket and Click on Upload (Right Corner)

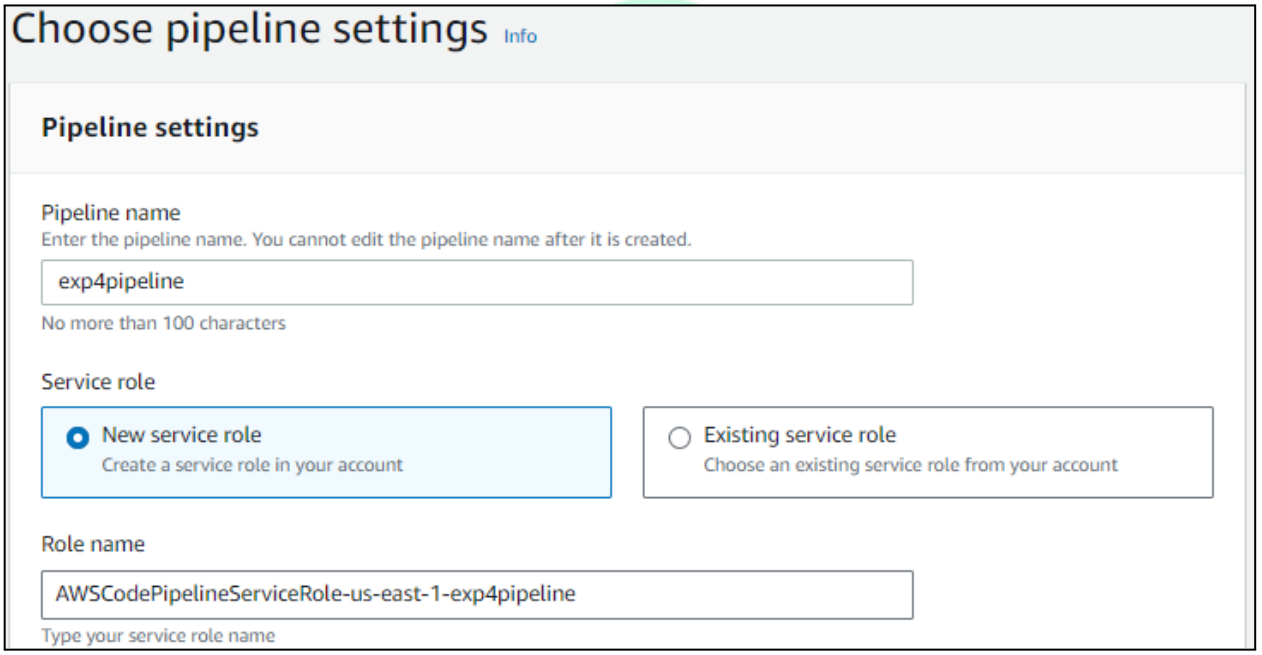
Add Files -- upload zip file from downloads of your computer. Click on Upload Button

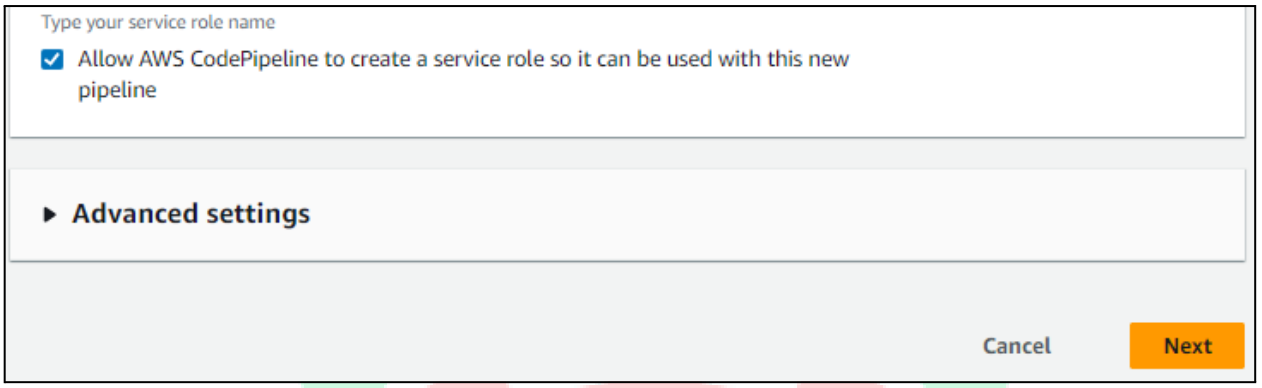
Step 3: Create Pipeline

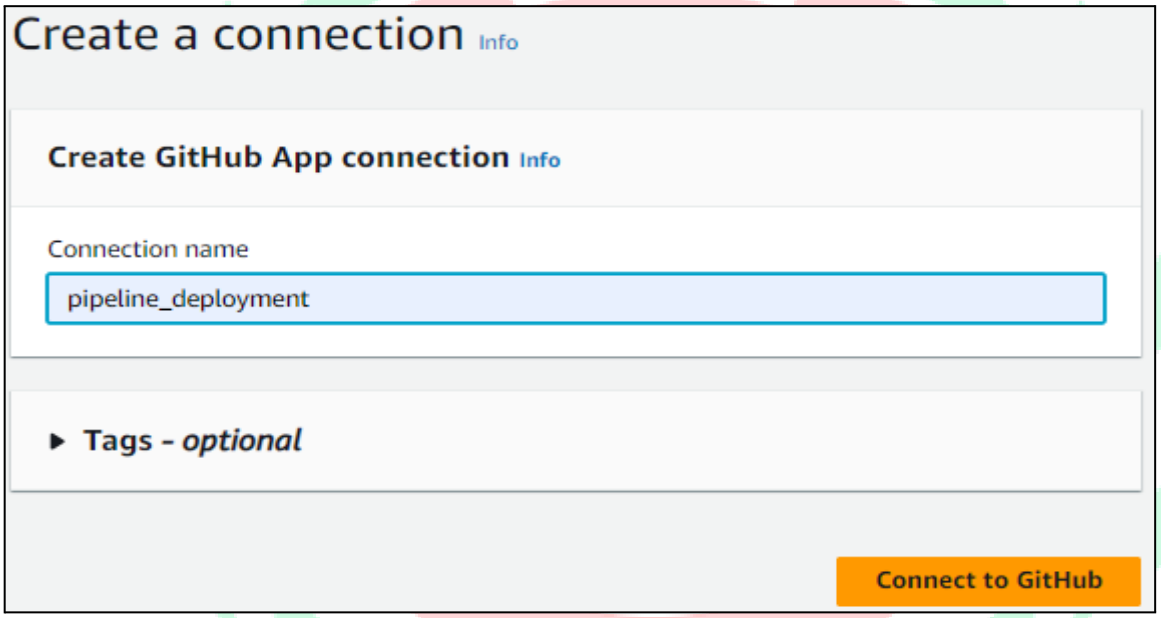
In this step, you will create and configure a simple pipeline with two actions: source and deploy. You will provide CodePipeline with the locations of your source repository and deployment environment.

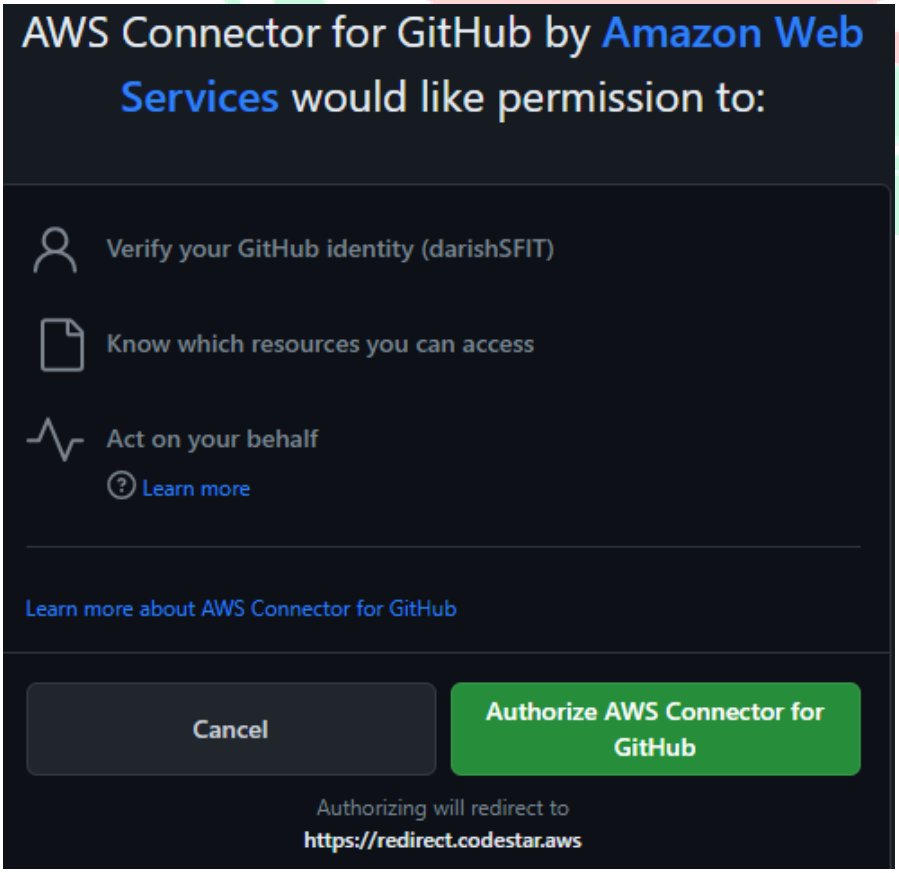


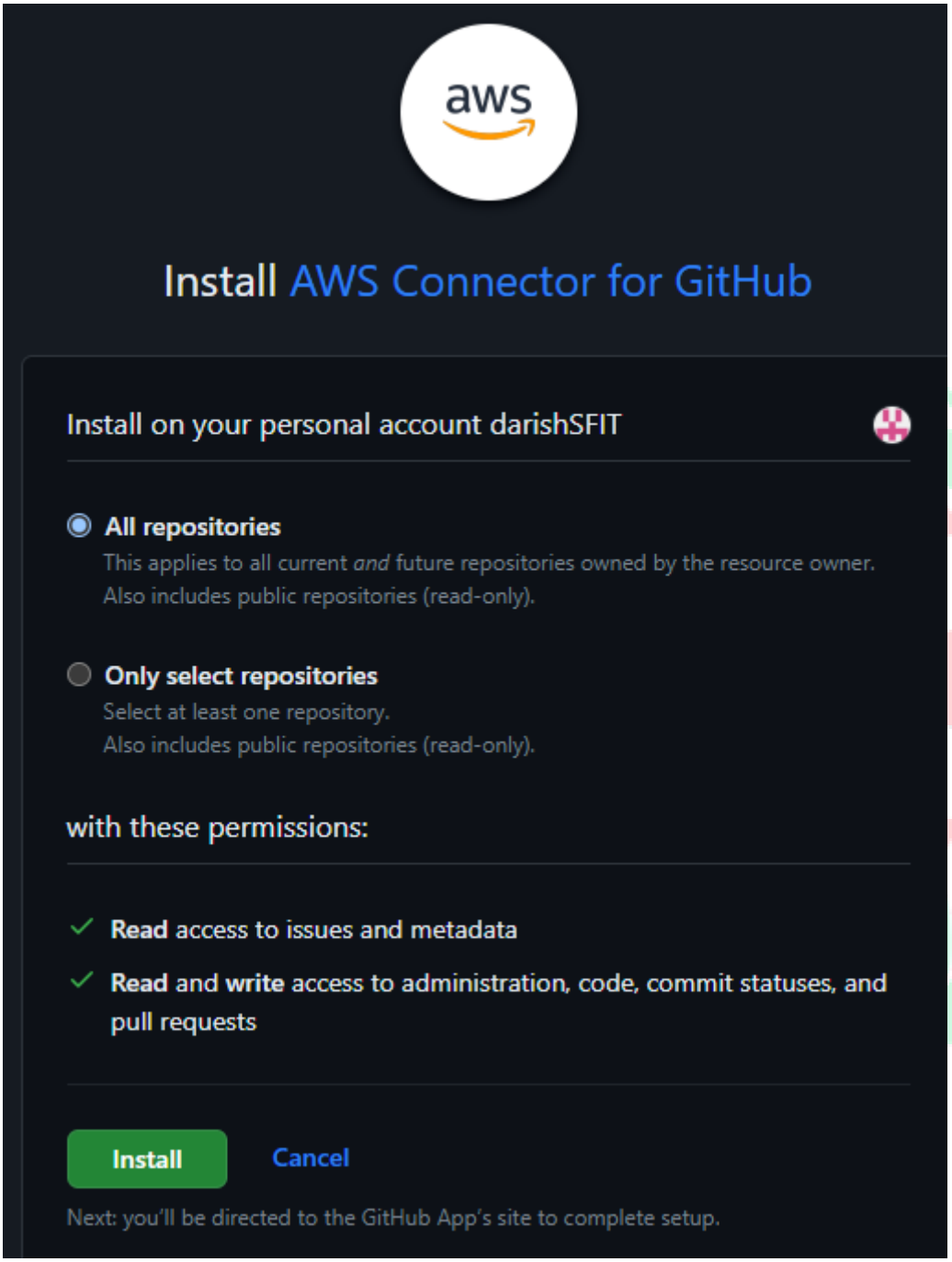


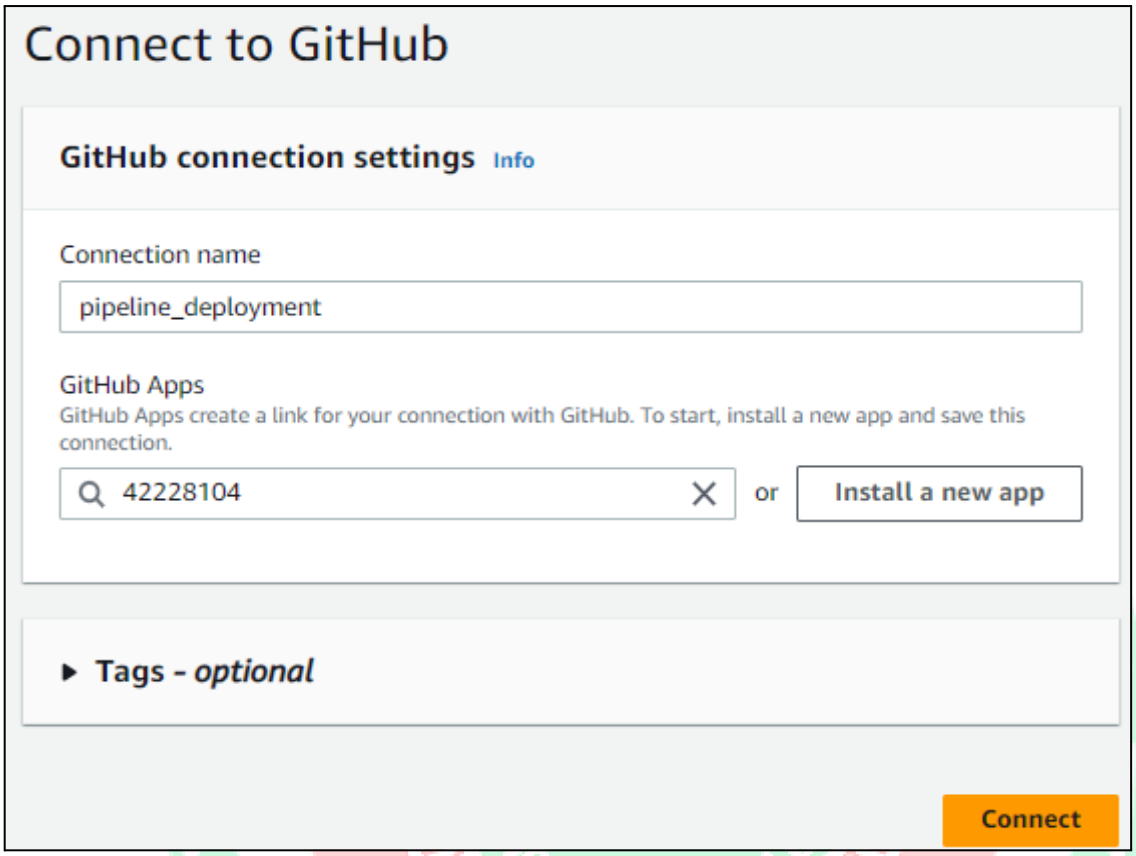


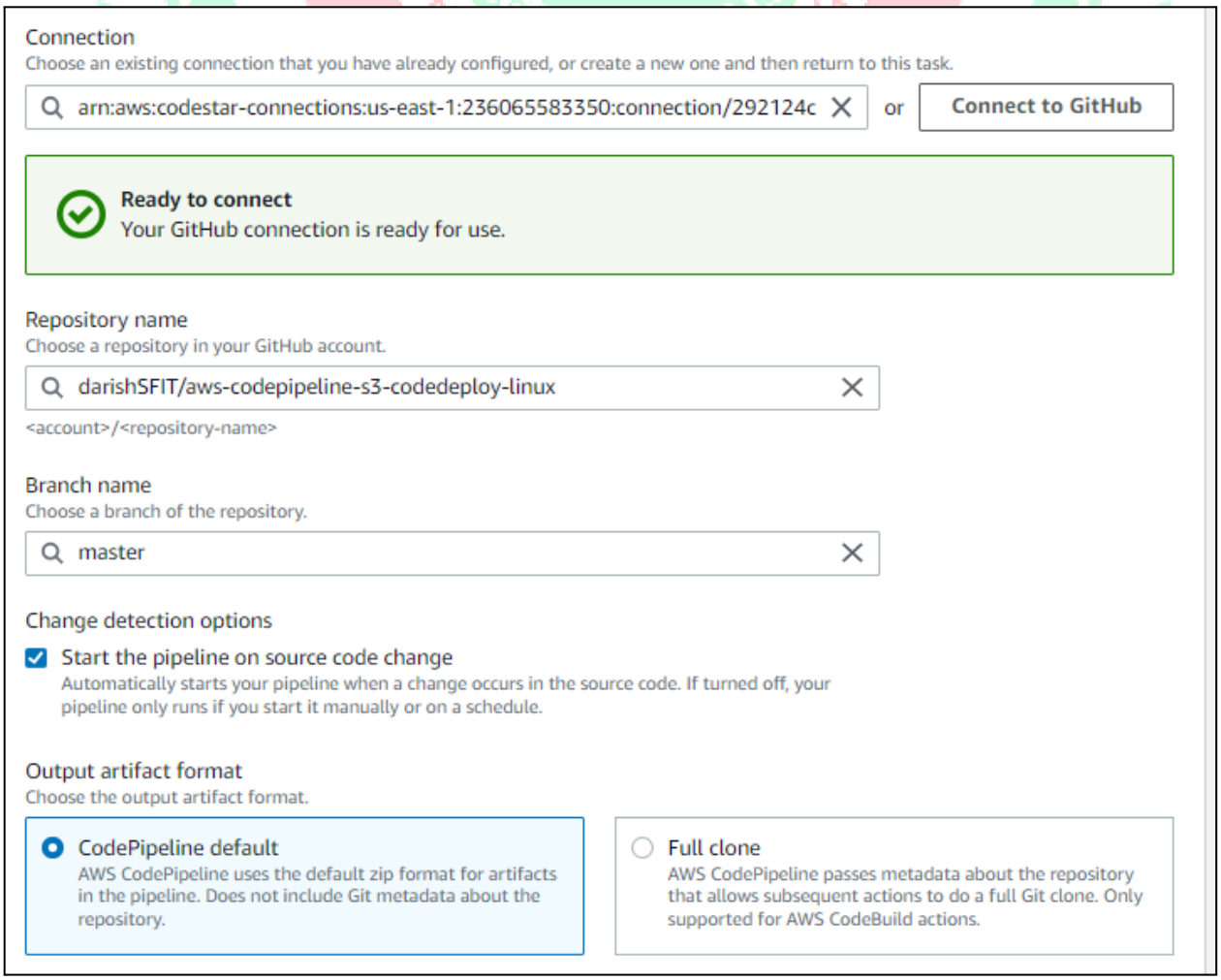

Type your service role name
Allow AWS CodePipeline to create a service role so it can be used with this new pipeline
Advanced settings
Cancel
Next


Create a connection Info
Create GitHub App connection Info
Connection name
pipeline_deployment
Tags - optional
Connect to GitHub

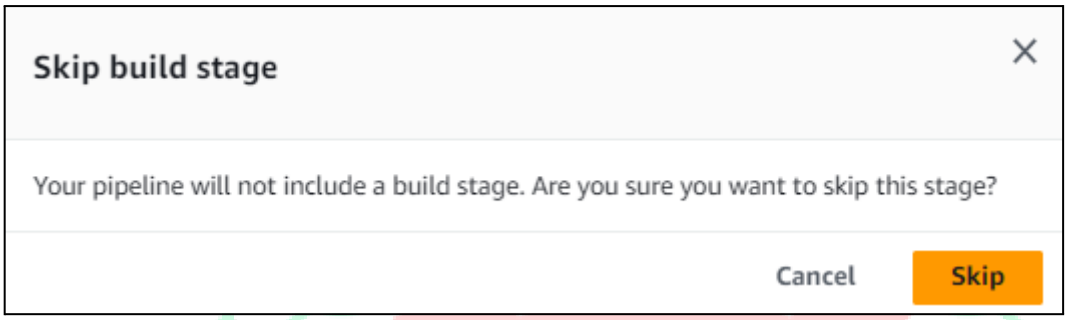

AWS Connector for GitHub by Amazon Web Services would like permission to:
Verify your GitHub identity (darishSFIT)
Know which resources you can access
Act on your behalf
Learn more
Learn more about AWS Connector for GitHub
Cancel
Authorize AWS Connector for GitHub
Authorizing will redirect to
https://redirect.codestar.aws

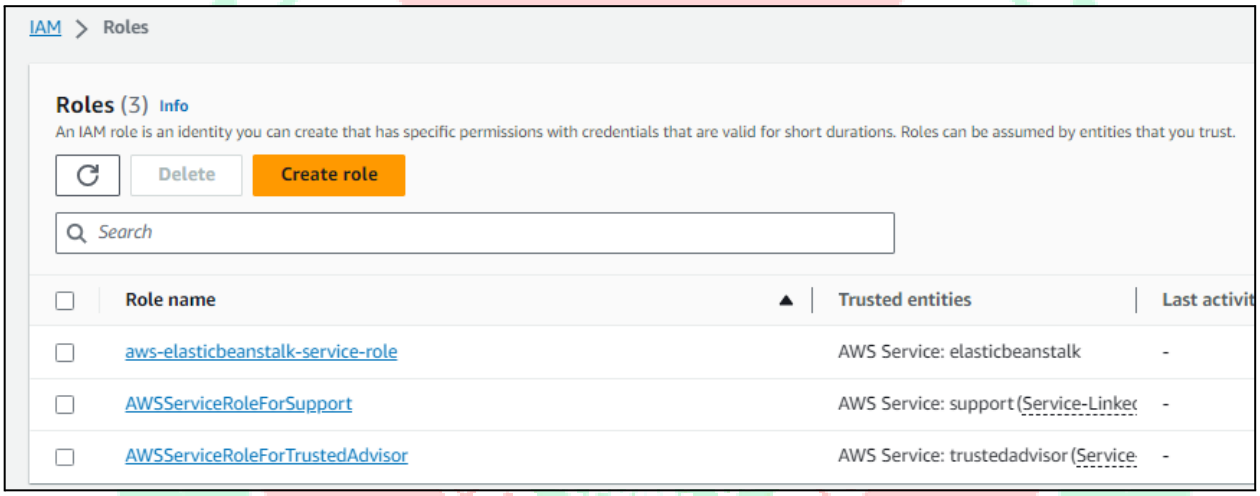

aws
Install AWS Connector for GitHub
Install on your personal account darishSFIT
All repositories
This applies to all current and future repositories owned by the resource owner.
Also includes public repositories (read-only).
Only select repositories
Select at least one repository.
Also includes public repositories (read-only).
with these permissions:
Read access to issues and metadata
Read and write access to administration, code, commit statuses, and pull requests
Install
Cancel
Next: you'll be directed to the GitHub App's site to complete setup.

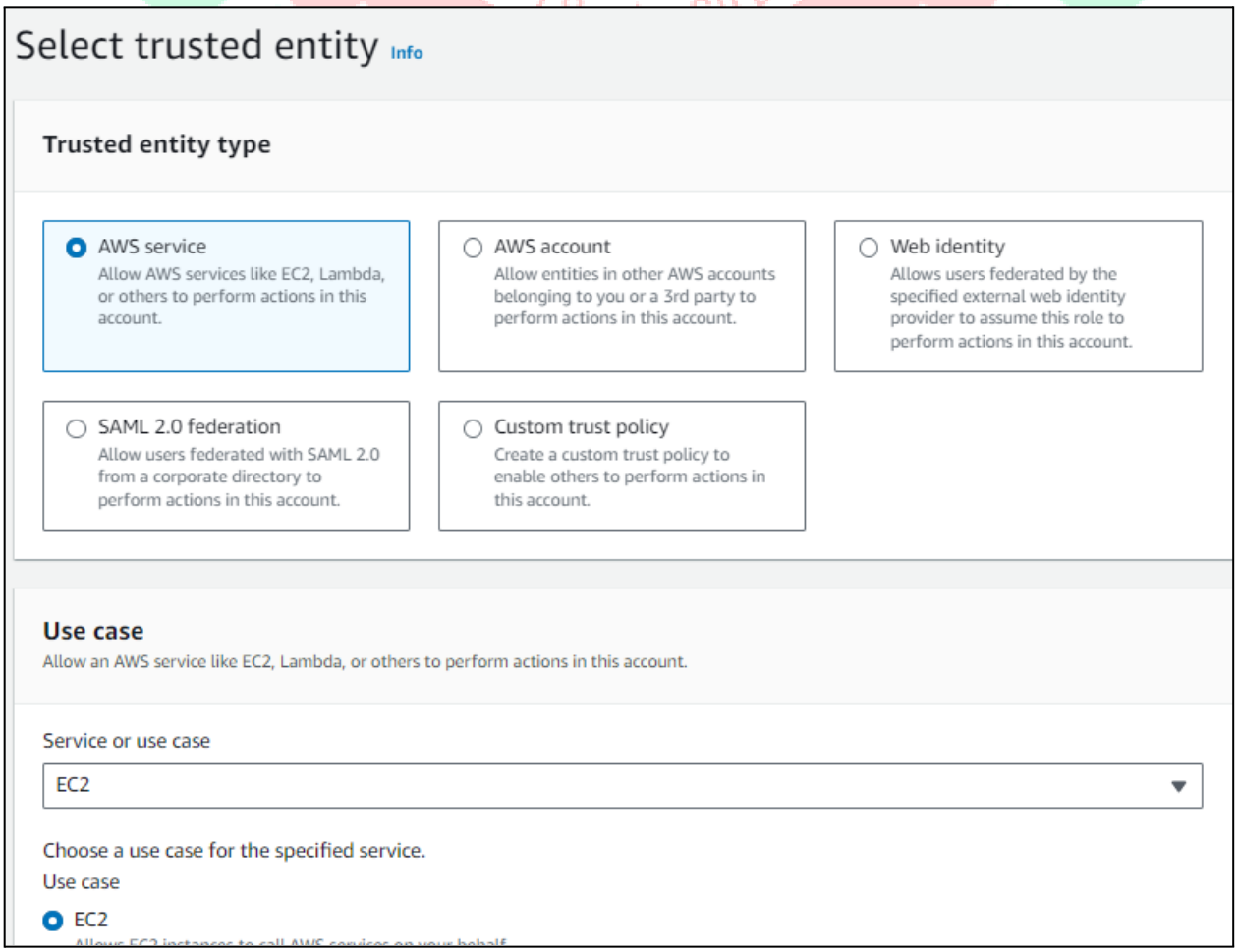

Connect to GitHub
GitHub connection settings Info
Connection name
pipeline_deployment
GitHub Apps
GitHub Apps create a link for your connection with GitHub. To start, install a new app and save this connection.
42228104
or
Install a new app
Tags - optional
Connect

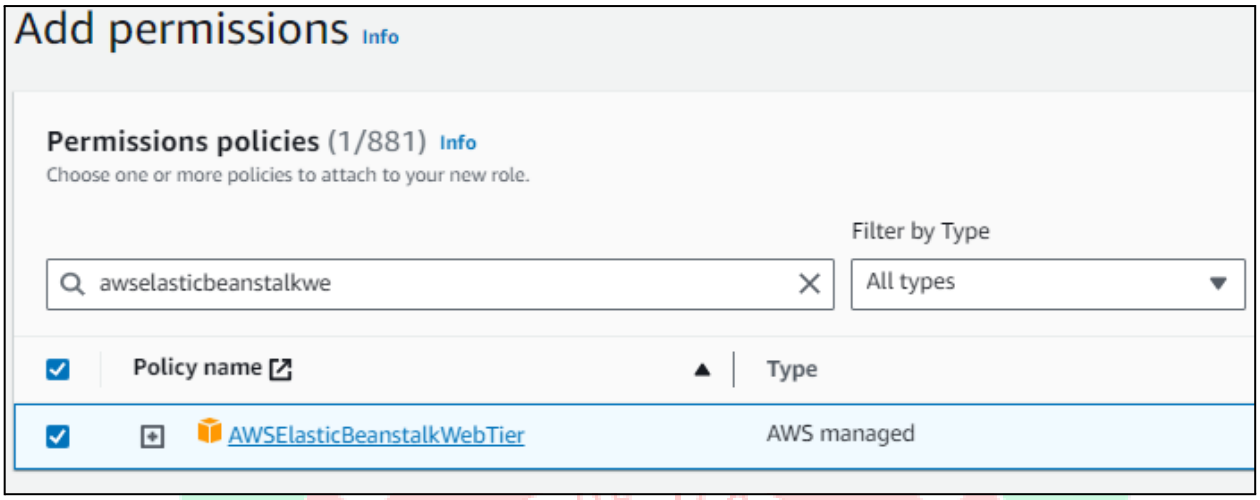

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.
arn:aws:codestar-connections:us-east-1:236065583350:connection/292124c
or
Connect to GitHub
Ready to connect
Your GitHub connection is ready for use.
Repository name
Choose a repository in your GitHub account.
darishSFIT/aws-codepipeline-s3-codedeploy-linux
<account>/<repository-name>
Branch name
Choose a branch of the repository.
master
Change detection options
Start the pipeline on source code change
Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.
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.

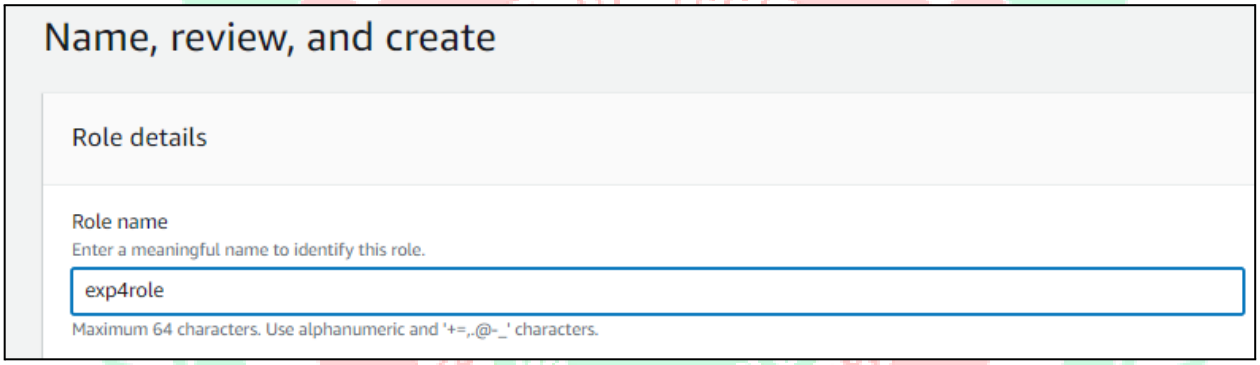

Cancel
Previous
Next

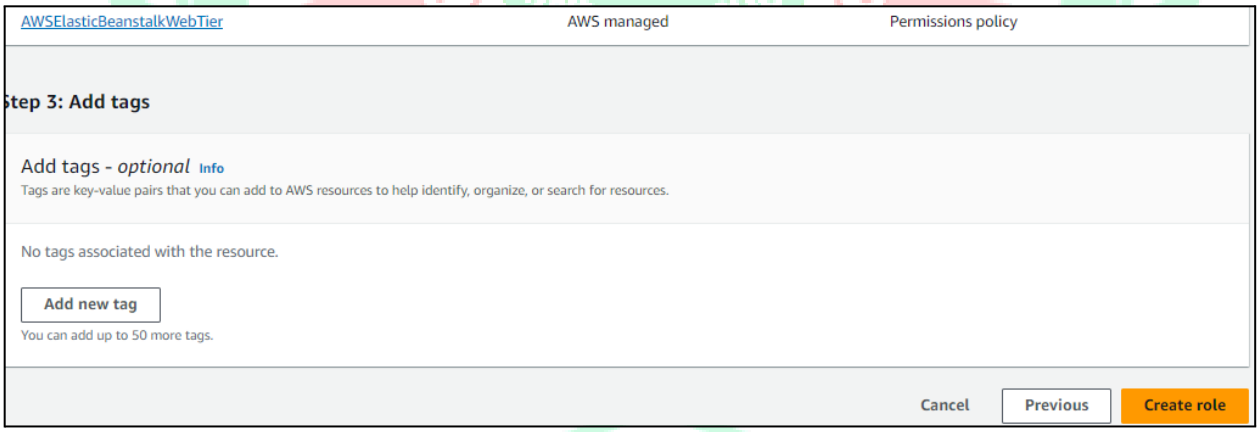

Skip build stage
Your pipeline will not include a build stage. Are you sure you want to skip this stage?
Cancel
Skip

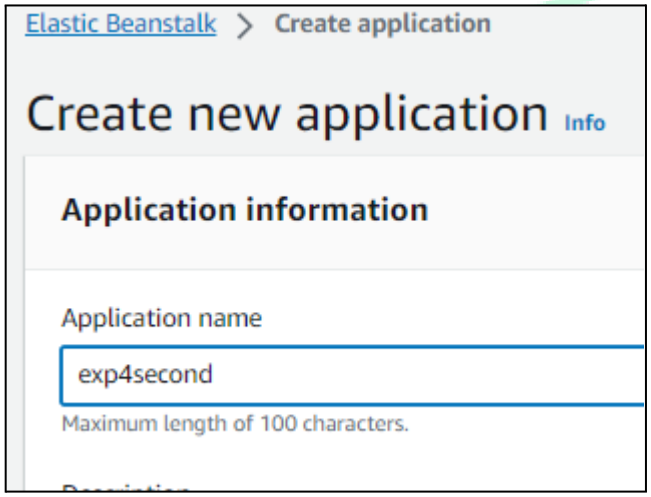

IAM > Roles
Roles (3) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Delete
Create role
Search
Role name
Trusted entities
Last activit
aws-elasticbeanstalk-service-role
AWS Service: elasticbeanstalk
AWSServiceRoleForSupport
AWS Service: support (Service-Linke
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service

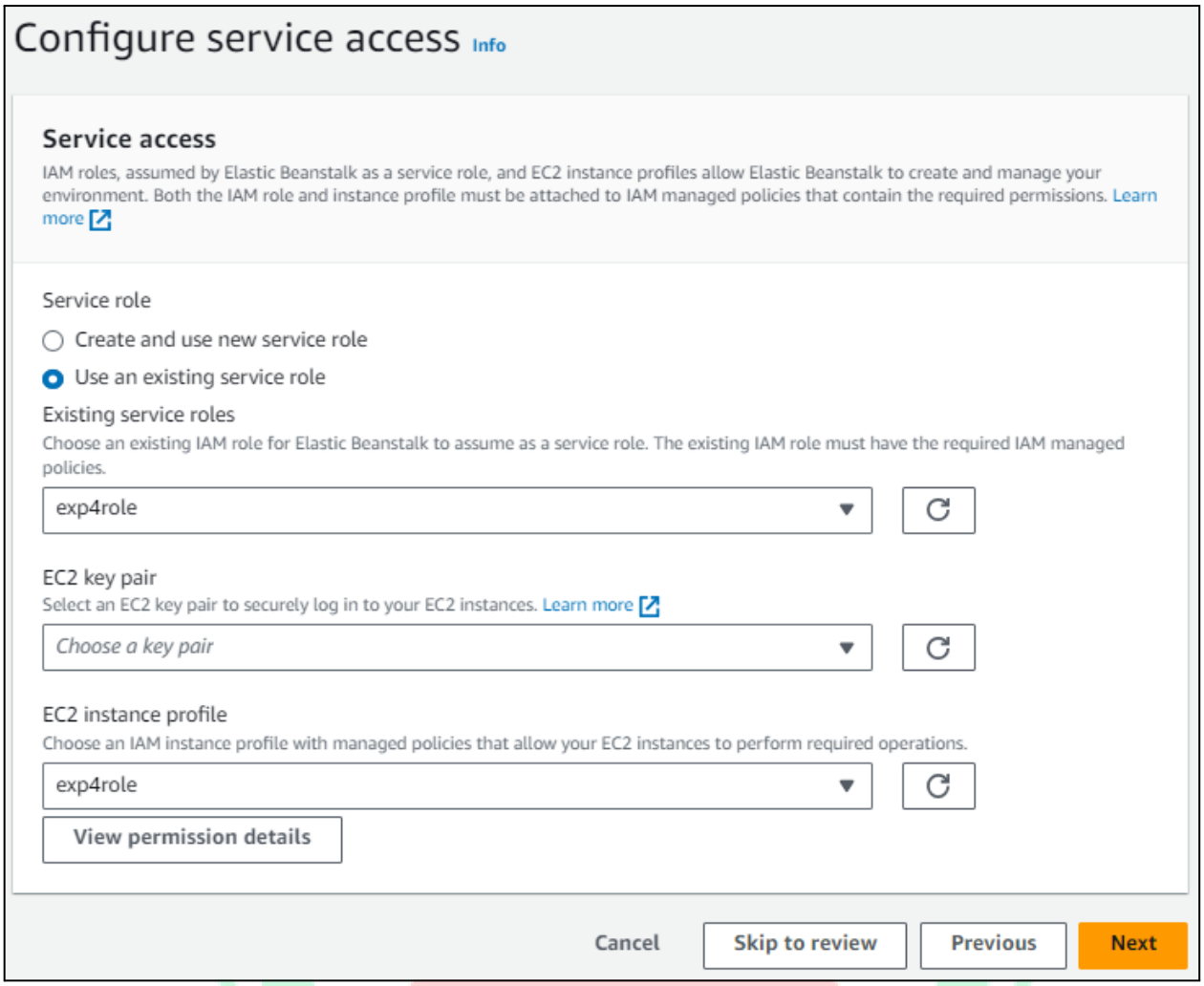

Select trusted entity Info
Trusted entity type
AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity
Allows users federated by the 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

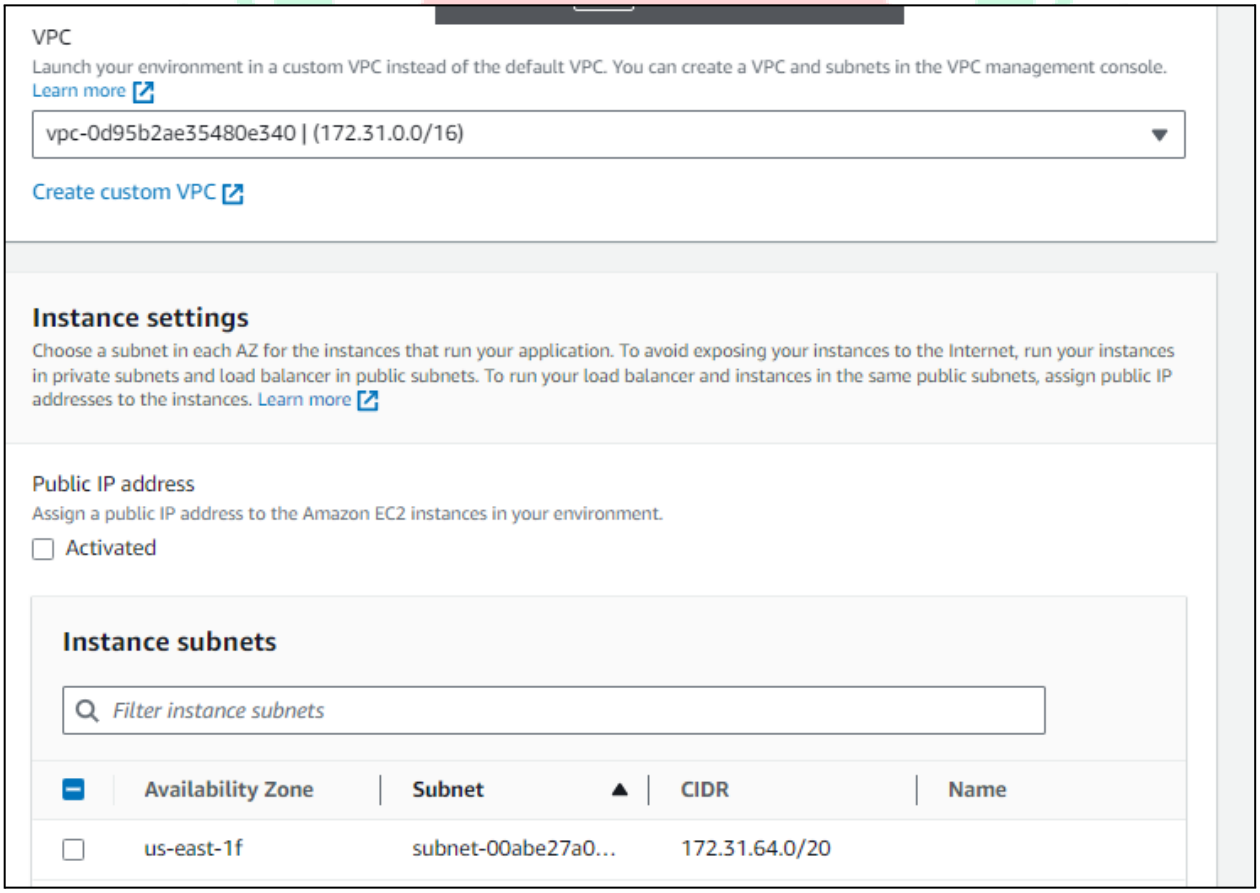

Add permissions Info
Permissions policies (1/881) Info
Choose one or more policies to attach to your new role.
Filter by Type
awselasticbeanstalkwe
All types
Policy name
Type
AWSElasticBeanstalkWebTier
AWS managed

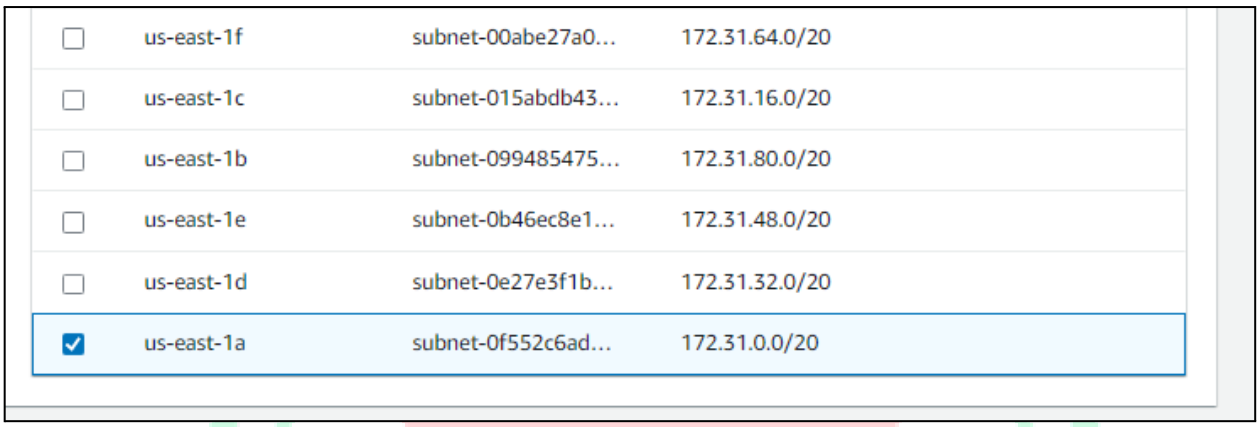

Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
exp4role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.


AWSElasticBeanstalkWebTier
AWS managed
Permissions policy
Step 3: Add tags
Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create role

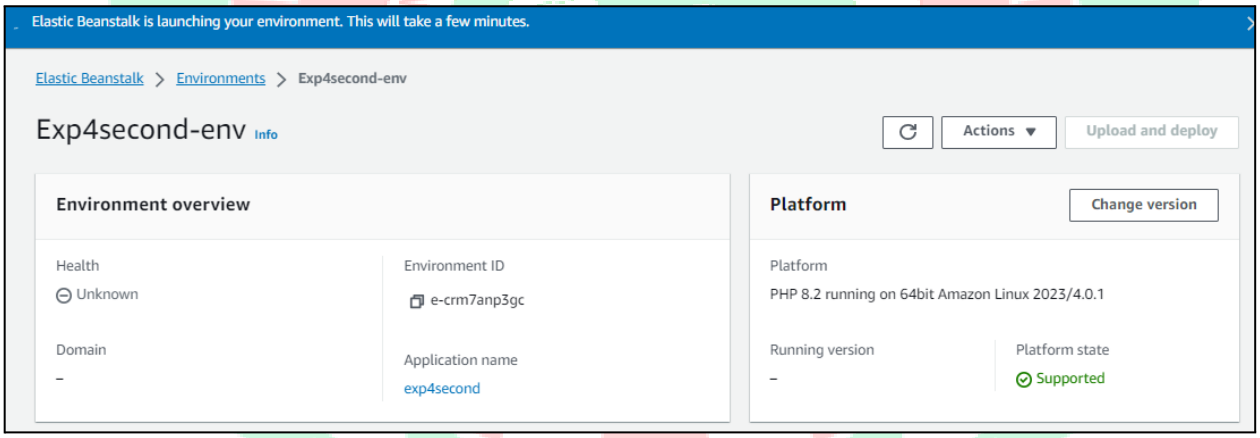

Elastic Beanstalk
Create application
Create new application Info
Application information
Application name
exp4second
Maximum length of 100 characters.

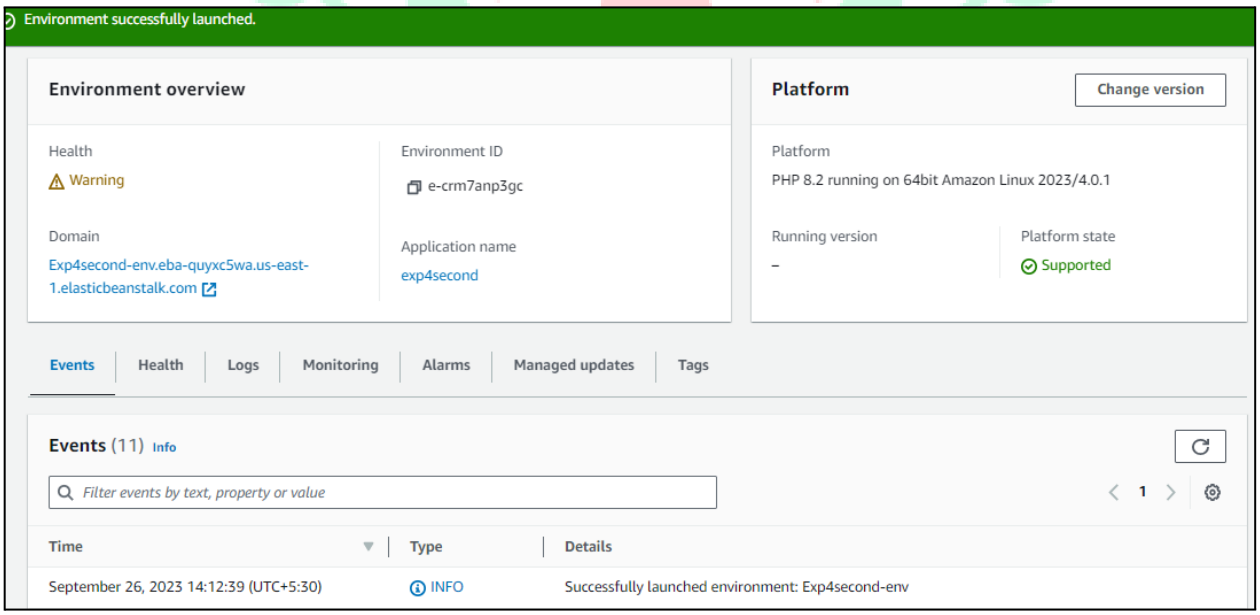

Configure service access Info
Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. 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.
exp4role
EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. Learn more
Choose a key pair
EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
exp4role
View permission details
Cancel
Skip to review
Previous
Next


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-0d95b2ae35480e340 | (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
us-east-1f
subnet-00abe27a0...
172.31.64.0/20

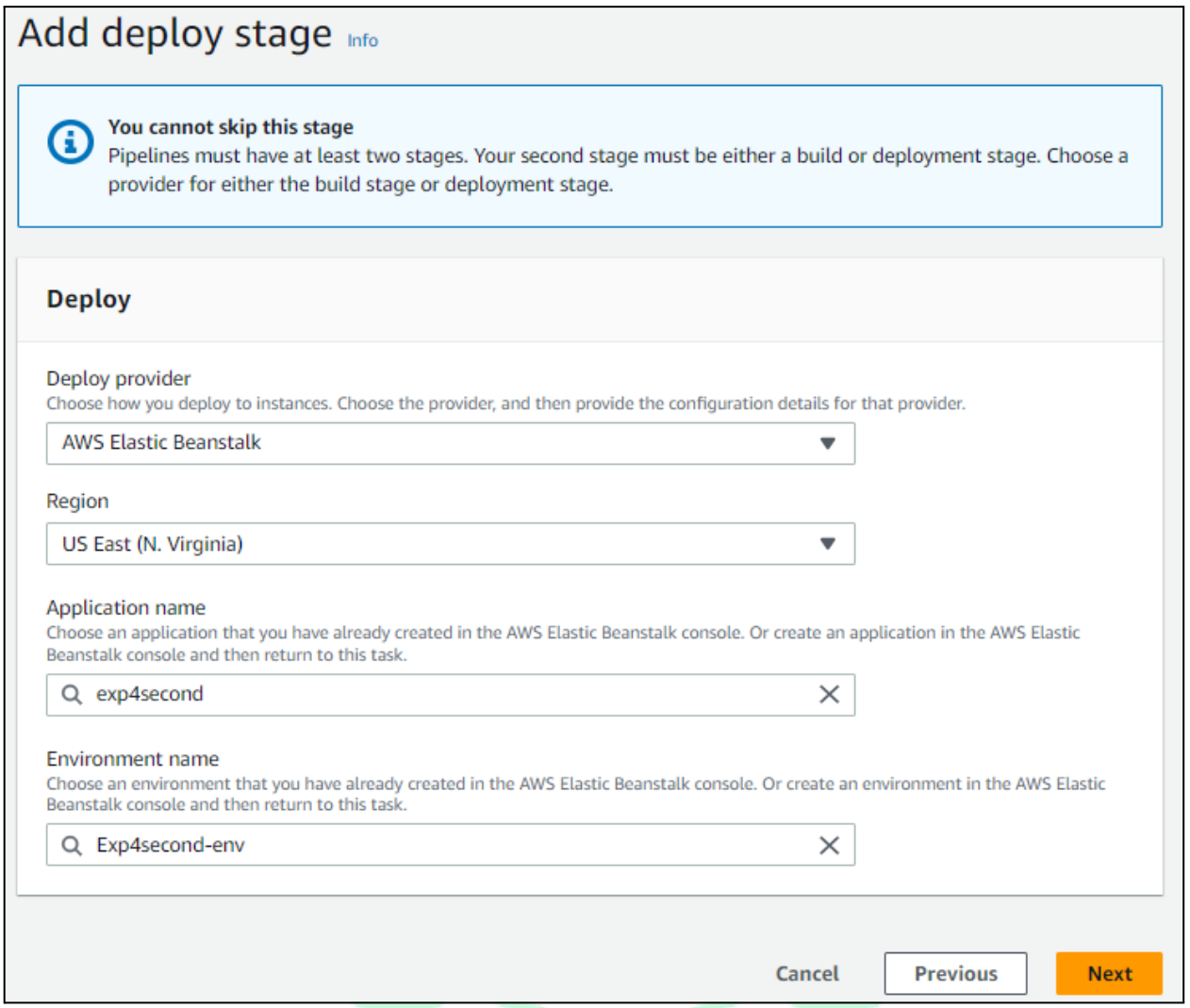


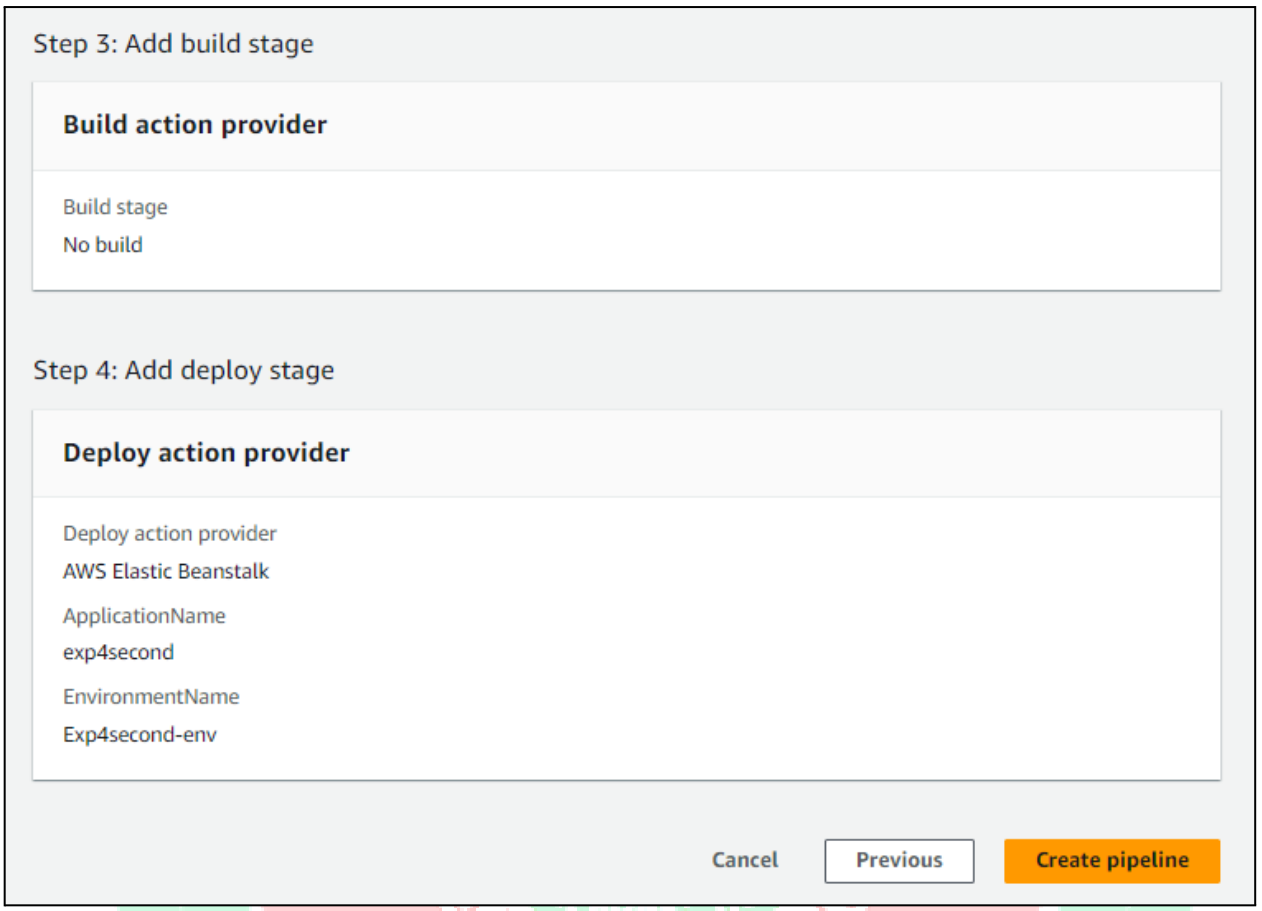
Create new pipeline: Source Provider = Amazon S3, Bucket = VaishaliBucket1, S3 Object Key = Copy S3 Uri from Amazon S3 bucket.
Add Deploy Stage:
Review the settings and create the pipeline.

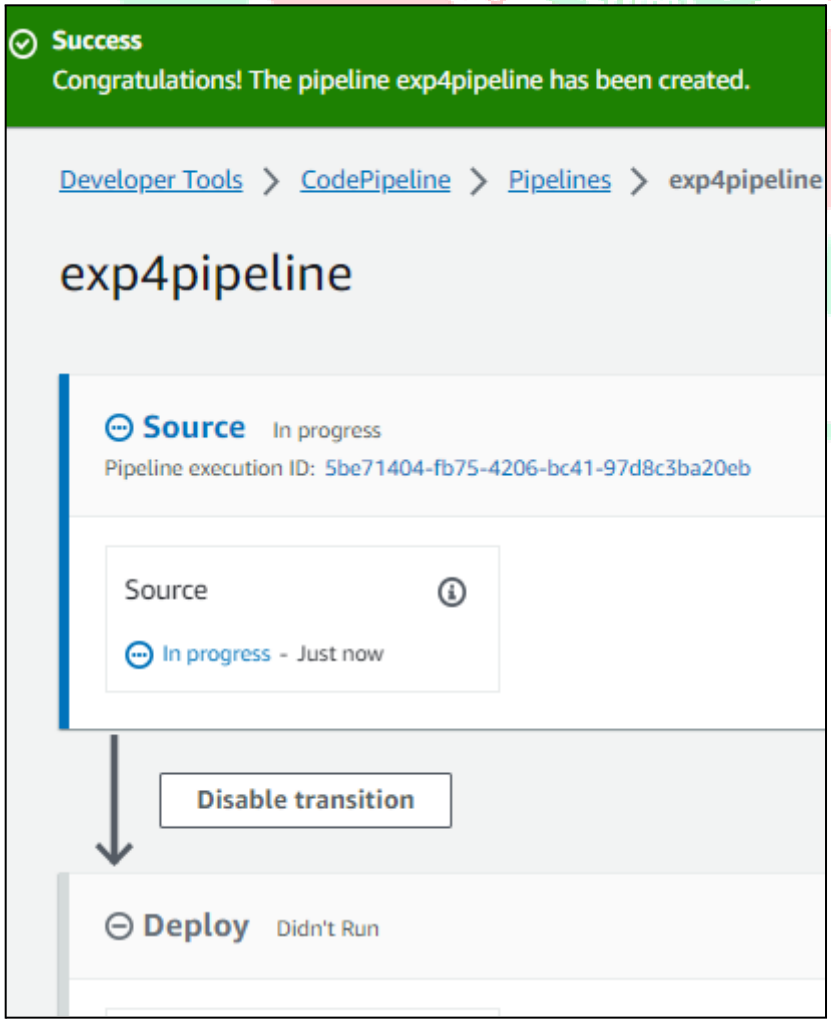


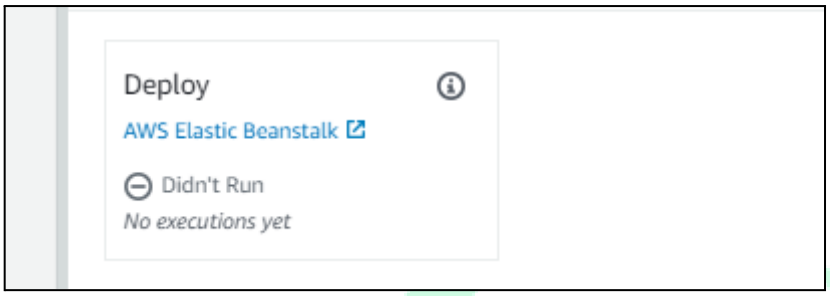


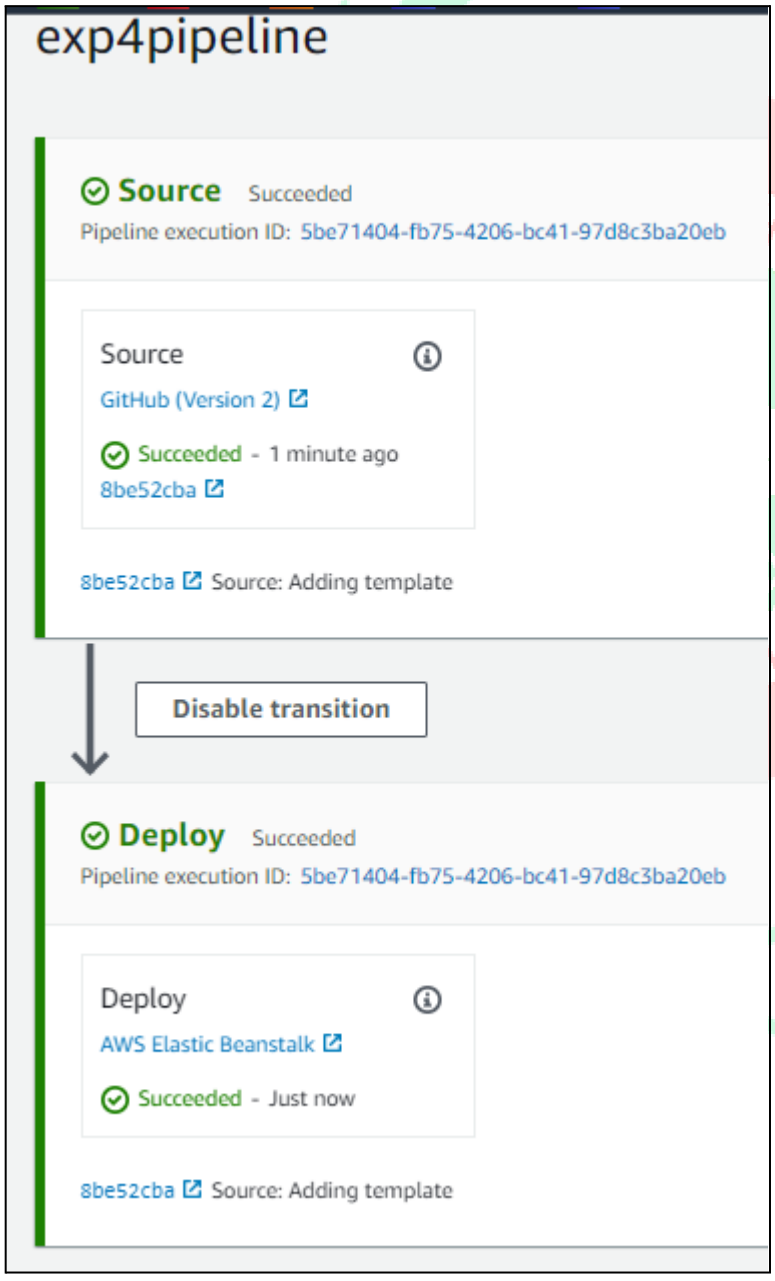
Add deploy stage and Give details : Deployement provider : AWS Elastic Beanstalk, Region,Application Name and Environment Name…(Auto suggested.)
Pipeline created successfully.




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
exp4second
EnvironmentName
Exp4second-env
Cancel
Previous
Create pipeline


Success
Congratulations! The pipeline exp4pipeline has been created.
Developer Tools
CodePipeline
Pipelines
exp4pipeline
exp4pipeline
Source
In progress
Pipeline execution ID: 5be71404-fb75-4206-bc41-97d8c3ba20eb
Source
In progress - Just now
Disable transition
Deploy
Didn't Run


Deploy
AWS Elastic Beanstalk
Didn't Run
No executions yet

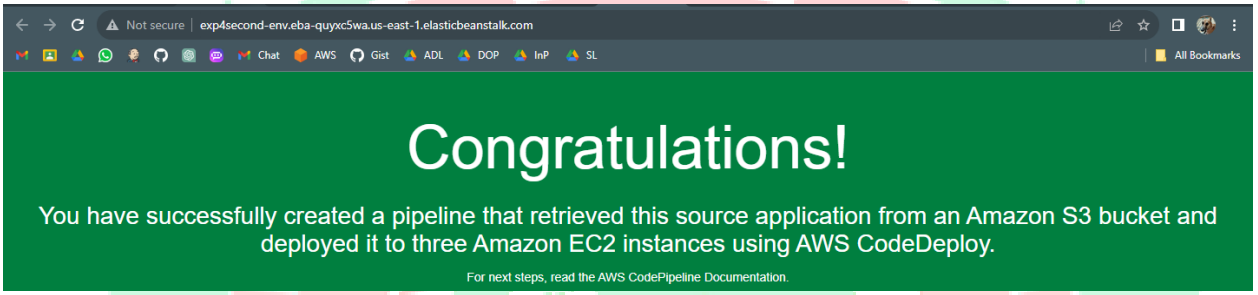

exp4pipeline
Source Succeeded
Pipeline execution ID: 5be71404-fb75-4206-bc41-97d8c3ba20eb
Source
GitHub (Version 2)
Succeeded - 1 minute ago
8be52cba
8be52cba Source: Adding template
Disable transition
Deploy Succeeded
Pipeline execution ID: 5be71404-fb75-4206-bc41-97d8c3ba20eb
Deploy
AWS Elastic Beanstalk
Succeeded - Just now
8be52cba Source: Adding template


Elastic Beanstalk > Applications > exp4second
Application exp4second environments (1) Info
Actions
Create new environment
Filter environments
Environment n... | Health | Date created | Domain | Running versions | Platform | Platform state
Exp4second-env | Warning | September 26, 20... | Exp4second-env.eba-quyxc5w... | code-pipeline-16... | PHP 8.2 running o... | Supported

After your pipeline is created, the pipeline status page appears and the pipeline automatically starts to run. You can view progress as well as success and failure messages as the pipeline perform each action.

To verify your pipeline ran successfully, monitor the progress of the pipeline as it moves through each stage. The status of each stage will change from No executions yet to In Progress, and then to either Succeeded or Failed. The pipeline should complete the first run within a few minutes.

Now go to your EBS environment and click on the URL to view the sample website you deployed.



Step 5: Commit a change and then update your app

In this step, you will revise the sample code and commit the change to your repository. CodePipeline will detect your updated sample code and then automatically initiate deploying it to

1.Visit your own copy of the repository that you forked in GitHub.

- Open index.html
- Select edit icon

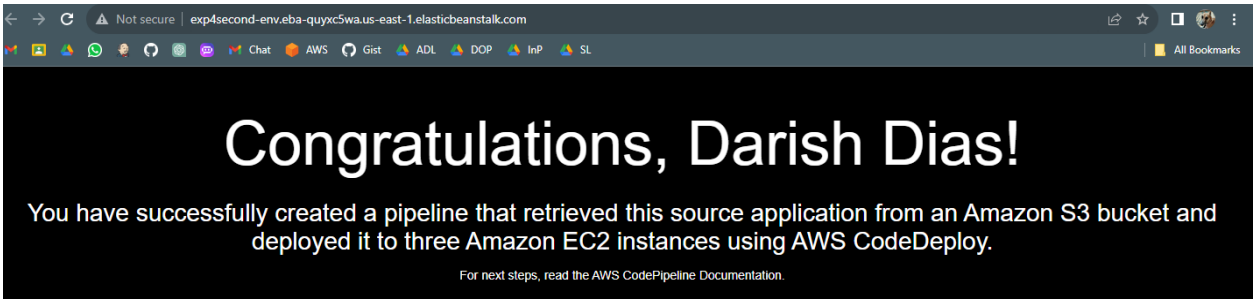
2.Update the webpage by copying and pasting the following text on line 30.

3.Commit the change to your repository.

4.Return to your pipeline in the CodePipeline console. In a few minutes, you should see the Source change to blue, indicating that the pipeline has detected the changes you made to your source repository. Once this occurs, it will automatically move the updated code to Elastic Beanstalk.

- After the pipeline status displays Succeeded, in the status area for the Beta stage, click AWS Elastic Beanstalk.

5. The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier. And click the URL again from EBS environment again.



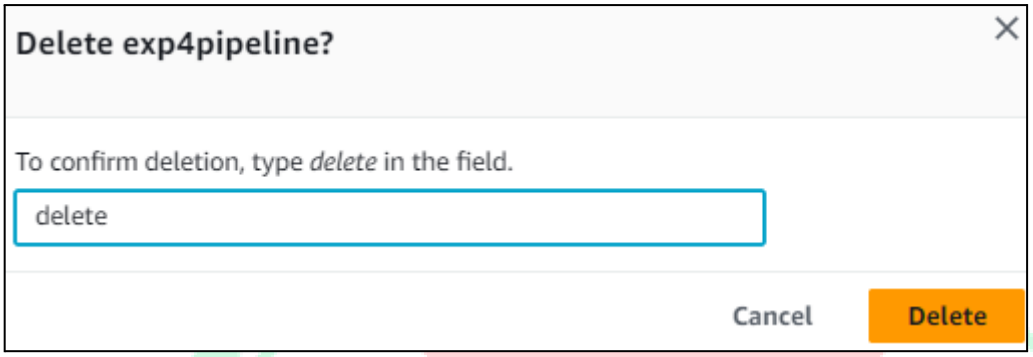
Step 6: Clean up the resources

To avoid future charges, you will delete all the resources you launched which includes the pipeline, the Elastic Beanstalk application, and the source you set up to host the code.

First, you will delete your pipeline:

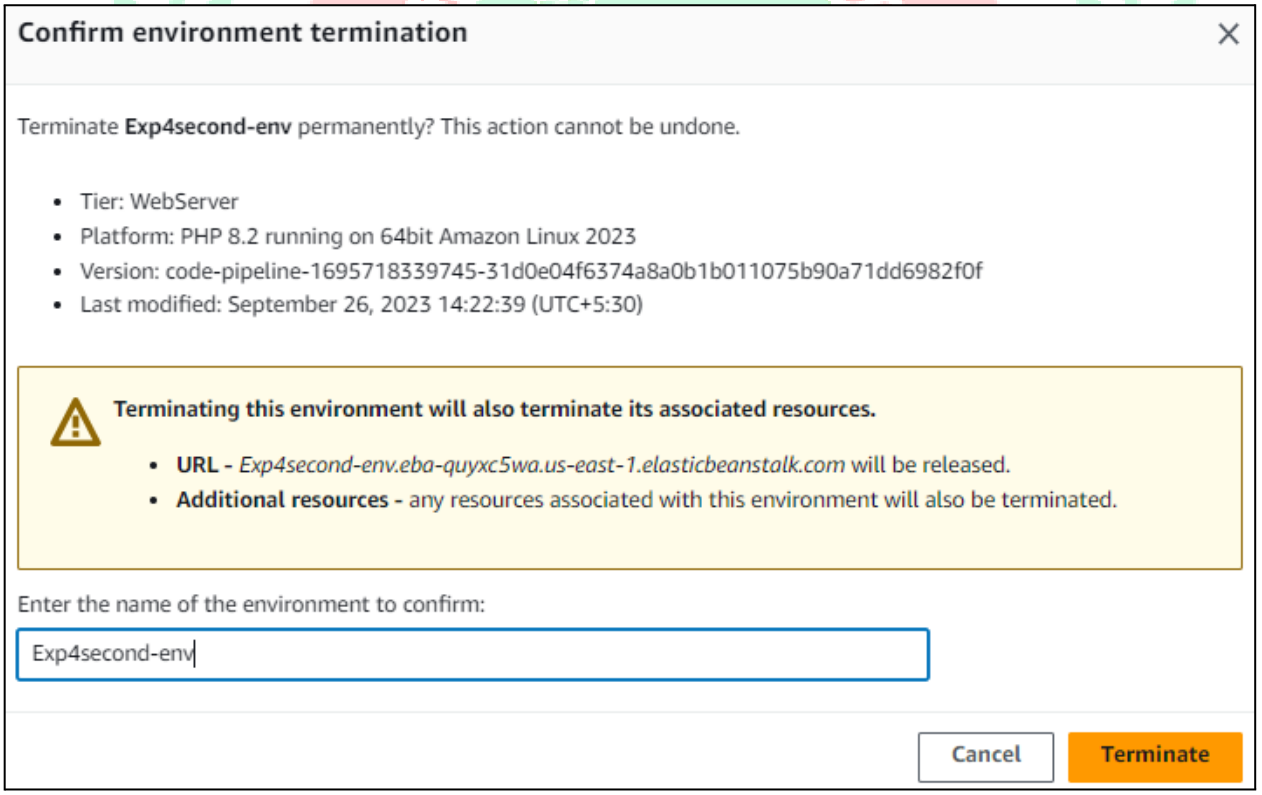
- In the pipeline view, click Edit.

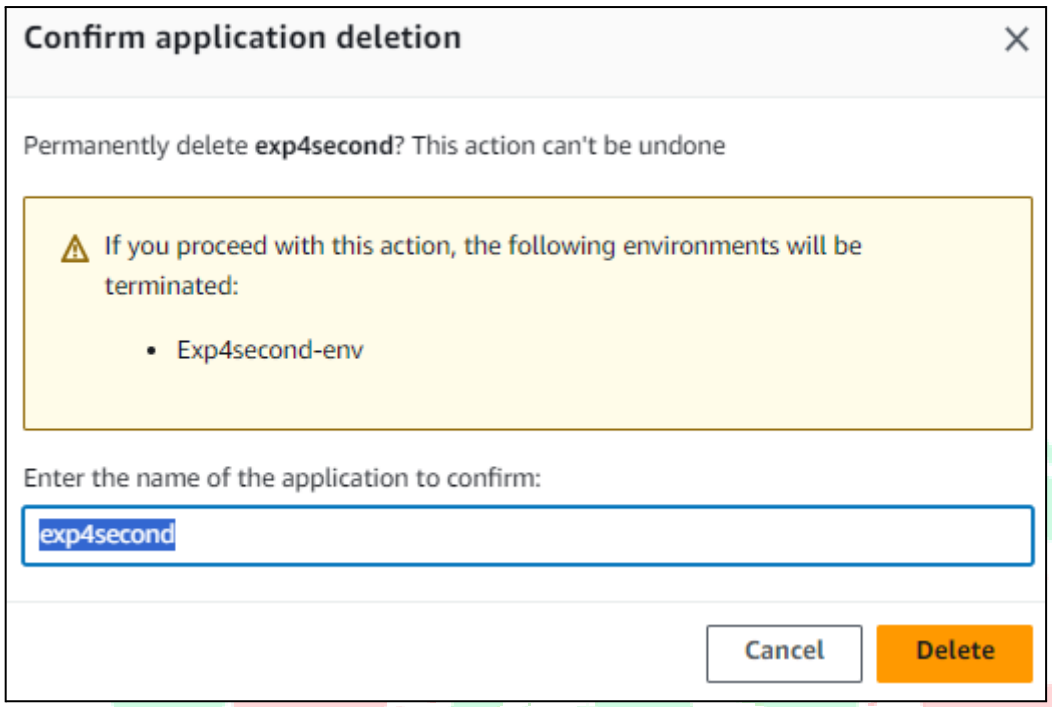
- Click Delete.
- Type in the name of your pipeline and click Delete.

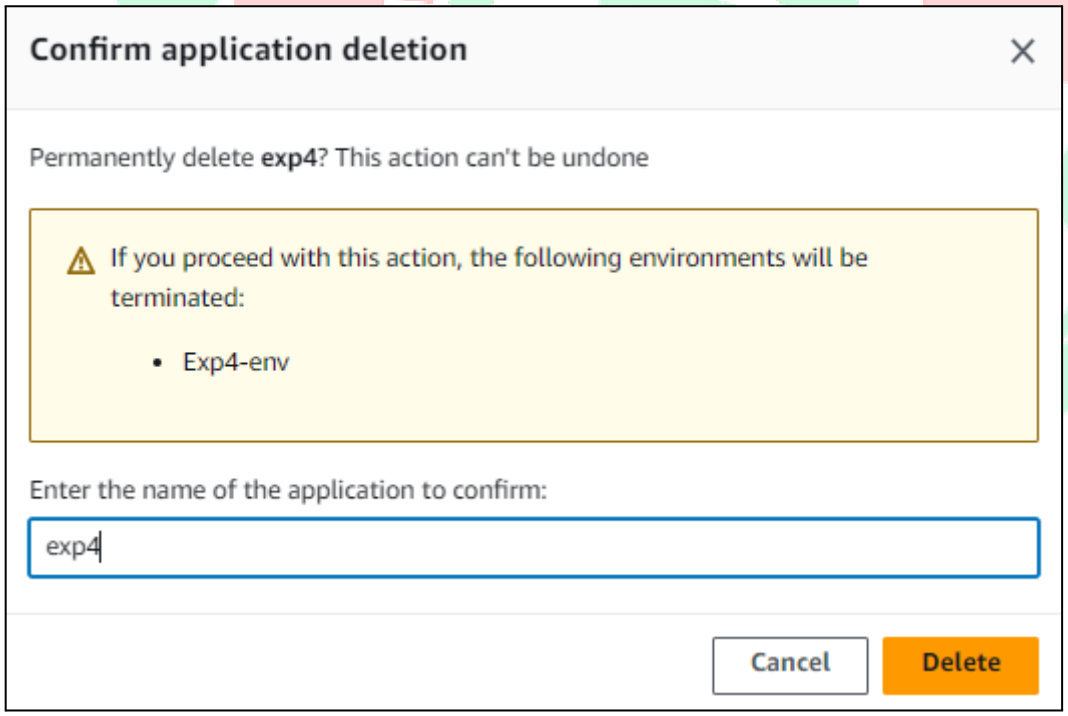


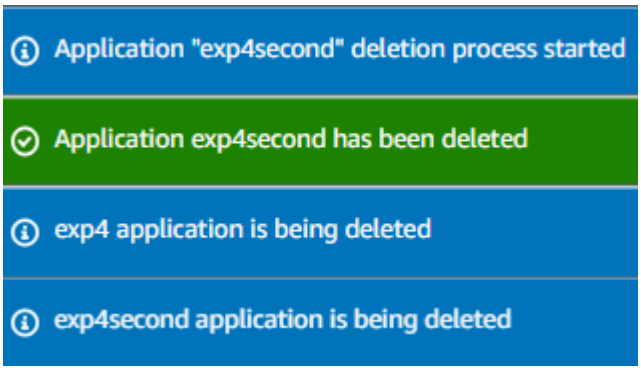
Second, delete your Elastic Beanstalk application:

- Visit the Elastic Beanstalk console.
- Click Actions.
- Then click Terminate Environment.




Confirm application deletion
Permanently delete exp4second? This action can't be undone
If you proceed with this action, the following environments will be terminated:
Exp4second-env
Enter the name of the application to confirm:
exp4second
Cancel
Delete


Confirm application deletion
Permanently delete exp4? This action can't be undone
If you proceed with this action, the following environments will be terminated:
Exp4-env
Enter the name of the application to confirm:
exp4
Cancel
Delete


Application "exp4second" deletion process started
Application exp4second has been deleted
exp4 application is being deleted
exp4second application is being deleted

- We have successfully created an automated software release pipeline using AWS CodePipeline! Using CodePipeline, We created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. Pipeline will automatically deploy your code every time there is a code change.

8. Post-Experiments Exercise

A. Extended Theory:

- Deploy Web Application using Elastic BeanStalk
- AWS Storage Service: S3

B. Questions:

Q.1 Does AWS Elastic Beanstalk store anything in Amazon S3?

Q. 2 What database solutions can we use with AWS Elastic Beanstalk?

C. Conclusion:

Write the significance of the topic studied in the experiment.

D. References:

https://aws.amazon.com/elasticbeanstalk/faqs/#

https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingBucket.html

---