SNEHAL A. PATIL D15A 39

EXPERIMENT NO: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.

Theory:

AWS Elastic Beanstalk

AWS Elastic Beanstalk is a versatile Platform as a Service (PaaS) offering from Amazon Web Services (AWS) that simplifies the deployment and management of applications in the cloud. It abstracts much of the underlying infrastructure complexity, allowing developers to focus on building and maintaining their applications while AWS takes care of provisioning resources like EC2 instances, load balancers, and databases.

Key Features of Elastic Beanstalk:

1. **Simplified Application Deployment:**
   - Elastic Beanstalk streamlines the deployment process by automatically handling the setup, configuration, and maintenance of the infrastructure required to run your application. This allows developers to concentrate on writing and refining their code rather than managing servers.
2. **Broad Language and Framework Support:**
   - Elastic Beanstalk supports multiple programming languages and frameworks, such as Java, .NET, Node.js, Python, Ruby, PHP, Go, and Docker. This flexibility makes it easy for developers to deploy applications built with their preferred technologies.
3. **Automatic Resource Scaling:**
   - Elastic Beanstalk includes automatic scaling capabilities, dynamically adjusting the number of instances running your application based on current demand. This ensures that your application remains responsive and cost-efficient during traffic fluctuations.
4. **Robust Health Monitoring:**
   - The platform continuously monitors the health of your application and provides detailed metrics and logs. It can automatically replace any unhealthy instances, ensuring high availability and reliability of your application.
5. **Multiple Environment Support:**
   - Elastic Beanstalk allows you to manage different environments, such as development, staging, and production. You can deploy updates to one environment without impacting others, providing a controlled and organized deployment process.

AWS CodeBuild

AWS CodeBuild is a fully managed continuous integration (CI) service that compiles your source code, runs tests, and produces deployable artifacts. It automates the build process, ensuring consistent compilation and testing across various environments.

Key Features of AWS CodeBuild:

1. **Managed Build Infrastructure:**

    - With AWS CodeBuild, there's no need to manage your own build servers. AWS handles all infrastructure management, allowing you to focus on developing and testing your applications.
2. **Automatic Scaling:**
    - CodeBuild automatically scales to meet your needs, handling multiple builds simultaneously. This ensures that even during peak times, your builds are processed quickly and efficiently.
3. **Customizable Build Environments:**
    - You can define custom build environments using Docker images, providing the flexibility to tailor the build environment to meet the specific needs of your application.
4. **Seamless AWS Integration:**
    - CodeBuild integrates effortlessly with other AWS services, such as CodePipeline, CodeCommit, and S3, enabling you to create a complete CI/CD pipeline that automates your entire build and deployment process.
5. **Cost-Effective Pricing:**
    - CodeBuild uses a pay-as-you-go pricing model, charging you only for the build time you consume. This makes it an economical solution for automating your build processes.

Deploying on S3 Using AWS CodePipeline

AWS CodePipeline is a fully managed continuous delivery service that automates the steps required to release your software, from building and testing to deployment. By integrating with services like CodeBuild and S3, CodePipeline provides a streamlined process for deploying applications.

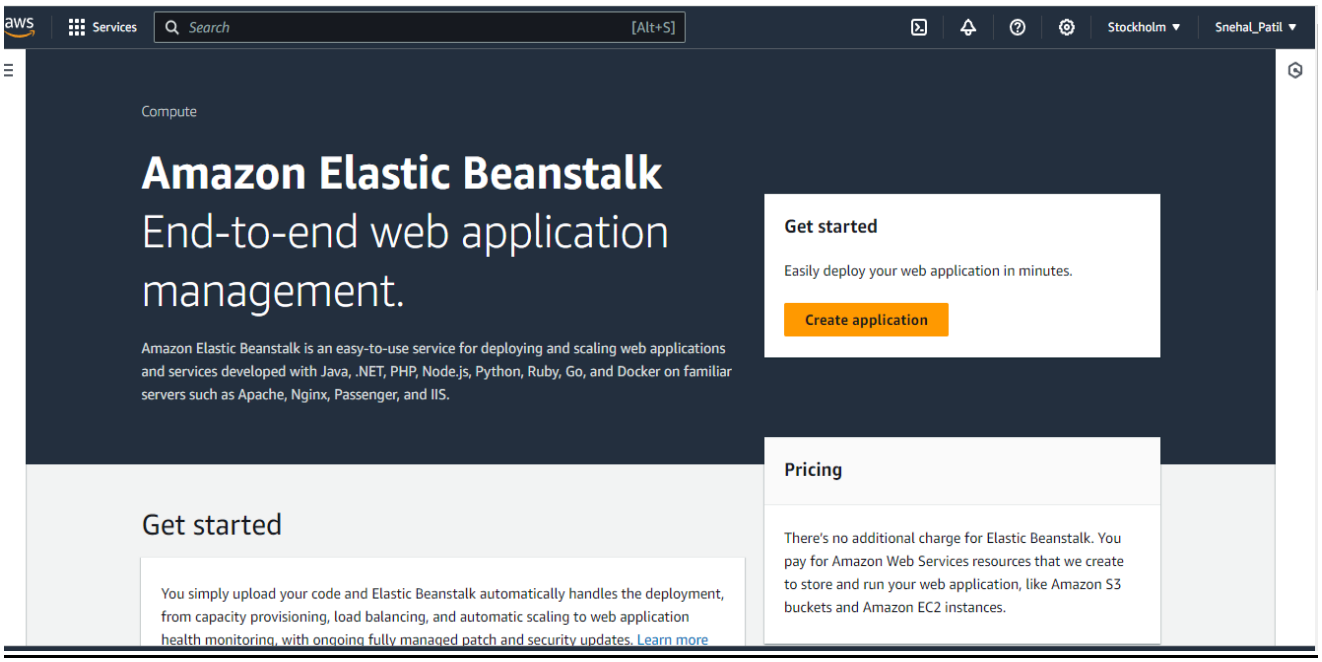
Key Steps to Deploy on S3 Using AWS CodePipeline:

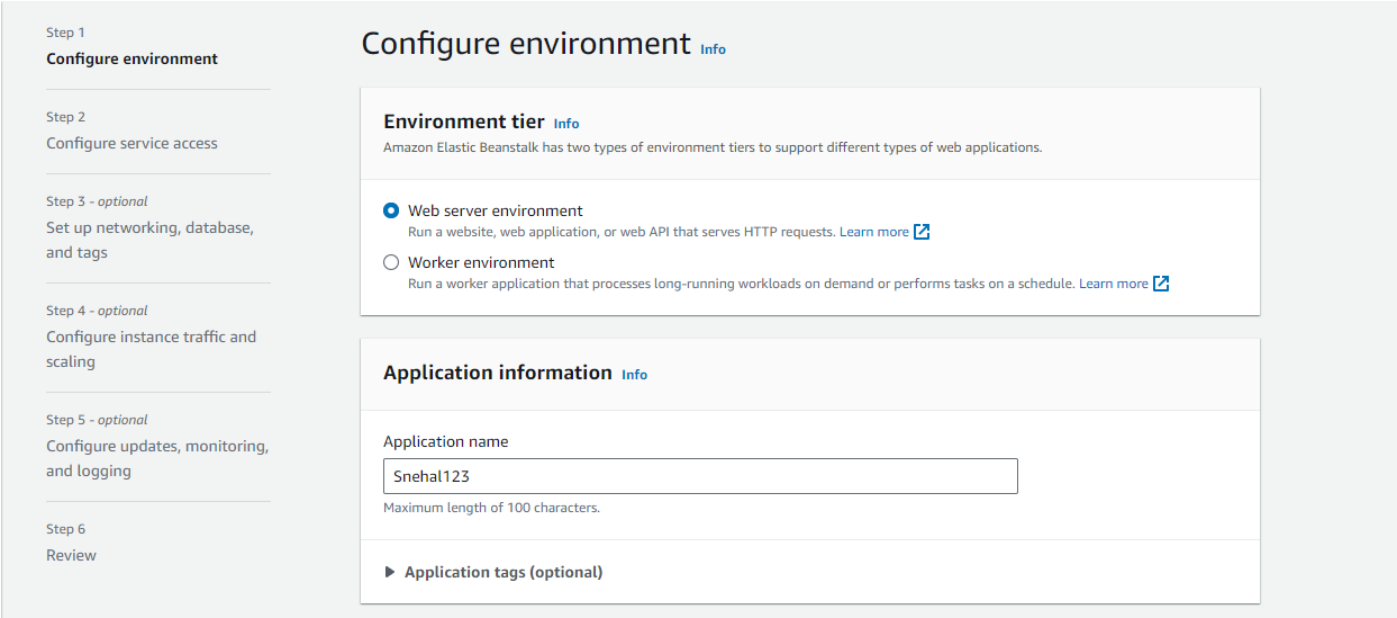
1. **Source Code Retrieval:**
    - The pipeline begins with a source stage, where the source code is retrieved from a repository such as AWS CodeCommit, GitHub, or an S3 bucket. This source code is then used in subsequent stages for building and deploying the application.
2. **Build and Package with CodeBuild:**
    - In the build stage, AWS CodePipeline triggers CodeBuild to compile the source code, run tests, and package the application. The output of this process is a deployable artifact stored in an S3 bucket.
3. **Deployment to S3:**
    - In the deployment stage, CodePipeline automatically deploys the artifacts generated during the build stage to an S3 bucket. The S3 bucket can be configured to host a static website or store files accessed by your application.
4. **Automation and Notifications:**
    - CodePipeline can be configured to automatically trigger builds and deployments when changes are made to the source code repository. Additionally, it can send notifications via Amazon SNS to keep you informed about the status of your pipeline.
5. **Pipeline Monitoring and Logging:**
    - CodePipeline provides tools for monitoring and logging the progress of each stage in your pipeline. This allows you to quickly identify and address any issues that arise during the build or deployment process, ensuring a smooth release cycle.

Implementation :

Elastic Beanstalk

Step 1: create environment





Environment information Info

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

Environment name

Snehal123-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 .eu-north-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

Python

Platform branch

Python 3.11 running on 64bit Amazon Linux 2023

Platform version

4.1.3 (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**

IAM > Dashboard

IAM Dashboard

Security recommendations 1

Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.

Add MFA

Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	2	0	0

AWS Account

Account ID
825765388229

Account Alias
Create

Sign-in URL for IAM users in this account
https://825765388229.signin.aws.amazon.com/console

Quick Links

My security credentials

IAM > Roles

Roles (2) Info

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

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

Roles Anywhere Info

Manage

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

X.509 Standard

Temporary credentials

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (3/946) Info

Choose one or more policies to attach to your new role.

Filter by Type

Beanstalk | All types | 14 matches

1

Policy name	Type	Description
AdministratorAccess-...	AWS managed	Grants account administrative permissions...
AWSElasticBeanstalkC...	AWS managed	Provide the instance in your custom platf...
AWSElasticBeanstalkE...	AWS managed	AWS Elastic Beanstalk Service policy for H...
AWSElasticBeanstalk...	AWS managed	This policy is for the AWS Elastic Beanstal...
AWSElasticBeanstalk...	AWS managed	Provide the instances in your multicontain...
AWSElasticBeanstalkR...	AWS managed	Grants read-only permissions. Explicitly all...

☑	AWSElasticBeanstalk...	AWS managed	Provide the instances in your multicontain...
☐	AWSElasticBeanstalkR...	AWS managed	Grants read-only permissions. Explicitly all...
☐	AWSElasticBeanstalkR...	AWS managed	AWSElasticBeanstalkRoleCore (Elastic Bea...
☐	AWSElasticBeanstalkR...	AWS managed	(Elastic Beanstalk operations role) Allows ...
☐	AWSElasticBeanstalkR...	AWS managed	(Elastic Beanstalk operations role) Allows ...
☐	AWSElasticBeanstalkR...	AWS managed	(Elastic Beanstalk operations role) Allows ...
☐	AWSElasticBeanstalkR...	AWS managed	(Elastic Beanstalk operations role) Allows ...
☐	AWSElasticBeanstalkR...	AWS managed	(Elastic Beanstalk operations role) Allows ...
☑	AWSElasticBeanstalk...	AWS managed	Provide the instances in your web server e...
☑	AWSElasticBeanstalk...	AWS managed	Provide the instances in your worker envir...

▸ **Set permissions boundary - *optional***

Cancel | Previous | Next

Identity and Access Management (IAM)

Search IAM

- Dashboard
- Access management
  - User groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access Analyzer
    - External access

✓ Role aws-Snehal-ec2-role created. View role

Roles (3) Info

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

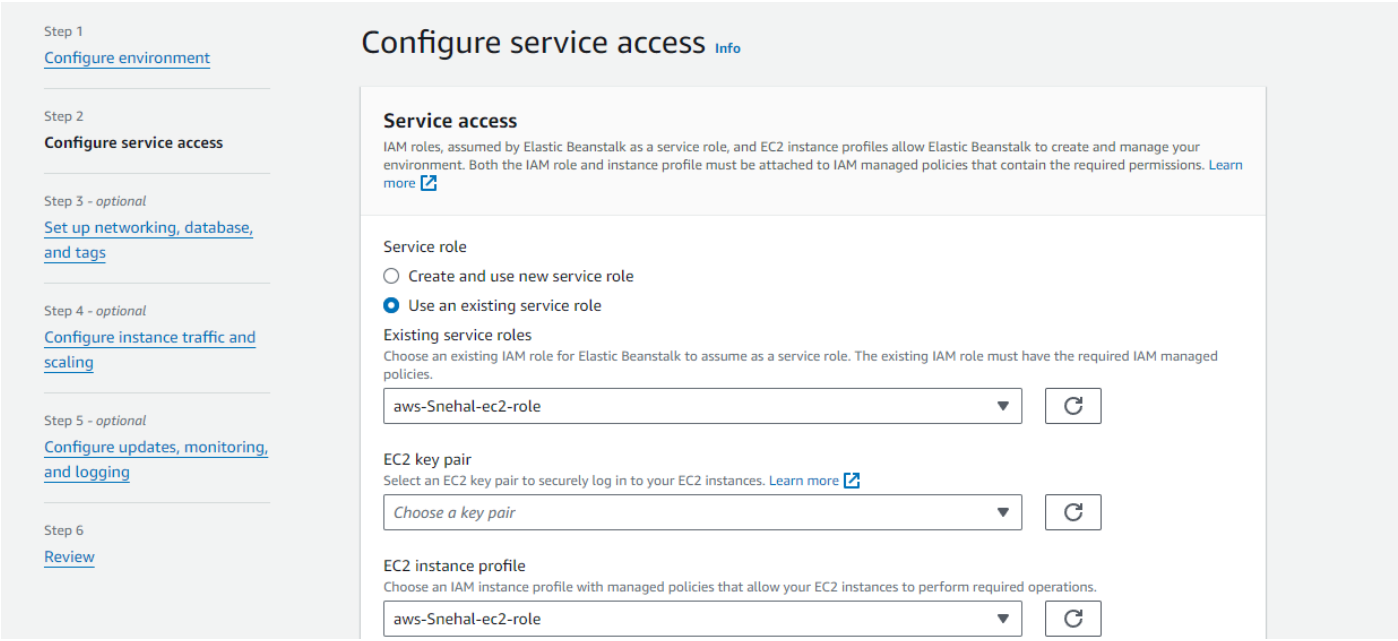
Role name	Trusted entities	Last activity
aws-Snehal-ec2-role	AWS Service: ec2	
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	

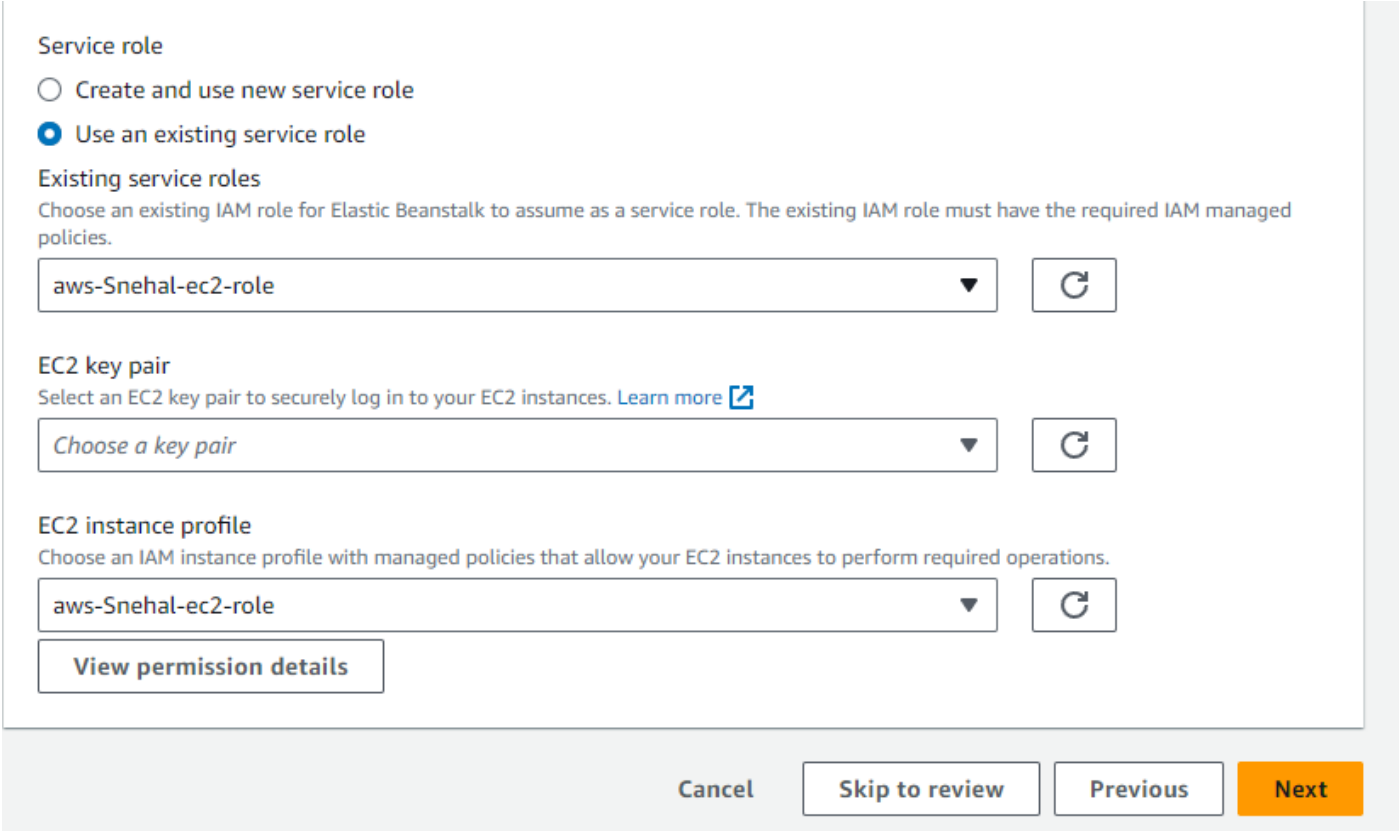
Roles Anywhere Info

Manage

Authenticate your non AWS workloads and securely provide access to AWS services.

Step 2 : add your Ec2 key pair and instance profile





Review Info

Step 1: Configure environment

Edit

Environment information

Environment tier
Web server environment

Application name
Snehal123

Environment name
Snehal123-env

Application code
Sample application

Platform
arn:aws:elasticbeanstalk:eu-north-1::platform/Python 3.11 running on 64bit Amazon Linux 2023/4.1.3

Step 2: Configure service access

Edit

Service access Info

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
arn:aws:iam::825765388229:role/aws-Snehal-ec2-role

EC2 instance profile
aws-Snehal-ec2-role

Step 3: Set up networking, database, and tags

Edit

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

No options configured

Tags

Step 4: Configure instance traffic and scaling

Edit

Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic.Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

IMDSv1

Deactivated

Capacity

Environment type	Fleet composition	On-demand base
Single instance	On-Demand instance	0
On-demand above base	Capacity rebalancing	Scaling cooldown
0	Deactivated	360
Processor type	Instance types	AMI ID
x86_64	t3.micro,t3.small	ami-030d0ebd08fe18778

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging Info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
enhanced	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false

Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold

Platform software

Lifecycle false	Log streaming Deactivated	NumProcesses 1
NumThreads 15	WSGIPath application	Proxy server nginx
Logs retention 7	Rotate logs Deactivated	Update level minor
X-Ray enabled Deactivated		

Environment properties

Key	Value
PYTHONPATH	/var/app/venv/staging-LQM1lest/bin

Cancel | Previous | Submit

Environment successfully launched.

Elastic Beanstalk > Environments > Snehal123-env

Snehal123-env Info

Actions | Upload and deploy

Environment overview

Health
Warning

Environment ID
e-nm5tuux6pa

Domain
Snehal123-env.eba-rpipmwt9.eu-north-1.elasticbeanstalk.com

Application name
Snehal123

Platform

Change version

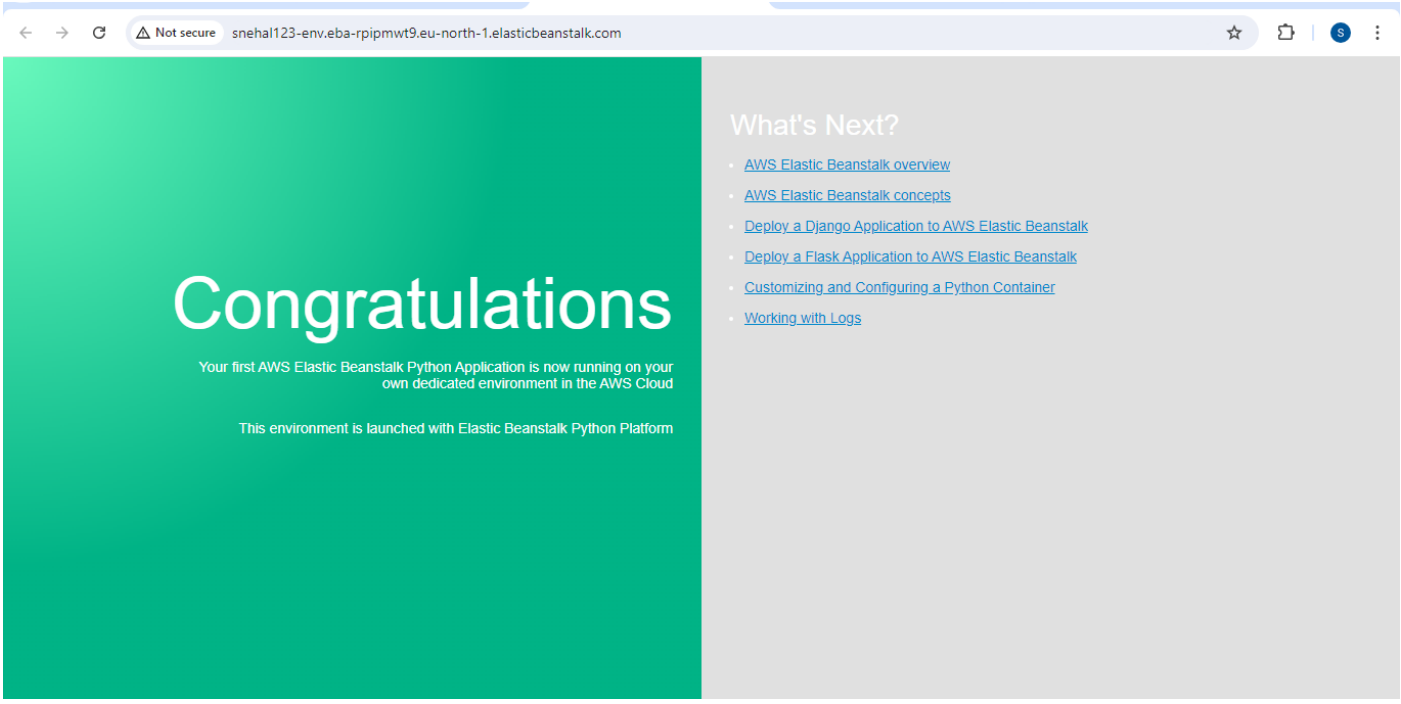
Platform
Python 3.11 running on 64bit Amazon Linux 2023/4.1.3

Running version
–

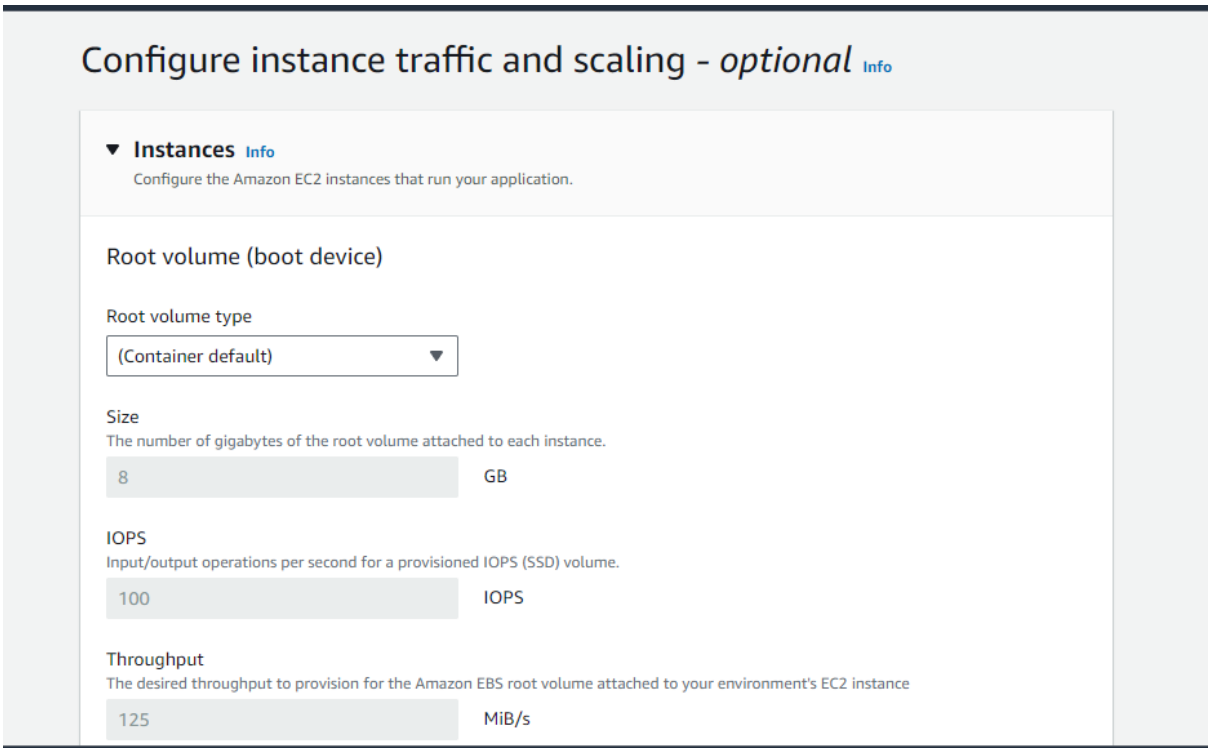
Platform state
Supported

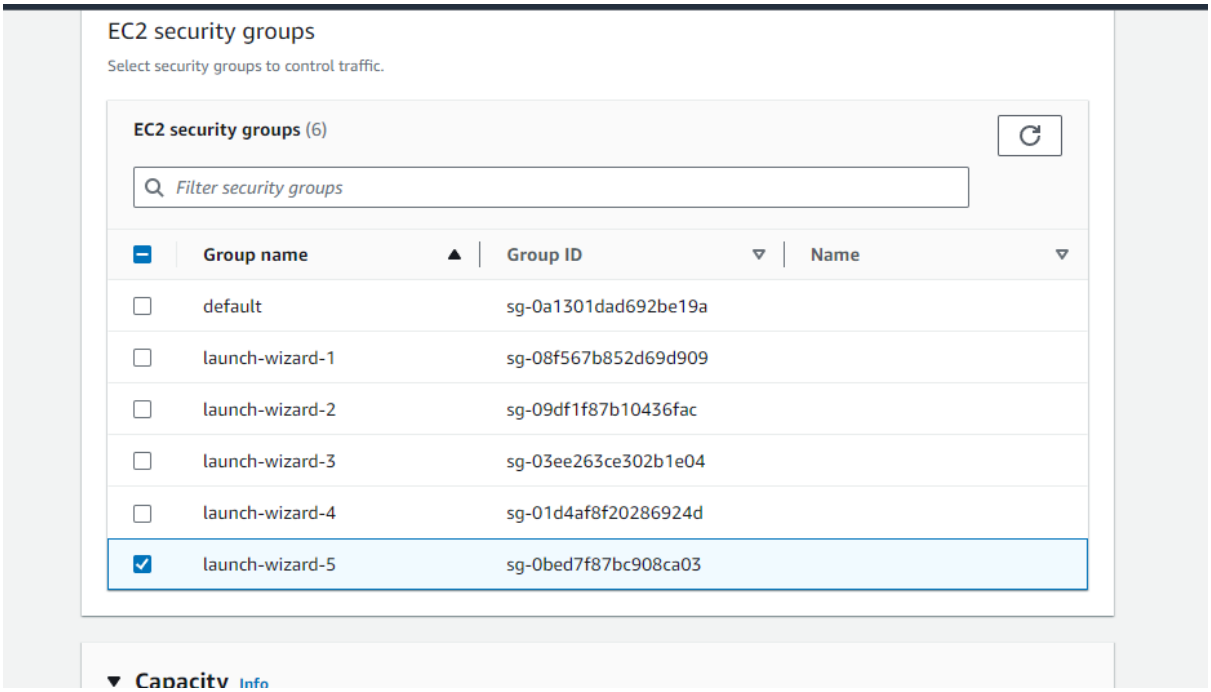
Events | Health | Logs | Monitoring | Alarms | Managed updates | Tags

Step 3: Beanstalk environment is created



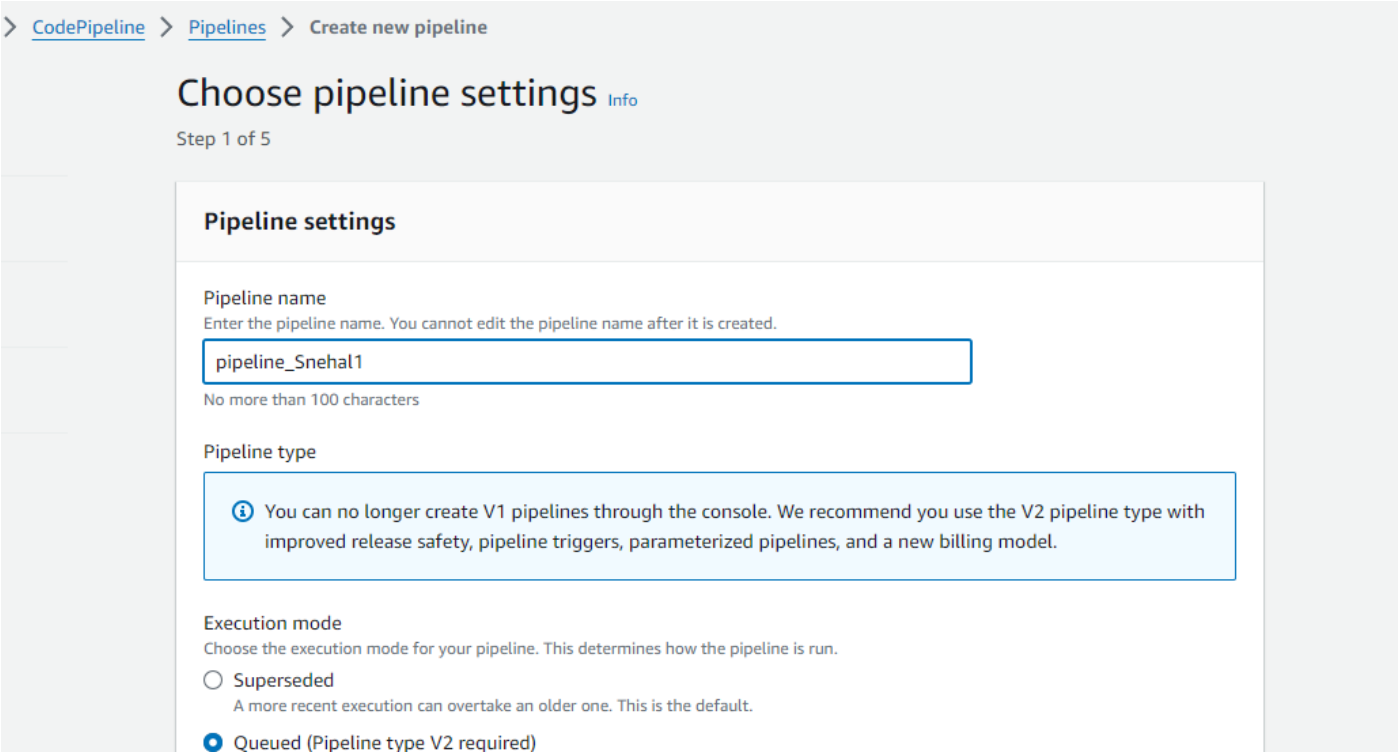
Step 4 : add security config and review all settings



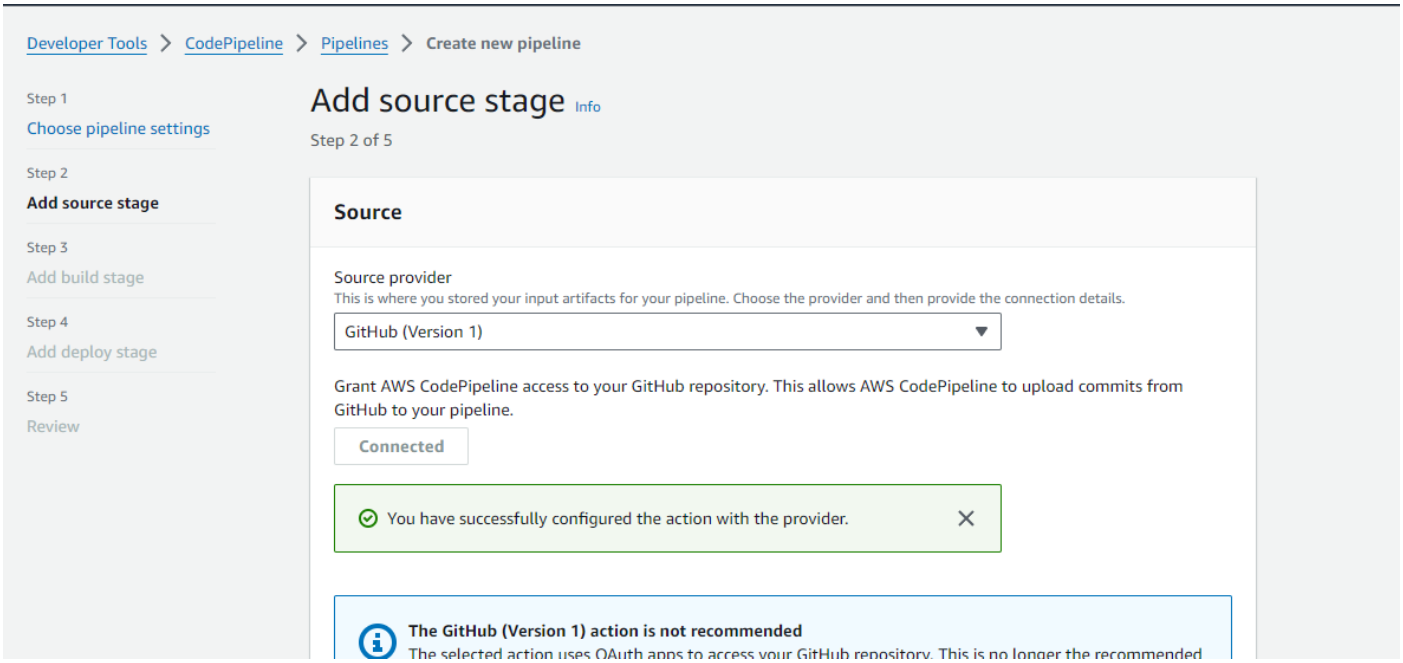


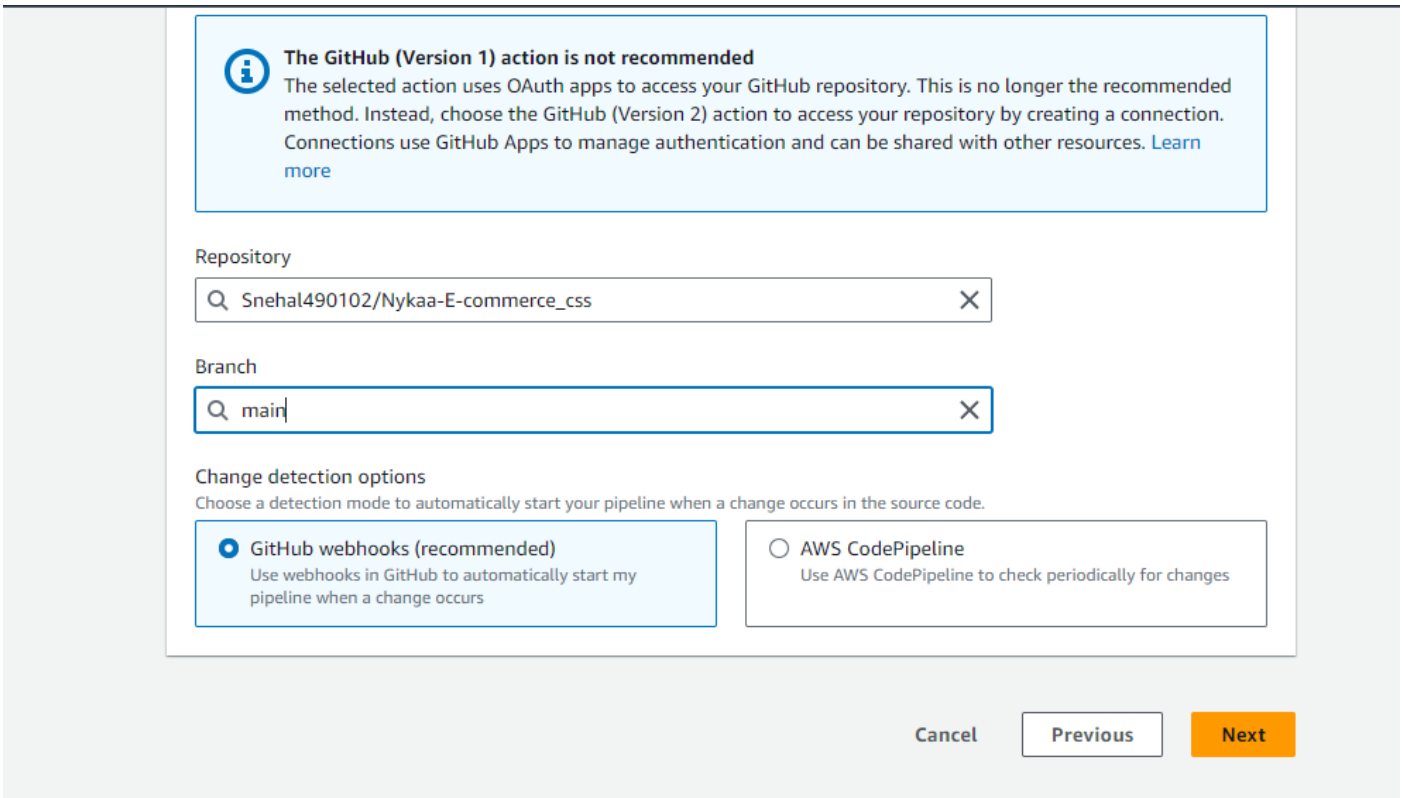
PIPELINE CREATION:

Step 1 : click on create pipeline and give name



Step 2 : Add Your github account and add the file to add to pipeline deployment





Step 3 : Add deploy config choosing the elastic beanstalk

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

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
Europe (Stockholm)

Input artifacts
Choose an input artifact for this action. Learn more

Region
Europe (Stockholm)

Input artifacts
Choose an input artifact for this action. Learn more

SourceArtifact

No more than 100 characters

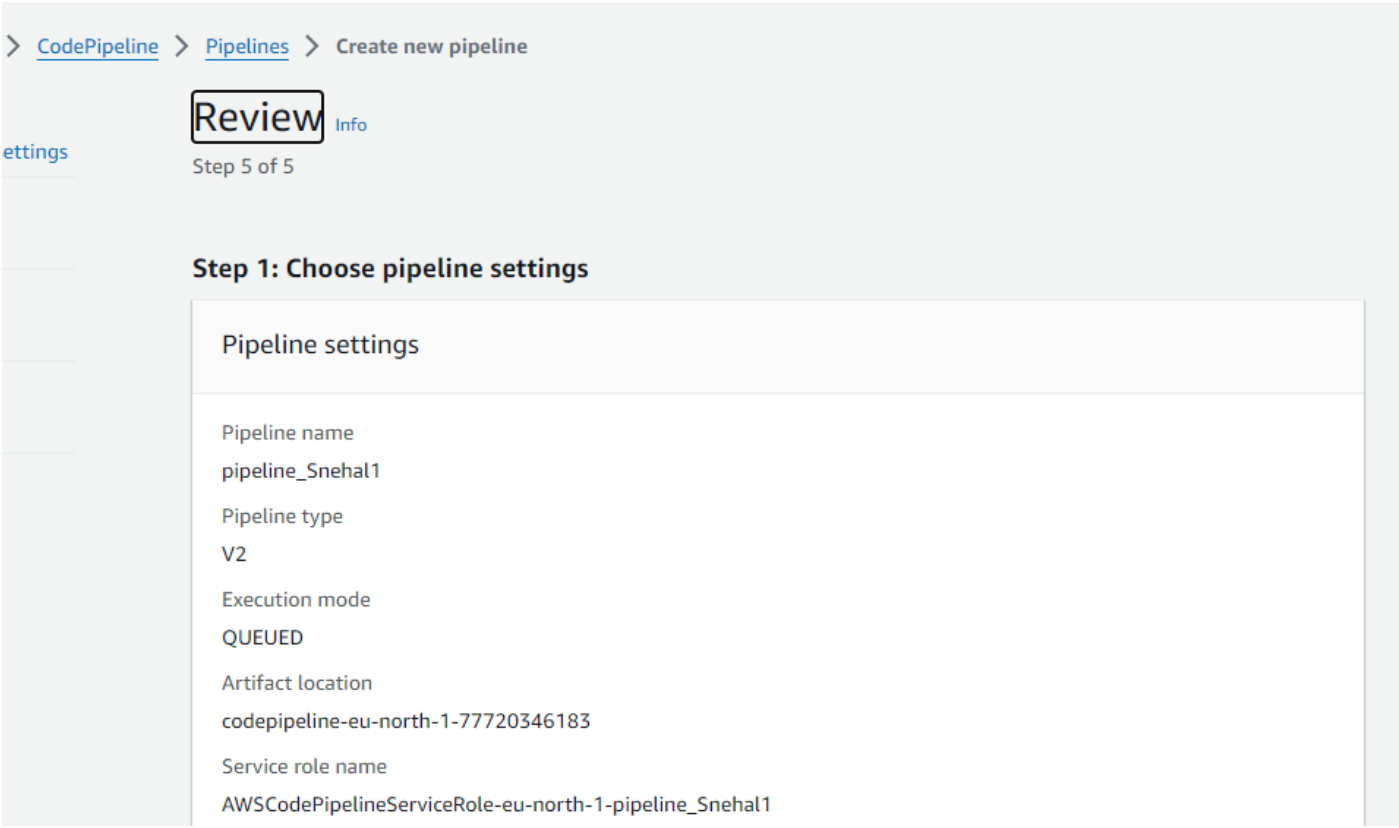
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
Snehal123

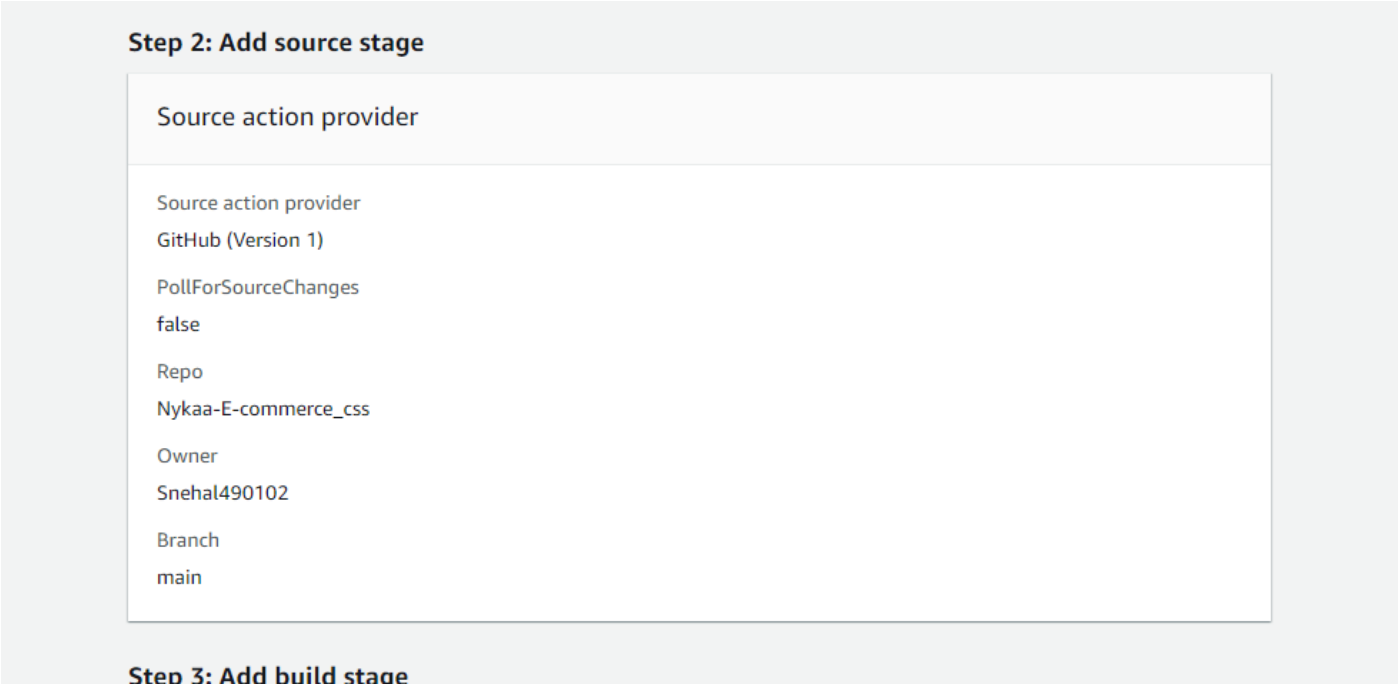
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
Snehal123-env

☐ Configure automatic rollback on stage failure

Cancel Previous Next

Step 4 : review changes and submit





Step 2: Add source stage

Source action provider

Source action provider
GitHub (Version 1)

PollForSourceChanges
false

Repo
Nykaa-E-commerce_css

Owner
Snehal490102

Branch
main

Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
Snehal123

EnvironmentName
Snehal123-env

Configure automatic rollback on stage failure
Disabled

Cancel | Previous | Create pipeline

pipeline_Snehal

Notify | Edit | Stop execution | Clone pipeline | Release change

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: bd6f8320-0a79-412d-9825-256bb1ec64d0

Source

GitHub (Version 1)

Succeeded - Just now

b8307259

View details

b8307259 Source: Update style.css

Disable transition

Deploy In progress

Disable transition

Deploy In progress

Pipeline execution ID: bd6f8320-0a79-412d-9825-256bb1ec64d0

Deploy

AWS Elastic Beanstalk

In progress - Just now

View details

b8307259 Source: Update style.css

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.

Snehal123

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.

Snehal123-Env

Configure automatic rollback on stage failure

Step 6 : Check the deployed website at beanstalk link