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:

With Elastic Beanstalk you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Amazon Web Services (AWS) comprises over one hundred services, each of which exposes an area of functionality. While the variety of services offers flexibility for how you want to manage your AWS infrastructure, it can be challenging to figure out which services to use and how to provision them. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. Elastic Beanstalk also supports Docker platforms. With Docker containers you can choose your own programming language and application dependencies that may not be supported by the other Elastic Beanstalk platforms. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.

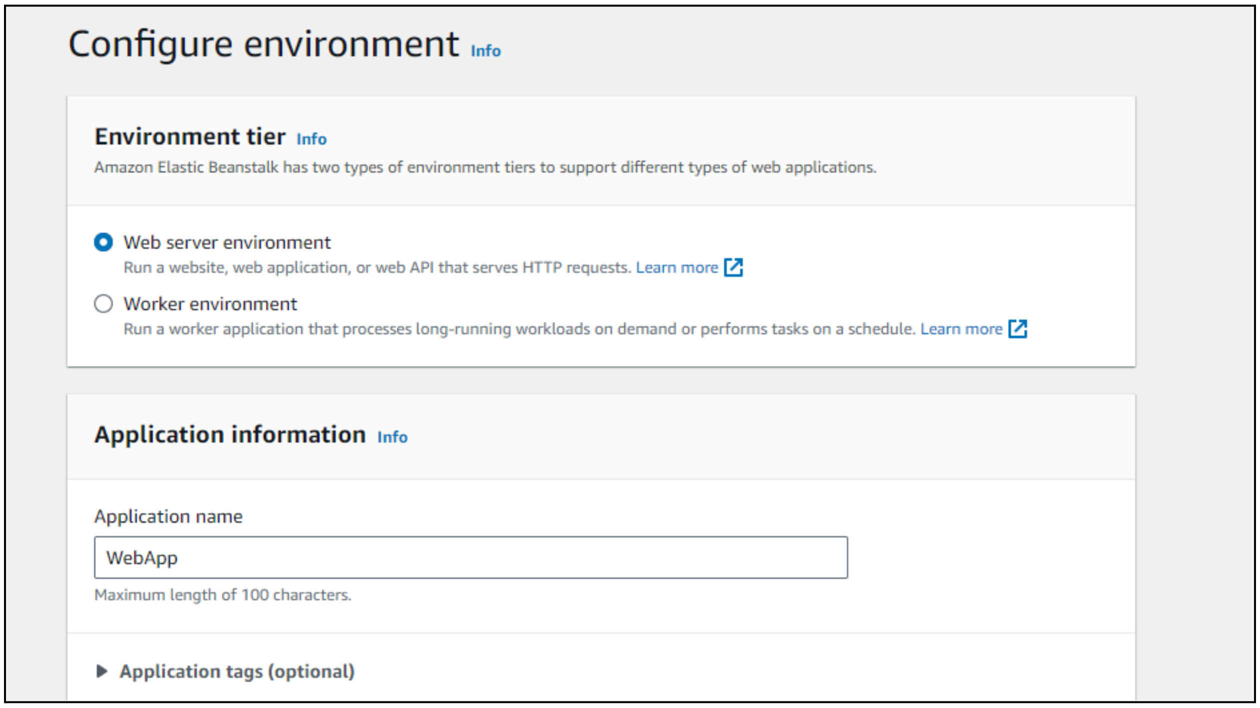
You can interact with Elastic Beanstalk by using the Elastic Beanstalk console, the AWS Command Line Interface (AWS CLI), or eb, a high-level CLI designed specifically for Elastic Beanstalk.

You can also perform most deployment tasks, such as changing the size of your fleet of Amazon EC2 instances or monitoring your application, directly from the Elastic Beanstalk web interface (console). To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application. Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code. After your environment is launched, you can then manage your environment and deploy new application versions. The following diagram illustrates the workflow of Elastic Beanstalk.

After you create and deploy your application, information about the application—including metrics, events, and environment status—is available through the Elastic Beanstalk console, APIs, or Command Line Interfaces, including the unified AWS CLI.

Elastic Beanstalk

Step 1: create environment



Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

● Web server environment
Run a website, web application, or web API that serves HTTP requests. Learn more

○ Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more

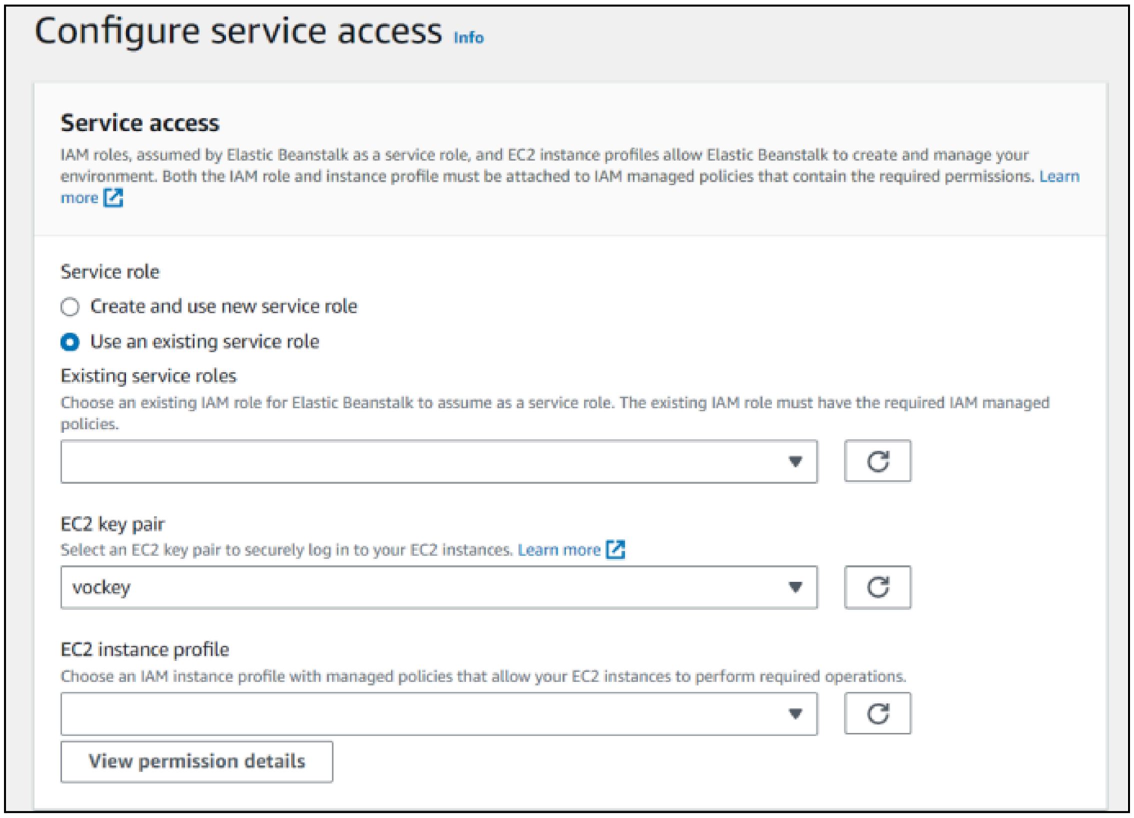
Application information Info

Application name: WebApp

Maximum length of 100 characters.

▶ Application tags (optional)

Step 2 : add your Ec2 key pair and instance profile



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.

EC2 key pair

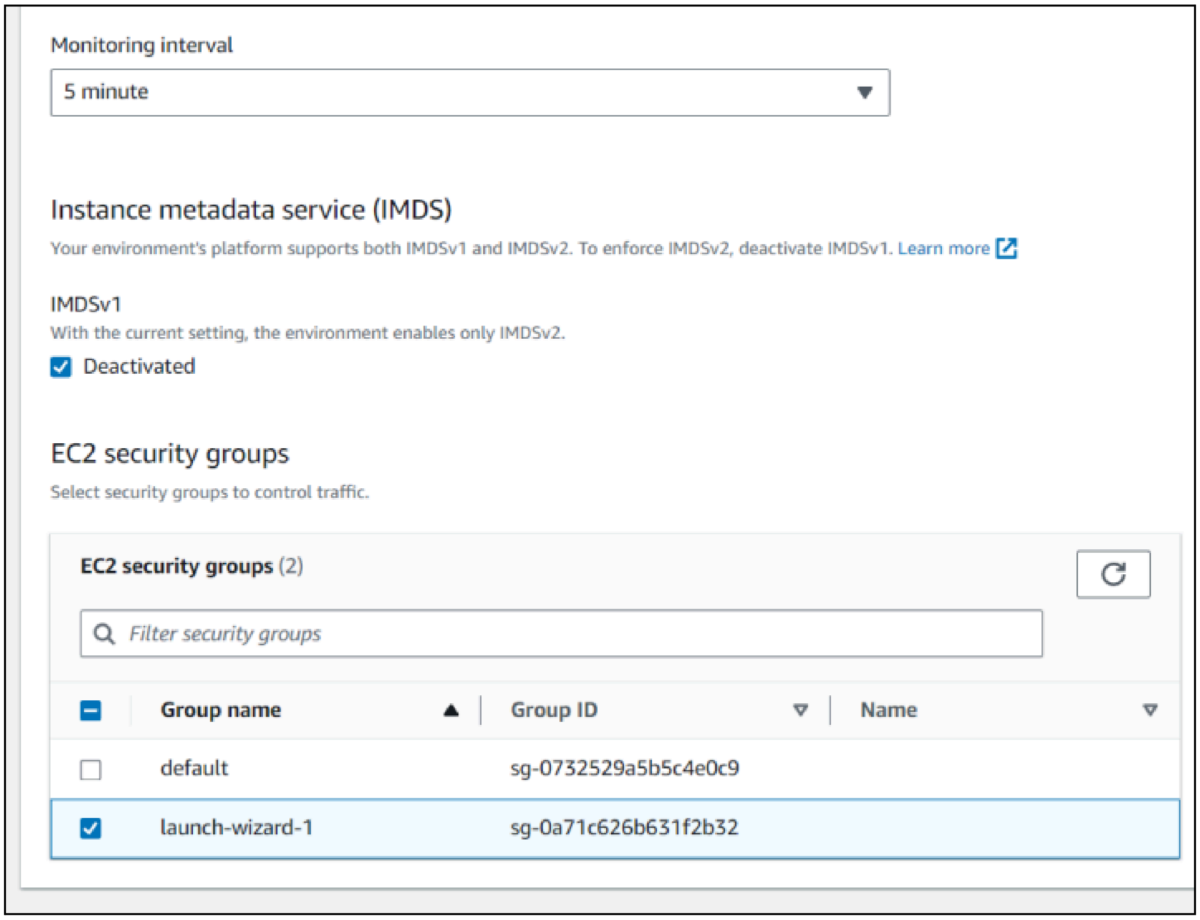
Select an EC2 key pair to securely log in to your EC2 instances. Learn more

vockey

EC2 instance profile

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

Step 3 : add security config and review all settings



Monitoring interval

5 minute

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. Learn more

IMDSv1

With the current setting, the environment enables only IMDSv2.

☑ Deactivated

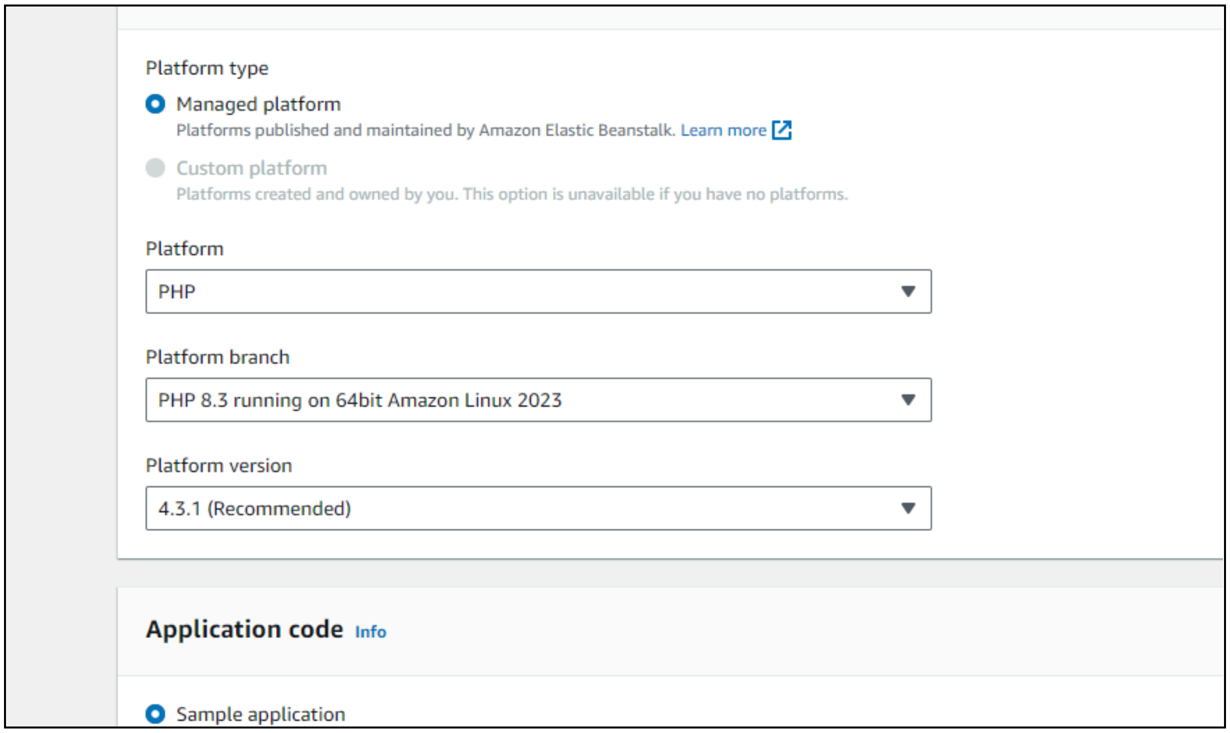
EC2 security groups

Select security groups to control traffic.

EC2 security groups (2)

Filter security groups

	Group name	Group ID	Name
☐	default	sg-0732529a5b5c4e0c9	
☑	launch-wizard-1	sg-0a71c626b631f2b32	


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.

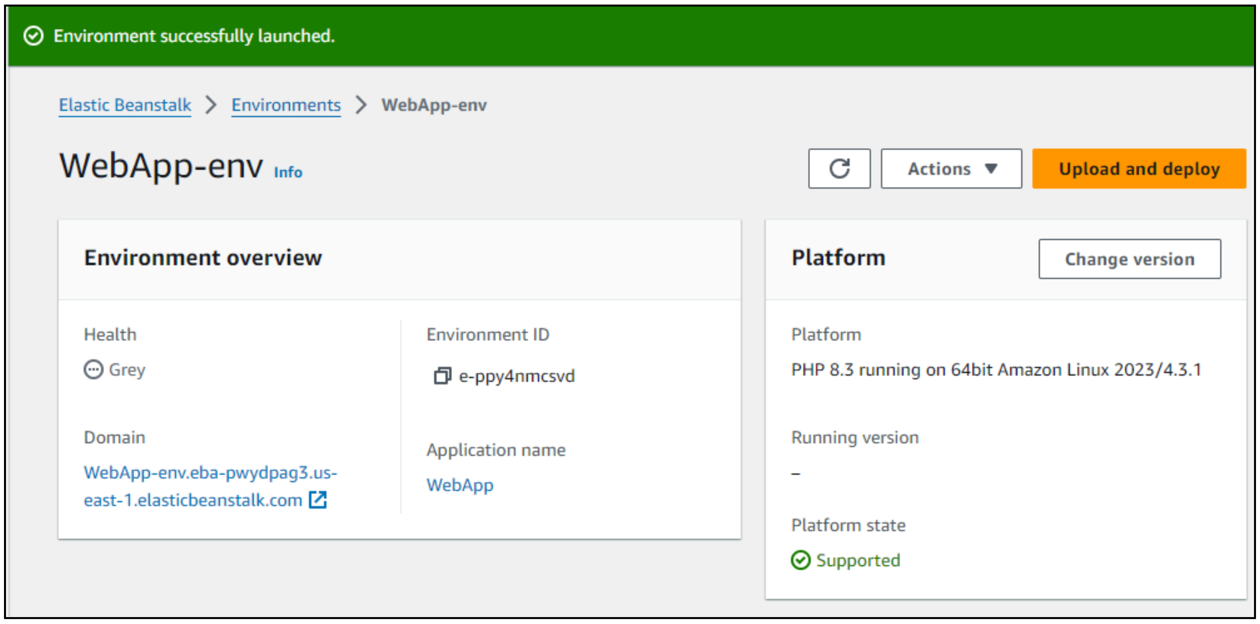
Platform
PHP

Platform branch
PHP 8.3 running on 64bit Amazon Linux 2023

Platform version
4.3.1 (Recommended)

Application code Info

● Sample application


Environment successfully launched.

Elastic Beanstalk > Environments > WebApp-env

WebApp-env Info

Environment overview

Health	Environment ID
Grey	e-ppy4nmcsvd
Domain	Application name
WebApp-env.eba-pwydpag3.us-east-1.elasticbeanstalk.com	WebApp

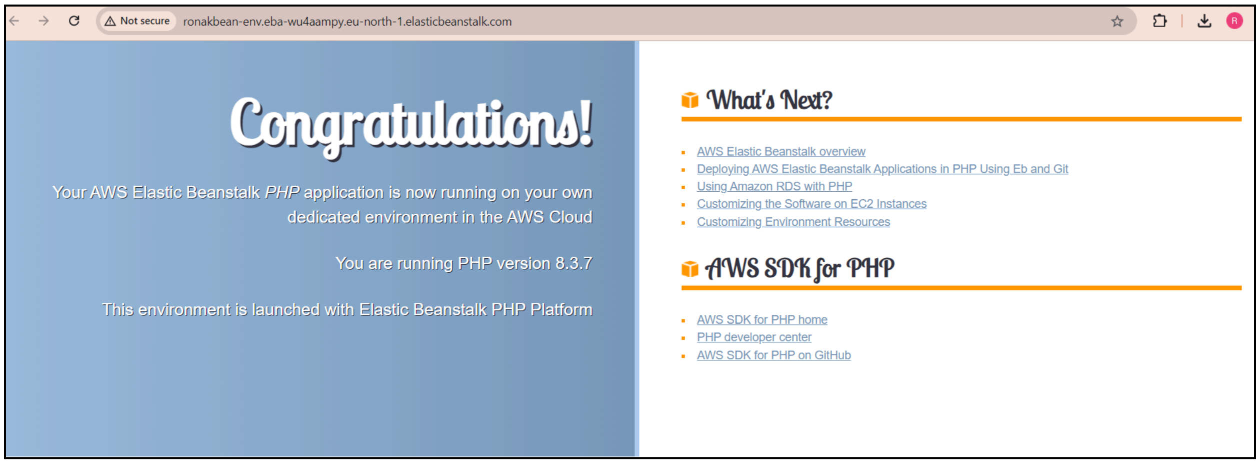
Platform

Platform
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1

Running version
–

Platform state
Supported

Step 4 : Beanstalk environment is created



Not secure ronakbean-env.eba-wu4aampy.eu-north-1.elasticbeanstalk.com

Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 8.3.7

This environment is launched with Elastic Beanstalk PHP Platform

What's Next?

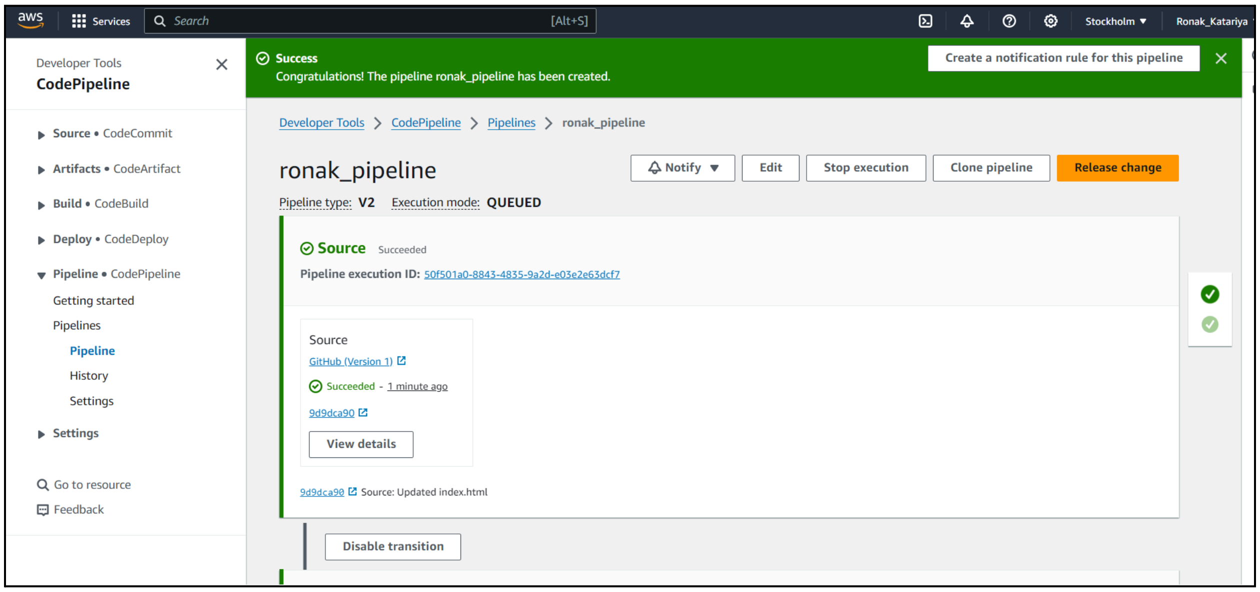
- AWS Elastic Beanstalk overview
- Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
- Using Amazon RDS with PHP
- Customizing the Software on EC2 Instances
- Customizing Environment Resources

AWS SDK for PHP

- AWS SDK for PHP home
- PHP developer center
- AWS SDK for PHP on GitHub

Pipeline Creation:

Step 1 : click on create pipeline and give name



Success

Congratulations! The pipeline ronak_pipeline has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > ronak_pipeline

ronak_pipeline

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

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: 50f501a0-8843-4835-9a2d-e03e2e63dcf7

Source

GitHub (Version 1)

Succeeded - 1 minute ago

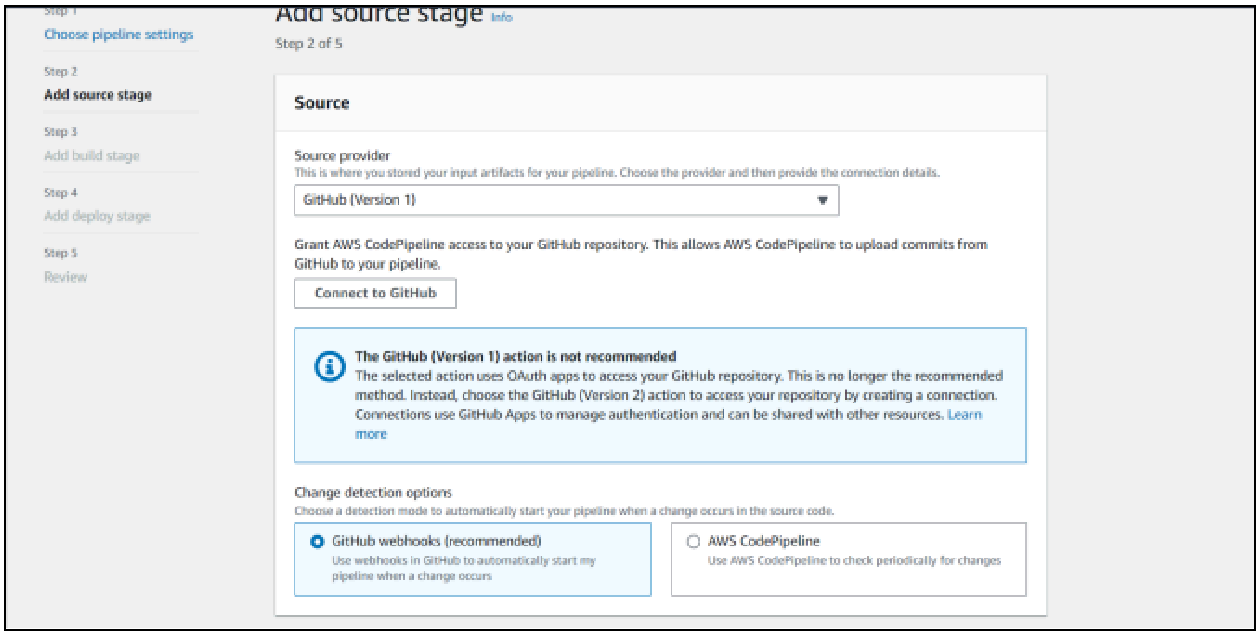
9d9dca90

View details

9d9dca90 Source: Updated index.html

Disable transition

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



Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Add source stage Info
Step 2 of 5

Source

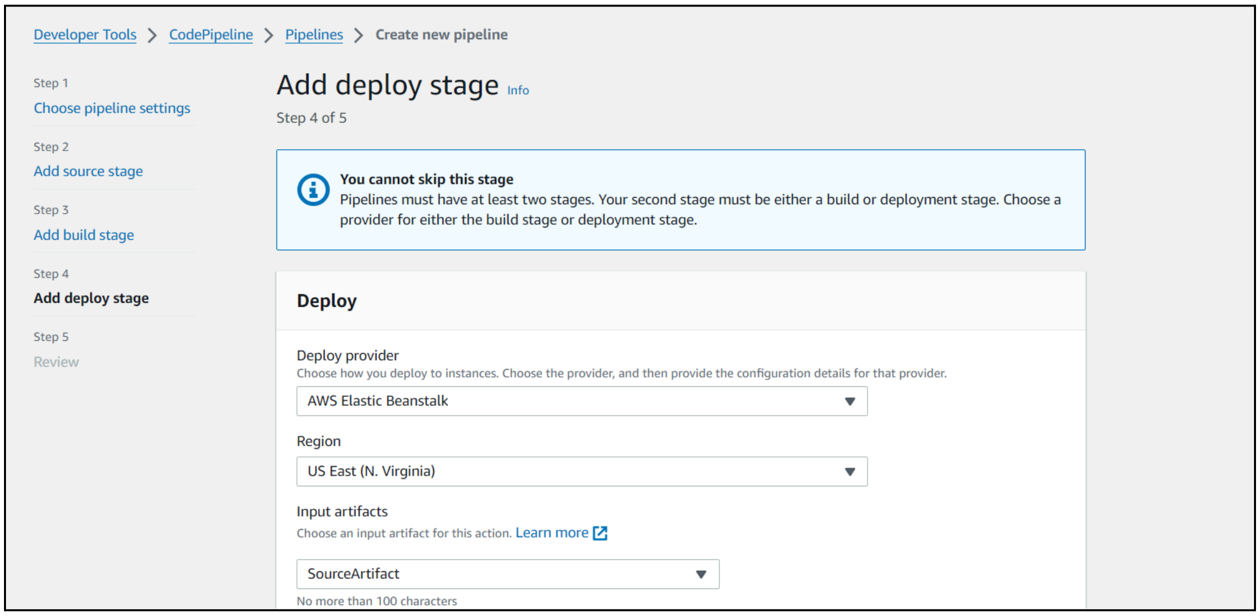
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 1)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.
Connect to GitHub

The GitHub (Version 1) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
[✓ GitHub webhooks (recommended)] Use webhooks in GitHub to automatically start my pipeline when a change occurs
AWS CodePipeline: Use AWS CodePipeline to check periodically for changes

Step 3 : Add deploy config choosing the elastic beanstalk



Developer Tools > CodePipeline > Pipelines > Create new pipeline

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.

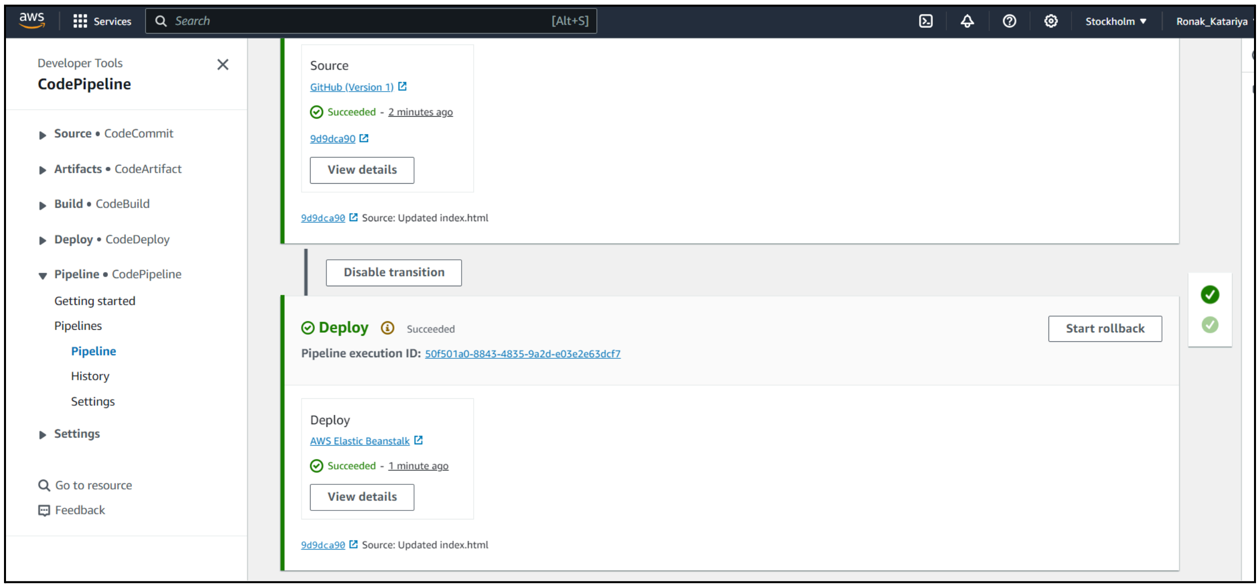
Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk

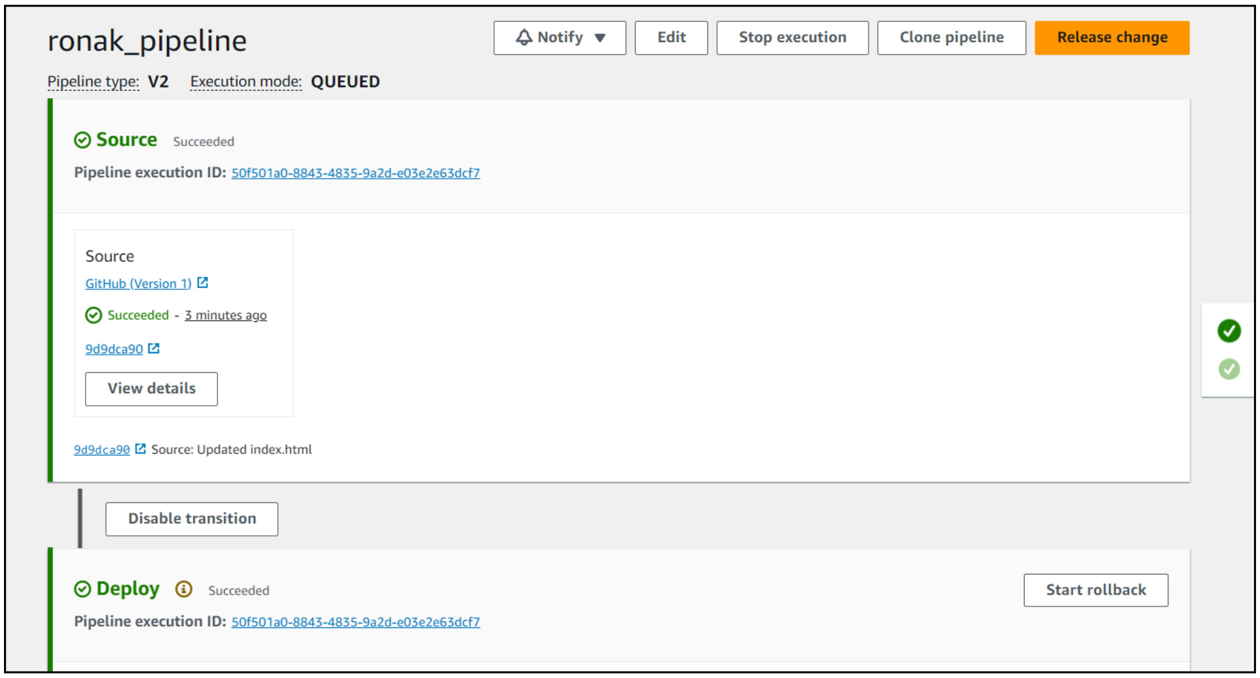
Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. Learn more
SourceArtifact
No more than 100 characters

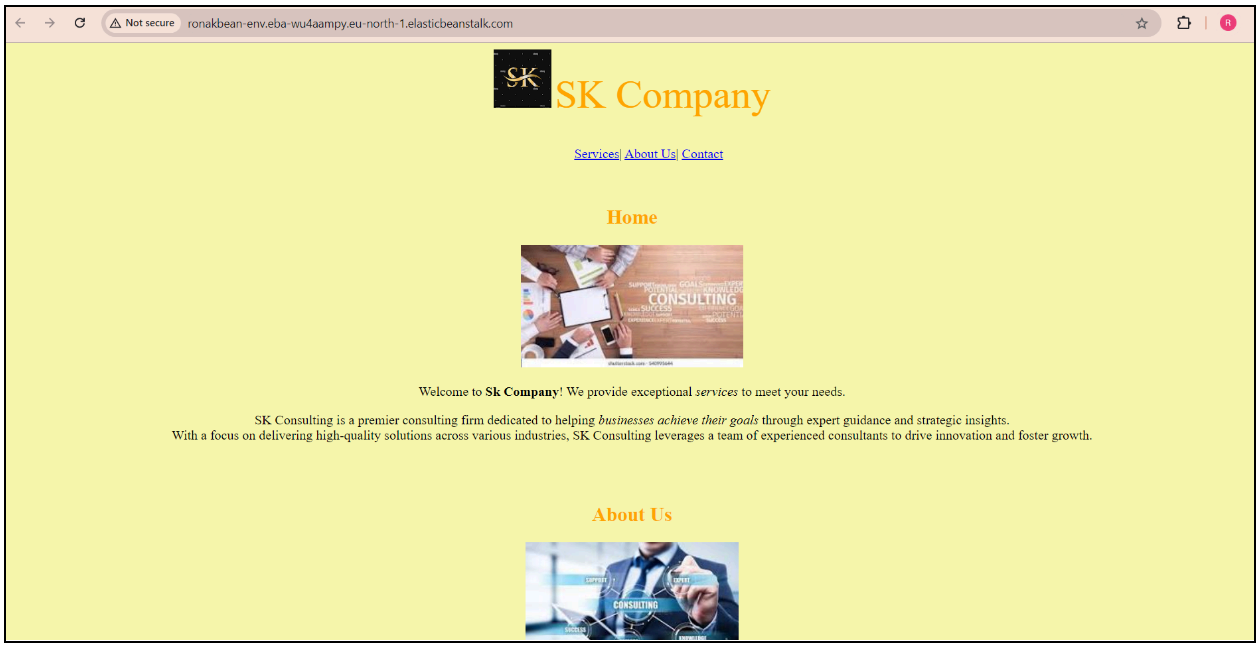
Step 4 : review changes and submit


Source
GitHub (Version 1)
Succeeded - 2 minutes ago
9d9dca90
View details
9d9dca90 Source: Updated index.html
Disable transition
Deploy Succeeded
Pipeline execution ID: 50f501a0-8843-4835-9a2d-e03e2e63dcf7
Start rollback
Deploy
AWS Elastic Beanstalk
Succeeded - 1 minute ago
View details
9d9dca90 Source: Updated index.html

Step 5 : view the pipeline build and deployment


ronak_pipeline
Pipeline type: V2 Execution mode: QUEUED
Source Succeeded
Pipeline execution ID: 50f501a0-8843-4835-9a2d-e03e2e63dcf7
Source
GitHub (Version 1)
Succeeded - 3 minutes ago
9d9dca90
View details
9d9dca90 Source: Updated index.html
Disable transition
Deploy Succeeded
Pipeline execution ID: 50f501a0-8843-4835-9a2d-e03e2e63dcf7
Start rollback

Step 6 : Check the deployed website at beanstalk link



SK Company

Services| About Us| Contact

Home

Welcome to **Sk Company**! We provide exceptional *services* to meet your needs.

SK Consulting is a premier consulting firm dedicated to helping *businesses achieve their goals* through expert guidance and strategic insights.
With a focus on delivering high-quality solutions across various industries, SK Consulting leverages a team of experienced consultants to drive innovation and foster growth.

About Us