Advanced DevOps Lab
Experiment 2

Name: Yash Rahate
Class: D15B
Roll No.: 48

Aim:

To build your application using AWS CodeBuild and deploy it on S3 or Elastic Beanstalk (SEBS) using AWS CodePipeline, and deploy a sample application on an EC2 instance using AWS CodeDeploy.

Theory:

Cloud-based services from AWS enable developers to automate their build, testing, and deployment processes using **Continuous Integration (CI)** and **Continuous Deployment (CD)** principles. AWS provides services like CodeBuild, CodePipeline, and CodeDeploy to achieve automation for development teams. Here's an in-depth look at these services and how they work together.

1. **AWS CodeBuild**: AWS CodeBuild is a fully managed build service that allows developers to compile source code, run tests, and produce deployable artifacts. It helps to create an automated pipeline for building and testing applications.

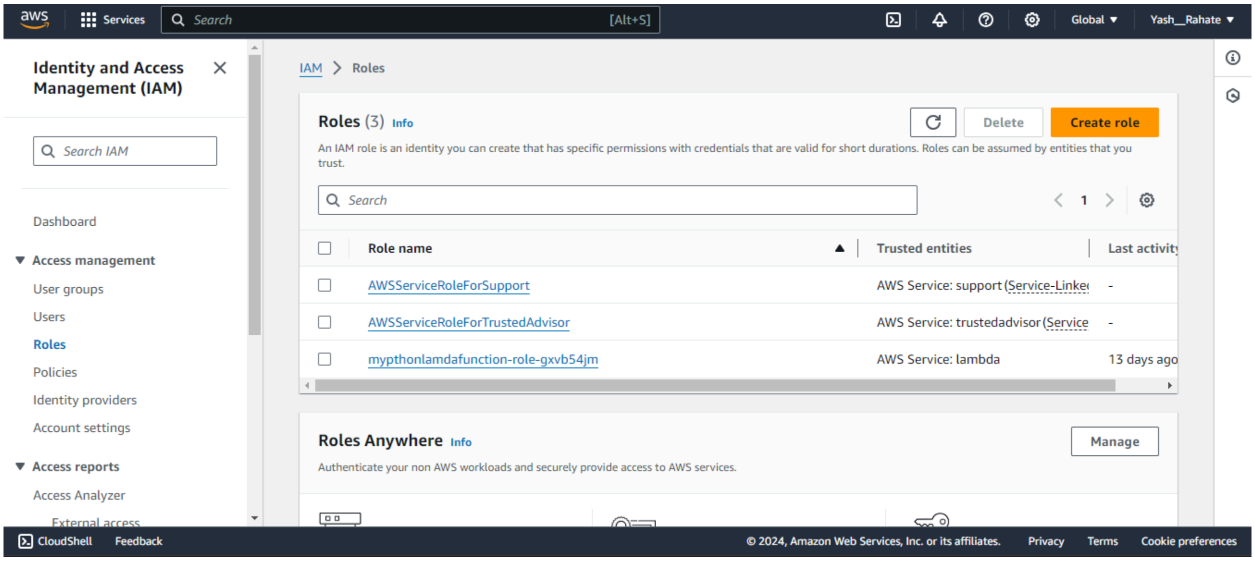
2. **AWS CodePipeline**: AWS CodePipeline automates the release process by creating pipelines that model the stages of software delivery, such as **source**, **build**, **test**, and **deploy**. CodePipeline integrates with CodeBuild to automate builds and with various deployment targets like S3, Elastic Beanstalk, CodeDeploy, etc.

3. **AWS CodeDeploy**: AWS CodeDeploy is a deployment service that automates the deployment of applications to **EC2 instances**, **Lambda functions**, or **on-premises**

servers. It allows developers to release new features, update applications, and minimize downtime during deployments

Steps to Build and Deploy using AWS CodeBuild, CodePipeline, and CodeDeploy:

Create a role in an IAM.



Roles (3) Info

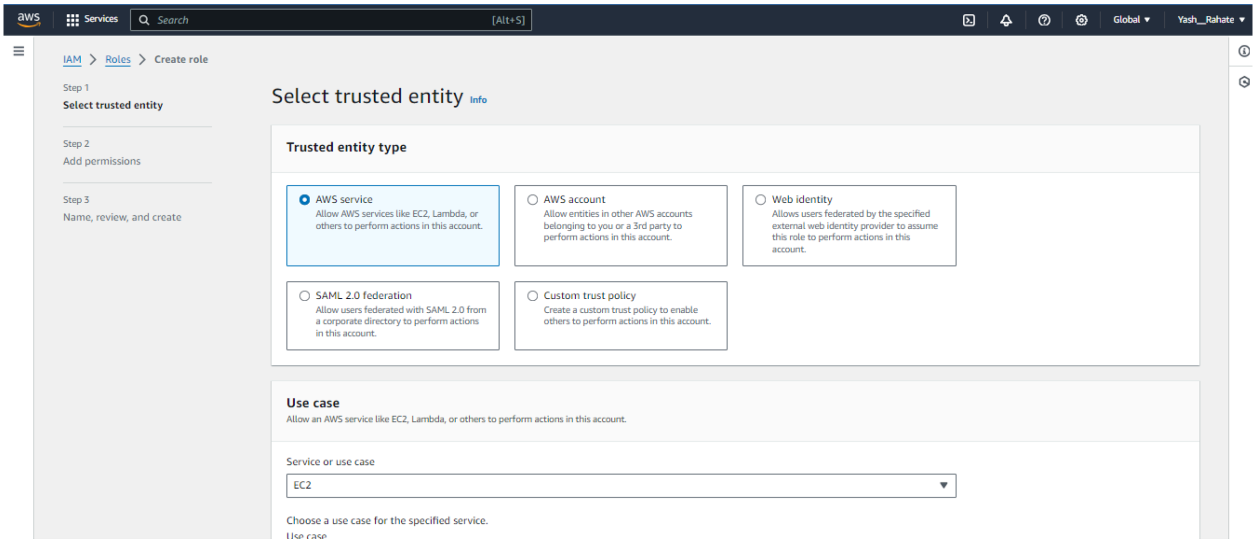
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name ▲	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	mypthonlamdafunction-role-gxvb54jm	AWS Service: lambda	13 days ago

Roles Anywhere Info

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

Add EC2 for a service or use case.



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

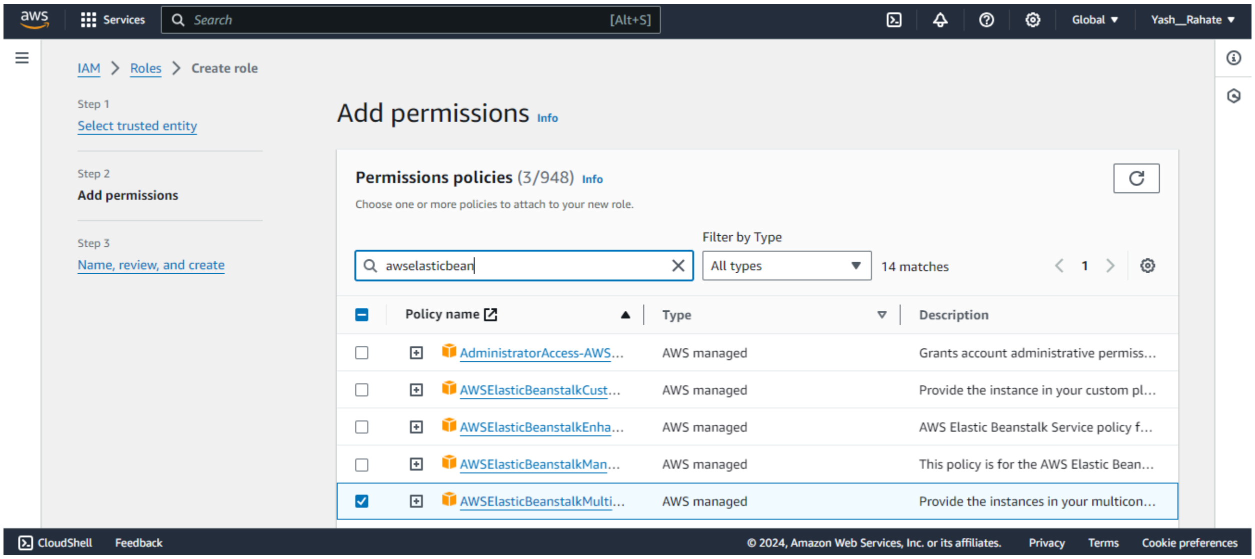
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case: EC2

Choose a use case for the specified service.

Use case

Give name to the role.



Add permissions Info

Permissions policies (3/948) Info

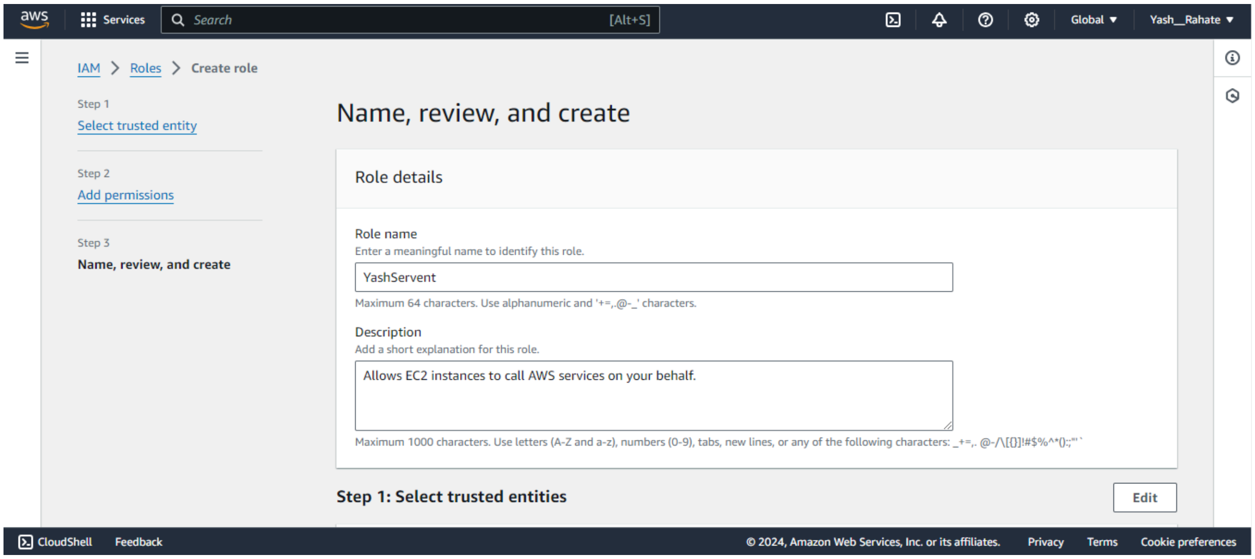
Choose one or more policies to attach to your new role.

Search: awselasticbean

Filter by Type: All types

14 matches

	Policy name	Type	Description
□	AdministratorAccess-AWS…	AWS managed	Grants account administrative permiss…
□	AWSElasticBeanstalkCust…	AWS managed	Provide the instance in your custom pl…
□	AWSElasticBeanstalkEnha…	AWS managed	AWS Elastic Beanstalk Service policy f…
□	AWSElasticBeanstalkMan…	AWS managed	This policy is for the AWS Elastic Bean…
☑	AWSElasticBeanstalkMulti…	AWS managed	Provide the instances in your multicon…



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

YashServent

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description

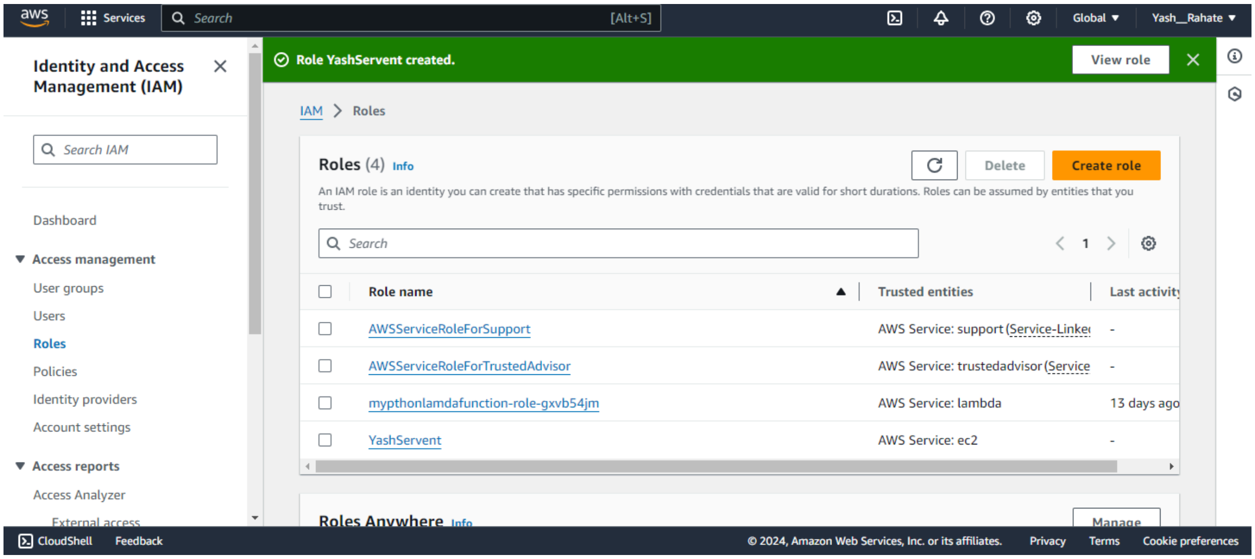
Add a short explanation for this role.

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

IAM role is being created



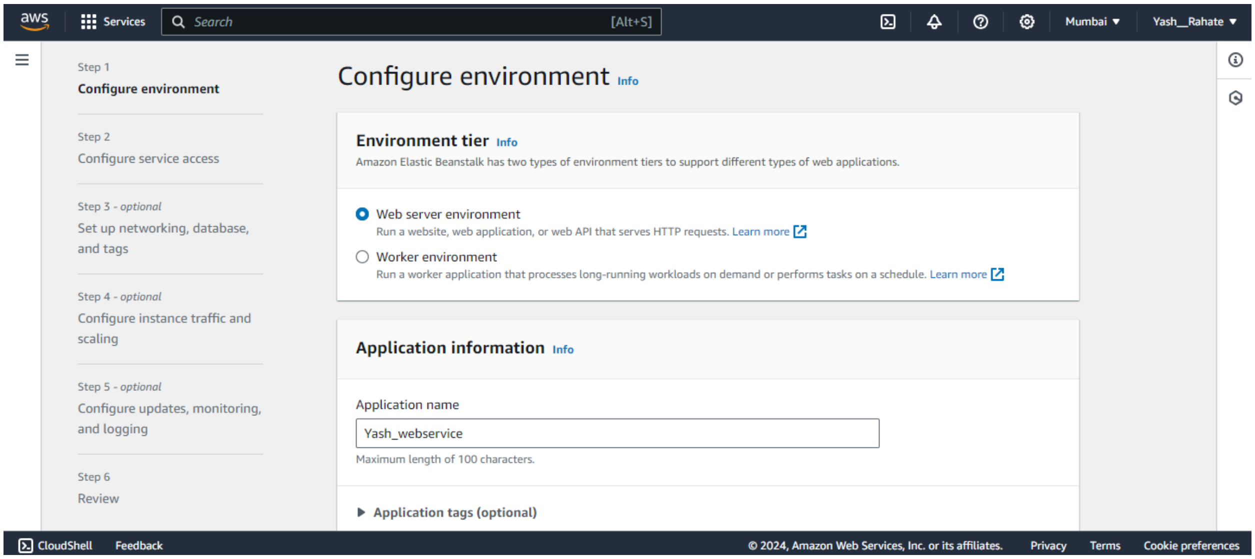
Role YashServent created.

Roles (4)

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.

	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	mypthonlamdafunction-role-gxvb54jm	AWS Service: lambda	13 days ago
□	YashServent	AWS Service: ec2	-

Go to the Elastic beanstalk and create an application. Give the appropriate name for the application.



Configure environment

Environment tier

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.

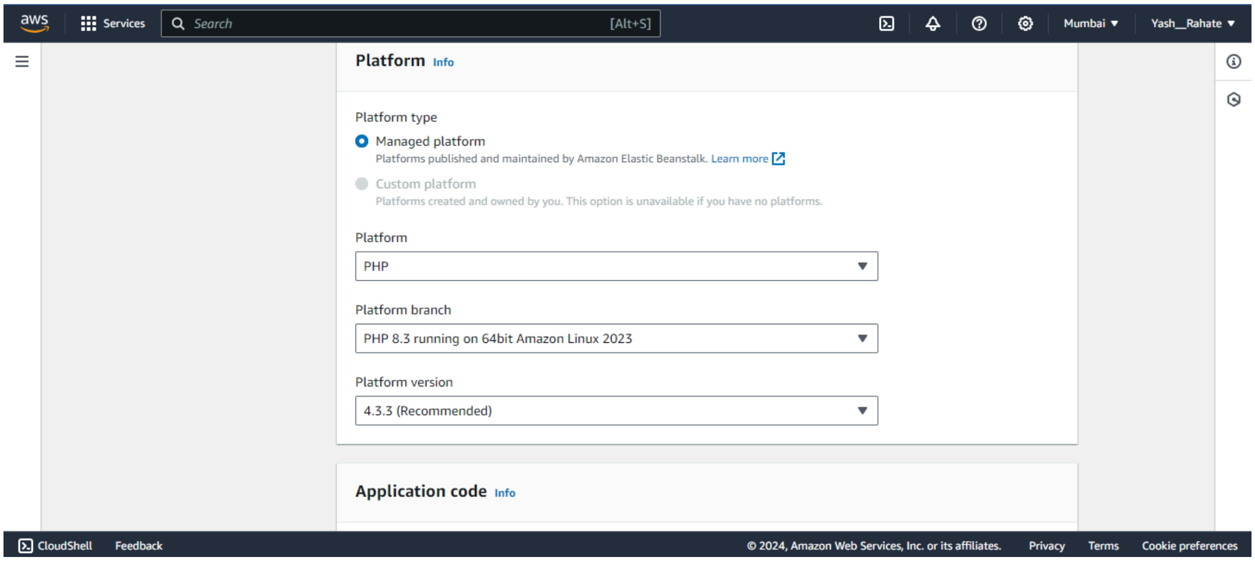
○ Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.

Application information

Application name: Yash_webservice

Maximum length of 100 characters.

Select the platform as PHP.



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.

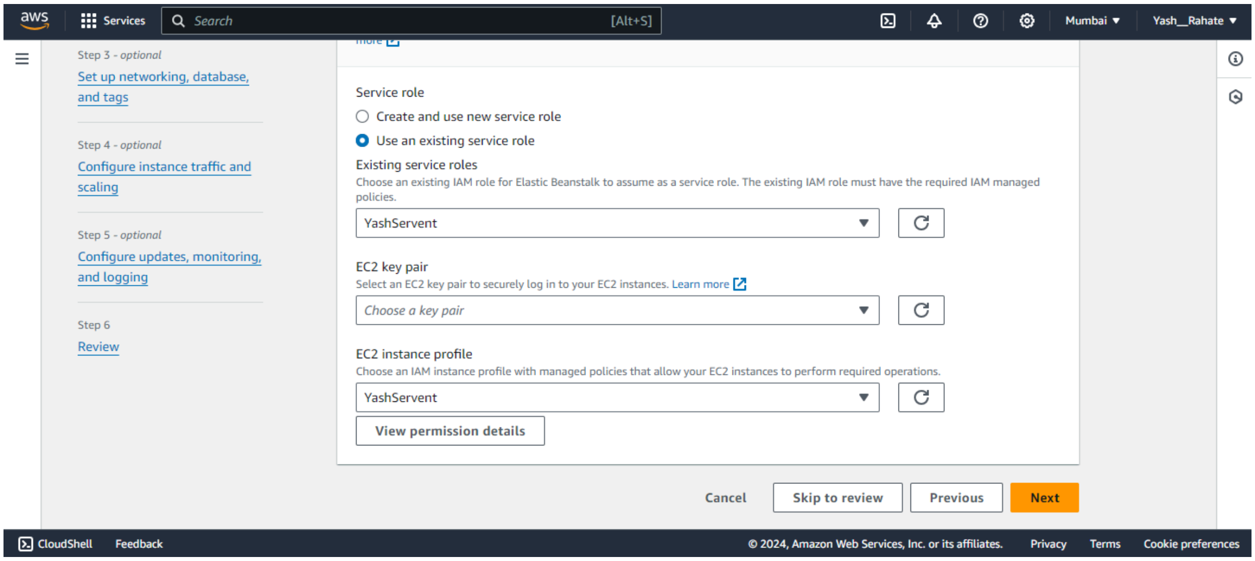
Platform: PHP

Platform branch: PHP 8.3 running on 64bit Amazon Linux 2023

Platform version: 4.3.3 (Recommended)

Application code Info

In an ec2 instance profile, select the created IAM role.



Service role

○ Create and use new service role

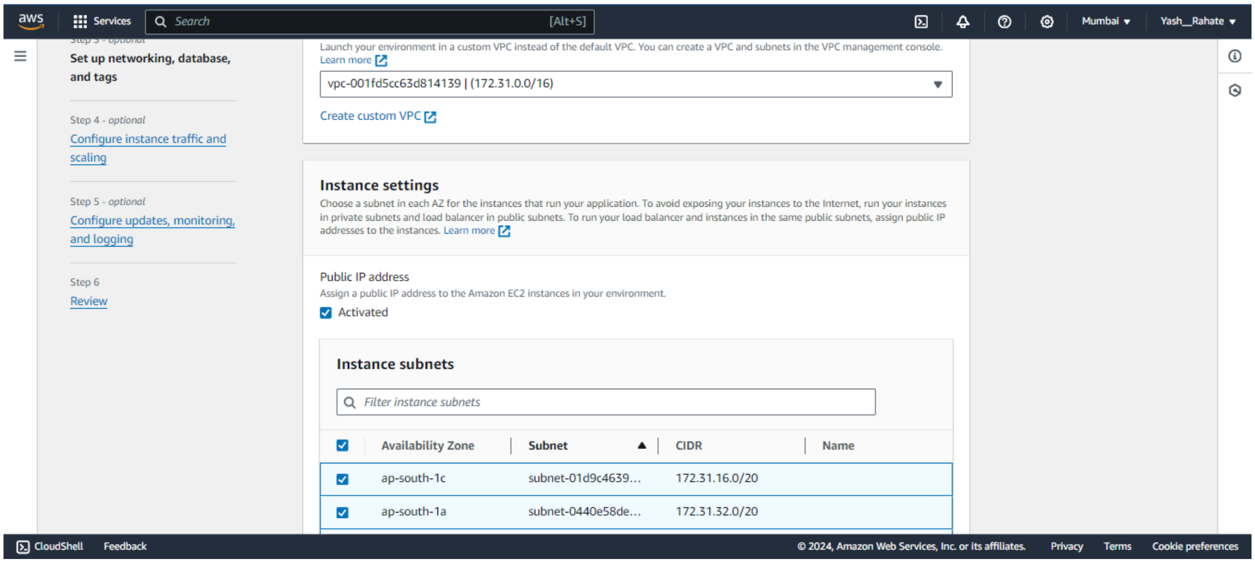
● Use an existing service role

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.
YashServent

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.
YashServent

Vpc is to be selected. Public IP address and availability is to be checked.



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

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-001fd5cc63d814139 | (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

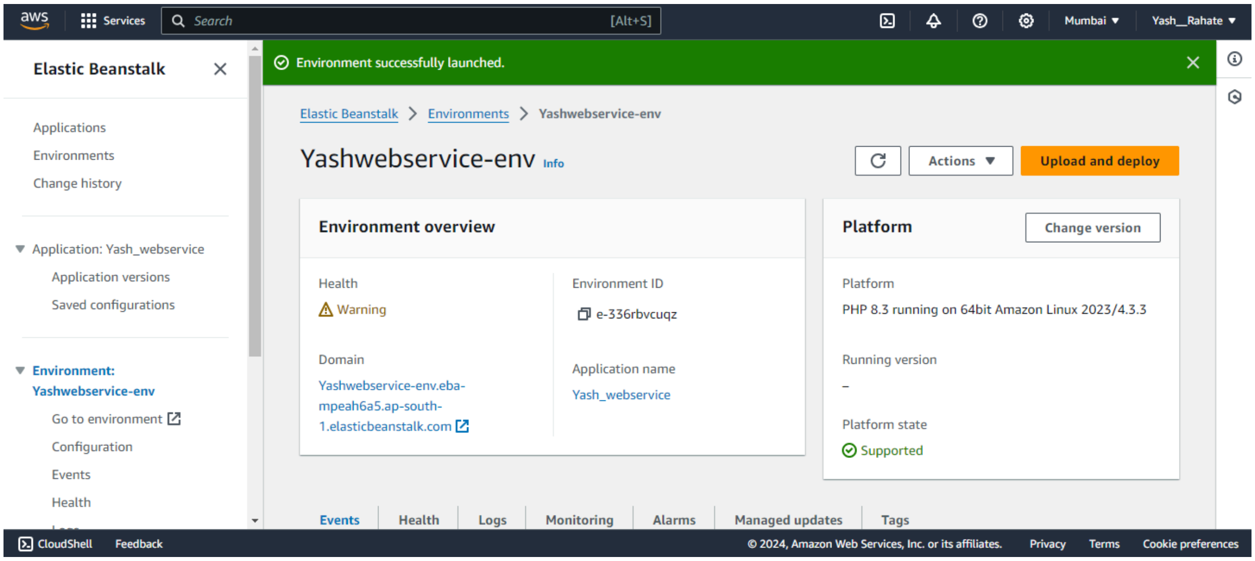
Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.
☑ Activated

Instance subnets

Filter instance subnets

☑	Availability Zone	Subnet ▲	CIDR	Name
☑	ap-south-1c	subnet-01d9c4639...	172.31.16.0/20	
☑	ap-south-1a	subnet-0440e58de...	172.31.32.0/20	

Environment is launched successfully.



Environment successfully launched.

Elastic Beanstalk > Environments > Yashwebservice-env

Yashwebservice-env Info

Environment overview

Health
⚠ Warning

Environment ID
e-336rbvcuqz

Domain
Yashwebservice-env.eba-mpeah6a5.ap-south-1.elasticbeanstalk.com

Application name
Yash_webservice

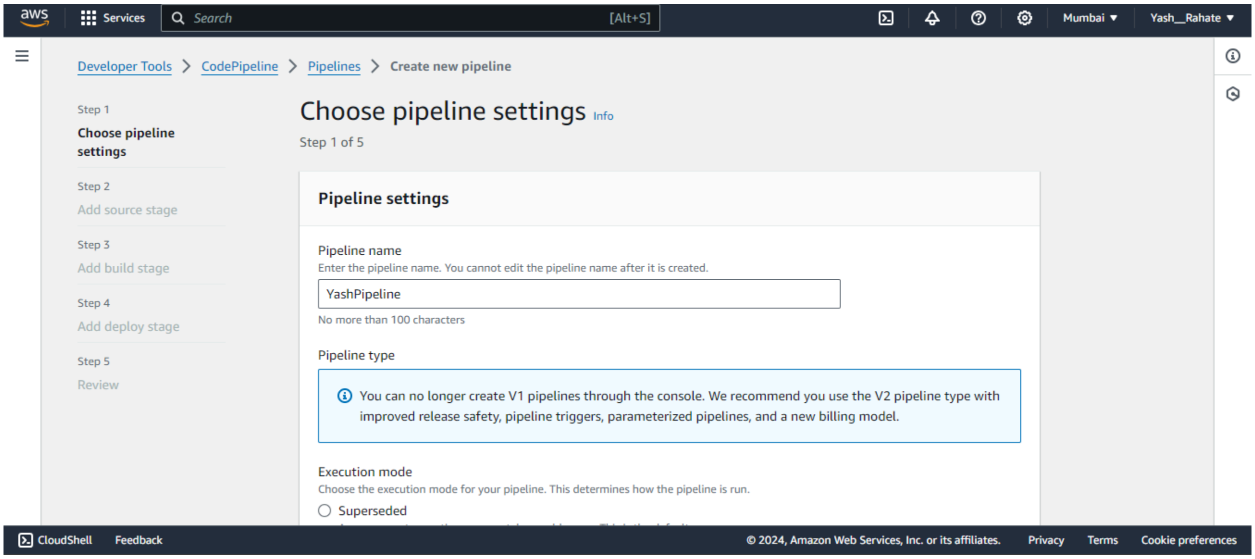
Platform

Platform
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.3

Running version
–

Platform state
✓ Supported

Go to the CodePipeline and select the source as GitHub (version 1).



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.

YashPipeline

No more than 100 characters

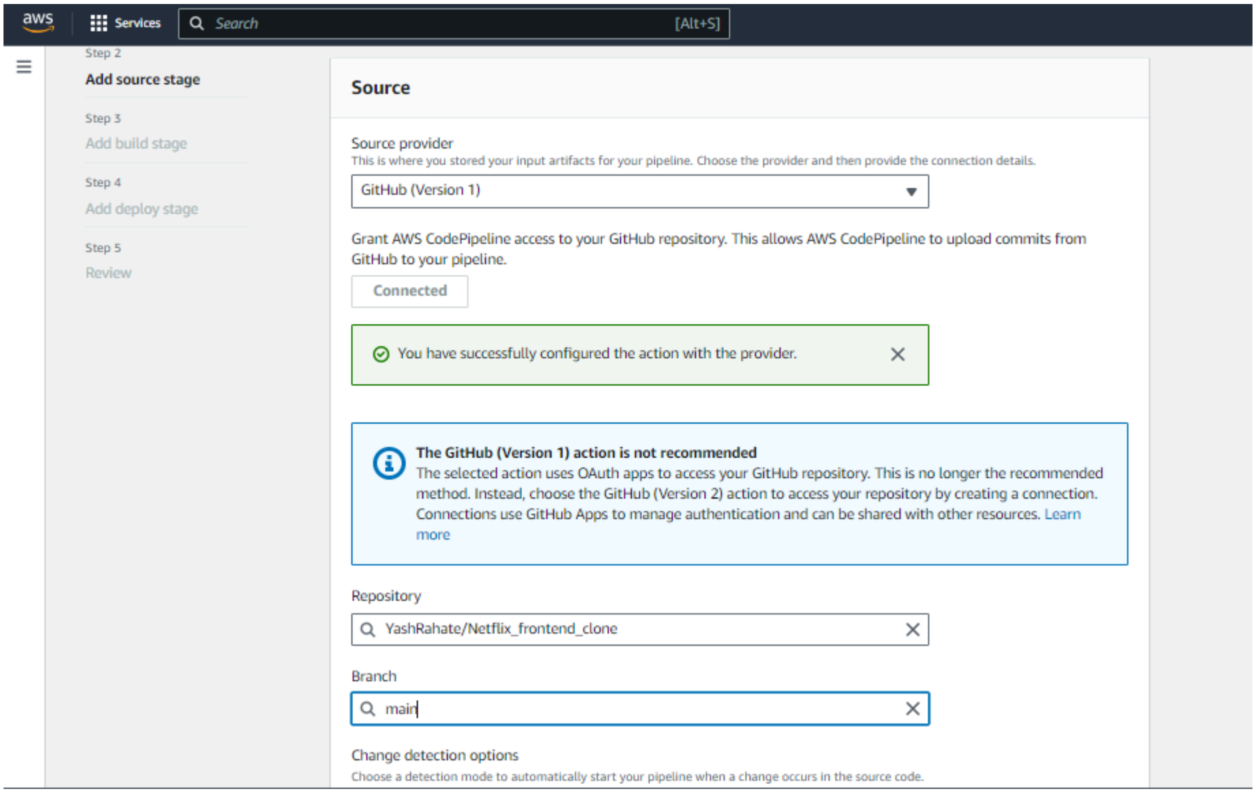
Pipeline type

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



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.

Connected

You have successfully configured the action with the provider.

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

Repository

YashRahate/Netflix_frontend_clone

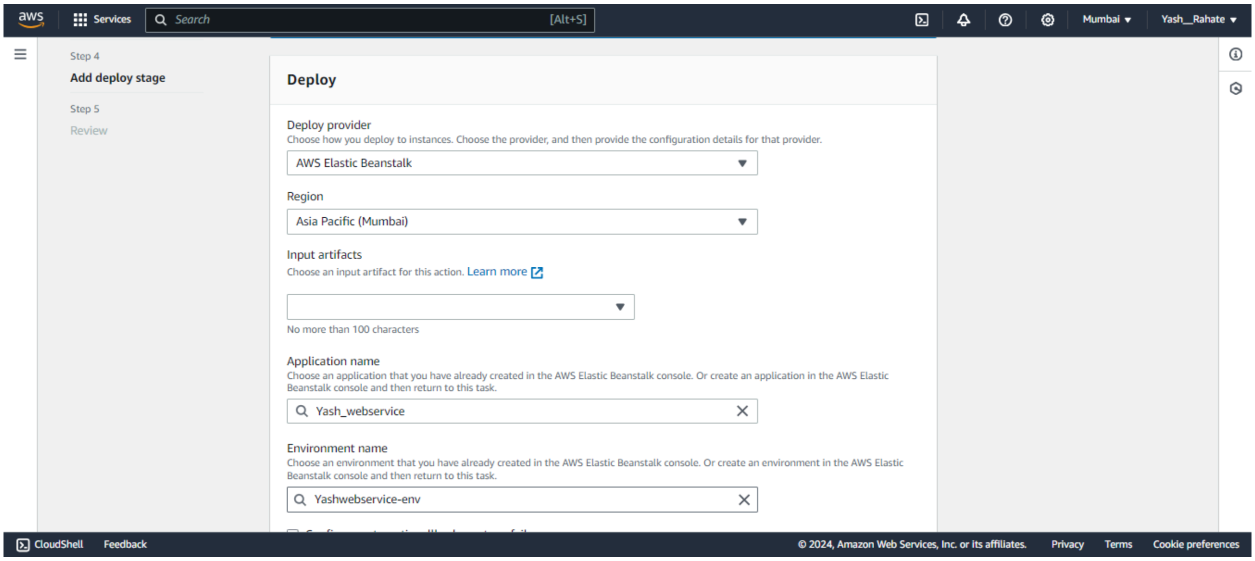
Branch

main

Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

After skipping the build stage, AWS Elastic beanstalk is to be selected in the Deploy Provider. Select your recently created application name and environment name.



Step 4
Add deploy stage
Step 5
Review

Deploy

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

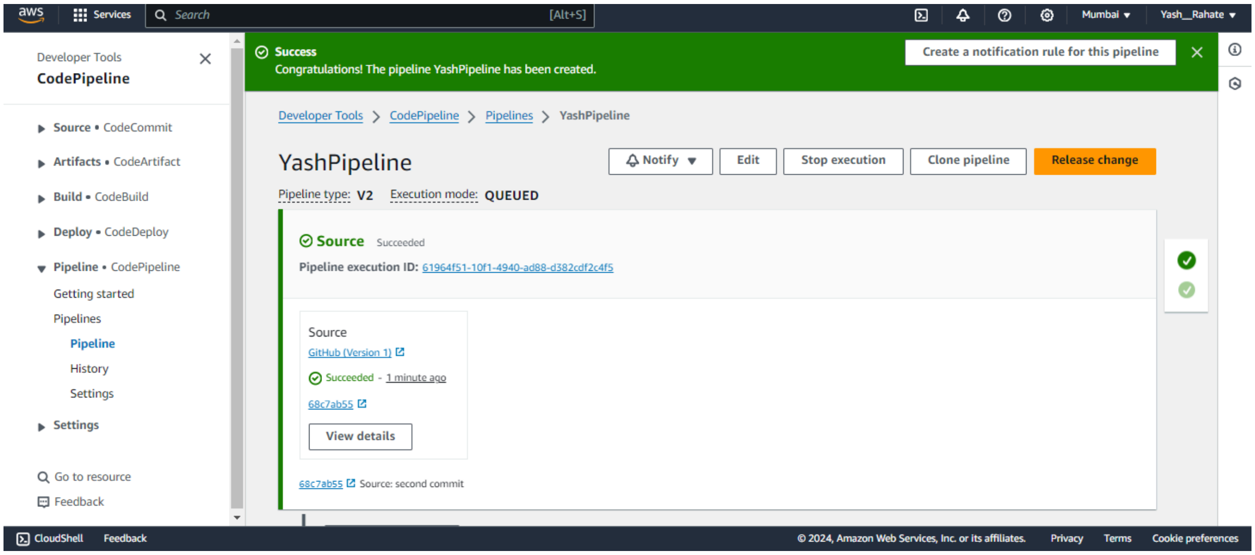
Region
Asia Pacific (Mumbai)

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.
Yash_webservice

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.
Yashwebservice-env

Pipeline is created. Source and Deploy section is also successful.



Success
Congratulations! The pipeline YashPipeline has been created.

Developer Tools > CodePipeline > Pipelines > YashPipeline

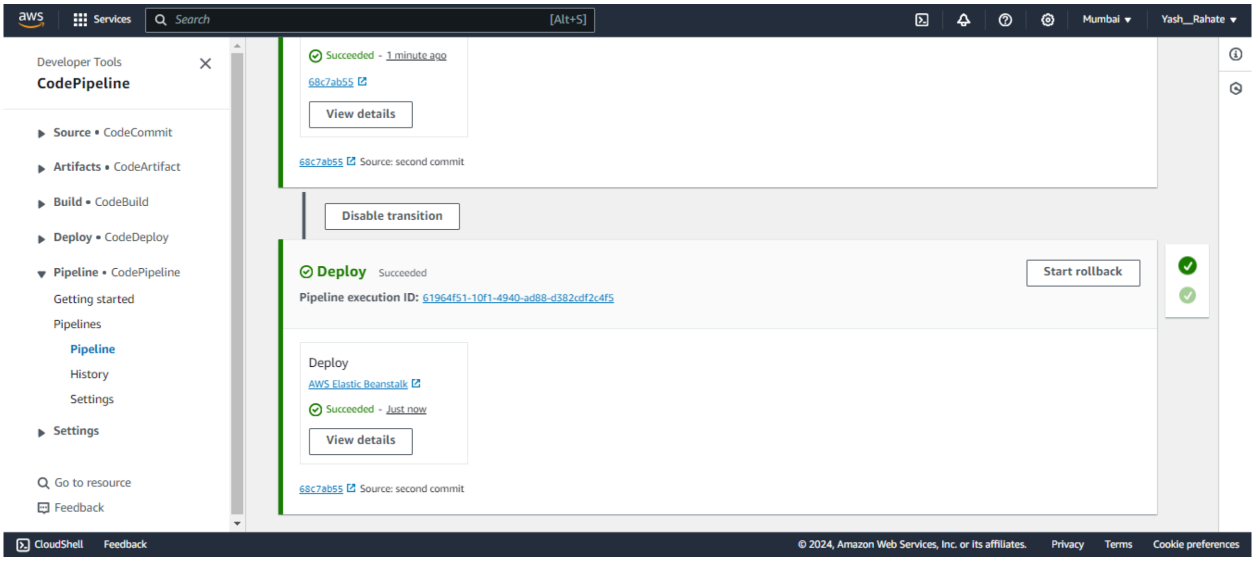
YashPipeline

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: 61964f51-10f1-4940-ad88-d382cdf2c4f5

Source
GitHub (Version 1)
Succeeded - 1 minute ago
68c7ab55

68c7ab55 Source: second commit



✓ Succeeded - 1 minute ago
68c7ab55

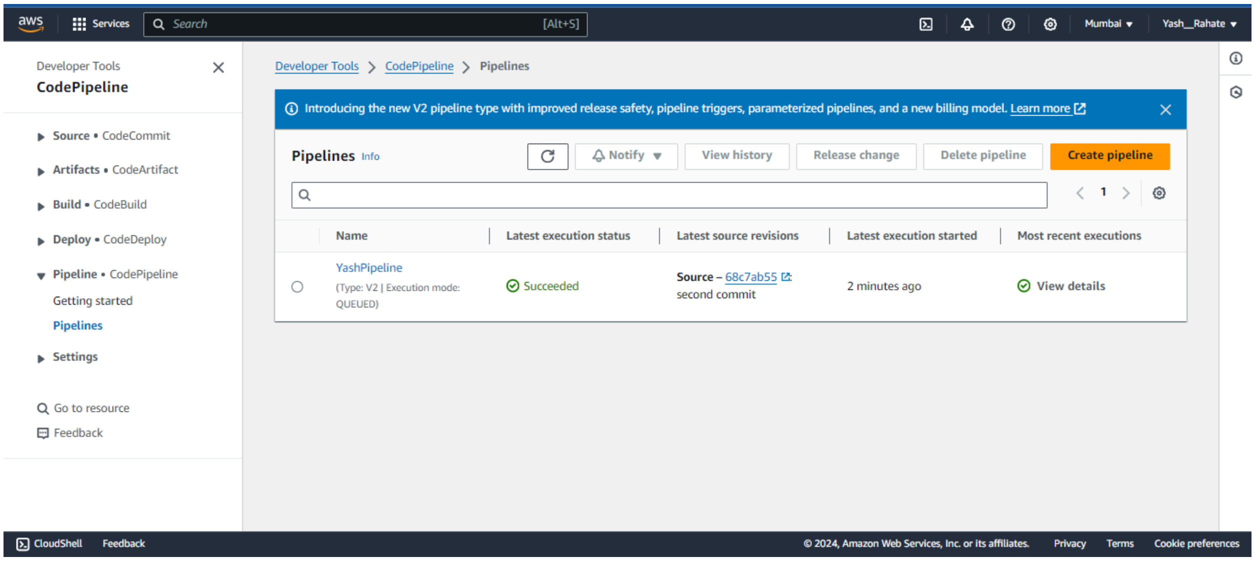
68c7ab55 Source: second commit

✓ Deploy Succeeded
Pipeline execution ID: 61964f51-10f1-4940-ad88-d382cdf2c4f5

Deploy
AWS Elastic Beanstalk
✓ Succeeded - Just now

68c7ab55 Source: second commit

Go to the elastic beanstalk environment and click on domain.



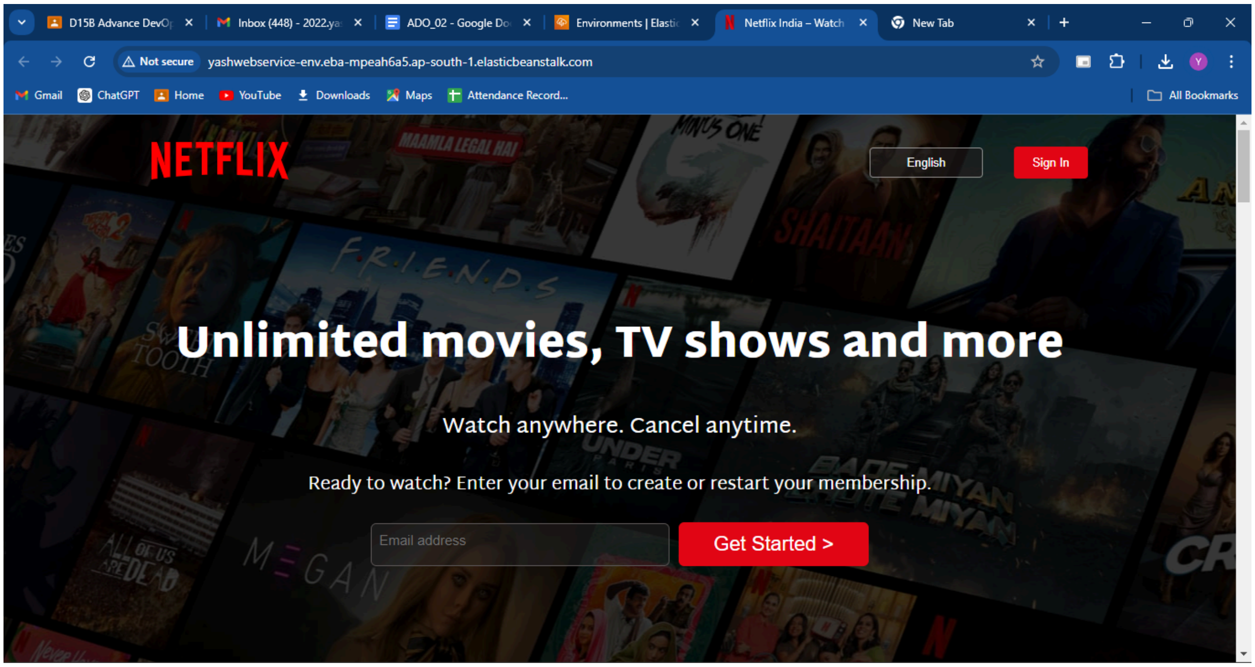
Developer Tools > CodePipeline > Pipelines

Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more

Pipelines Info

	Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
○	YashPipeline (Type: V2 | Execution mode: QUEUED)	✓ Succeeded	Source – 68c7ab55 second commit	2 minutes ago	✓ View details

Following output is to be generated of the code which is in the github repository.

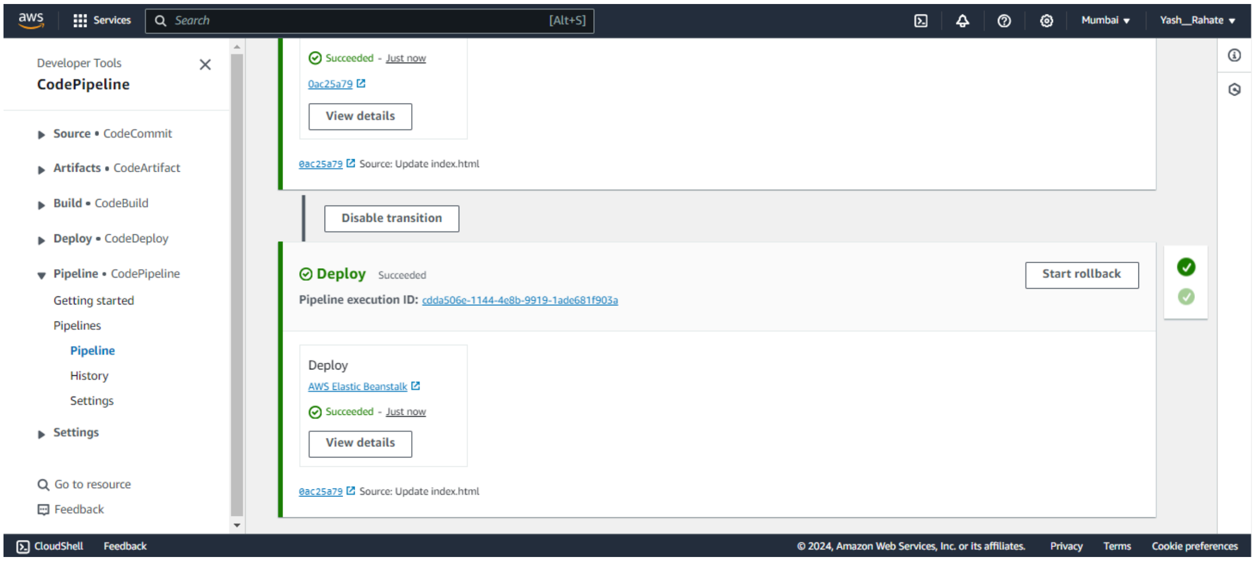


Unlimited movies, TV shows and more

Watch anywhere. Cancel anytime.

Ready to watch? Enter your email to create or restart your membership.

Changes are done in the code of the Github repository and it is being directly deployed without any configurations.



Succeeded - Just now
0ac25a79

0ac25a79 Source: Update index.html

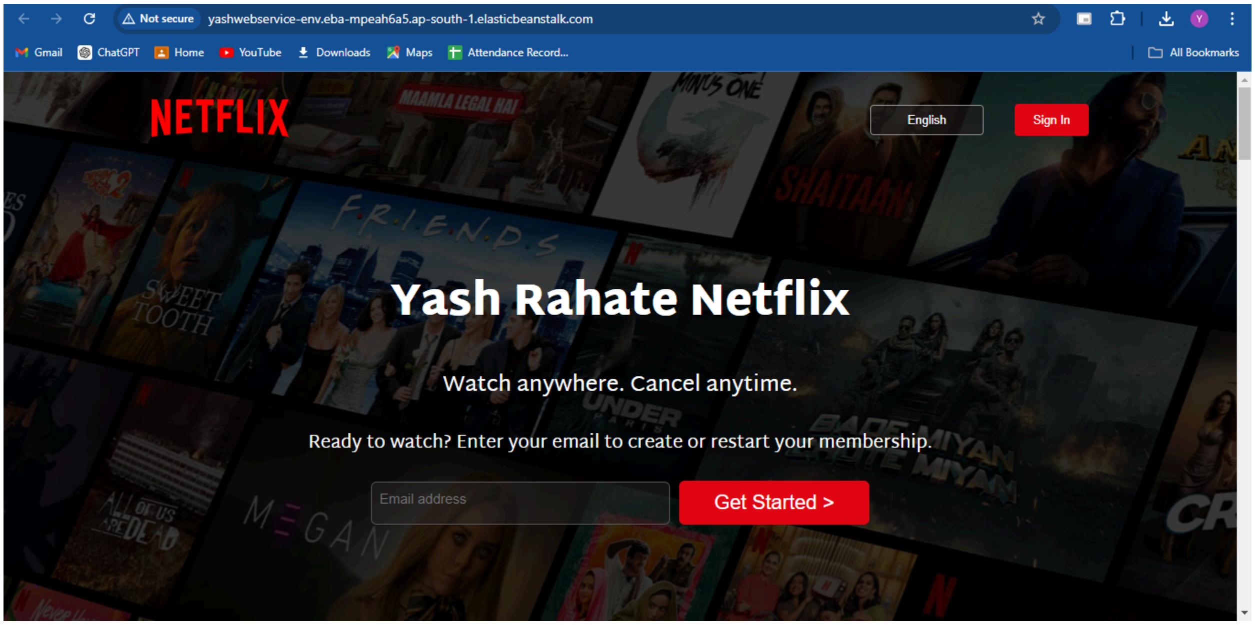
Deploy Succeeded

Pipeline execution ID: cdda506e-1144-4e8b-9919-1ade681f903a

Deploy
AWS Elastic Beanstalk
Succeeded - Just now

0ac25a79 Source: Update index.html

Changes are to be reflected.



Yash Rahate Netflix

Watch anywhere. Cancel anytime.

Ready to watch? Enter your email to create or restart your membership.

Conclusion:

Using AWS CodeBuild, CodePipeline, and CodeDeploy provides a comprehensive solution for automating the build, test, and deployment phases of application development. This not only improves the reliability of the software delivery process but also reduces human intervention, ensuring faster and more consistent application releases. With the ability to handle multiple stages such as source control integration, building, testing, and deployment, this CI/CD pipeline setup allows teams to focus on developing features and addressing issues while AWS manages the infrastructure automation at scale.