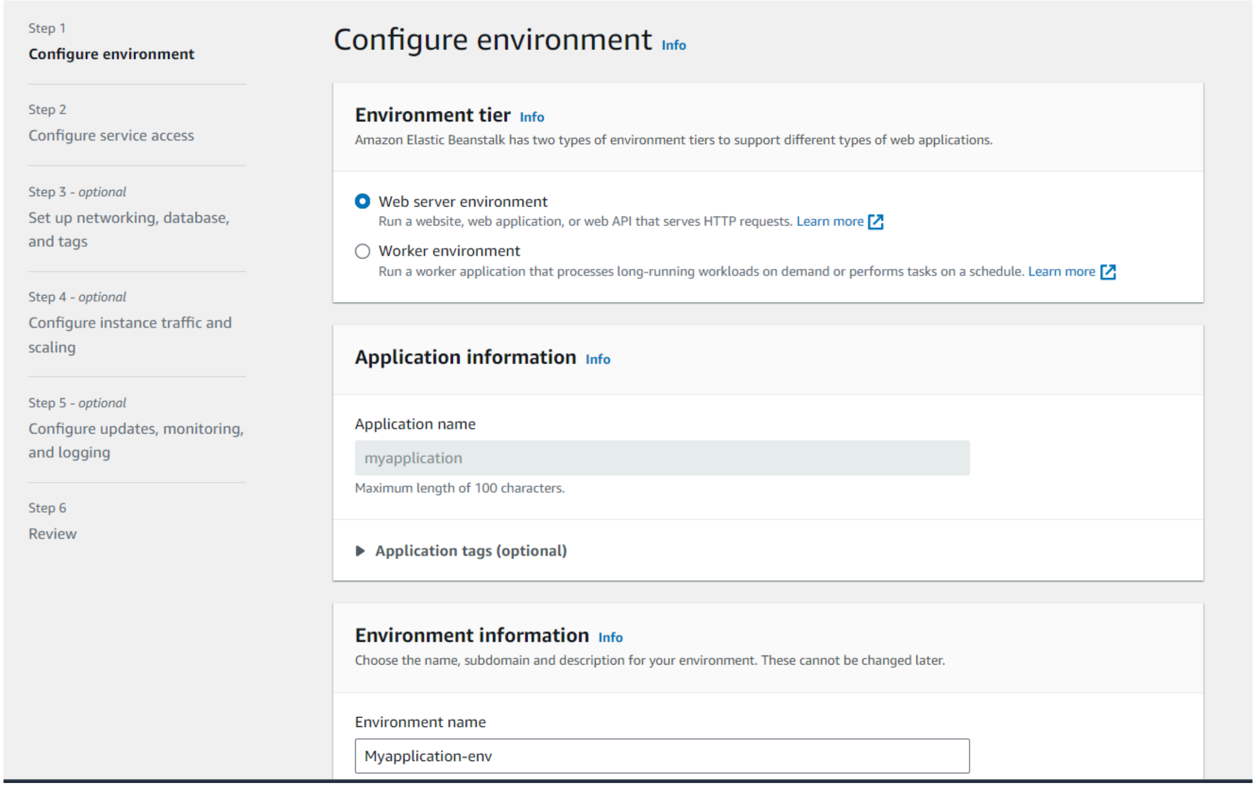
EXPERIMENT NO. 2

ATHARVA PRABHU
D15A 43

Step 1: create environment



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

Step 6
Review

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

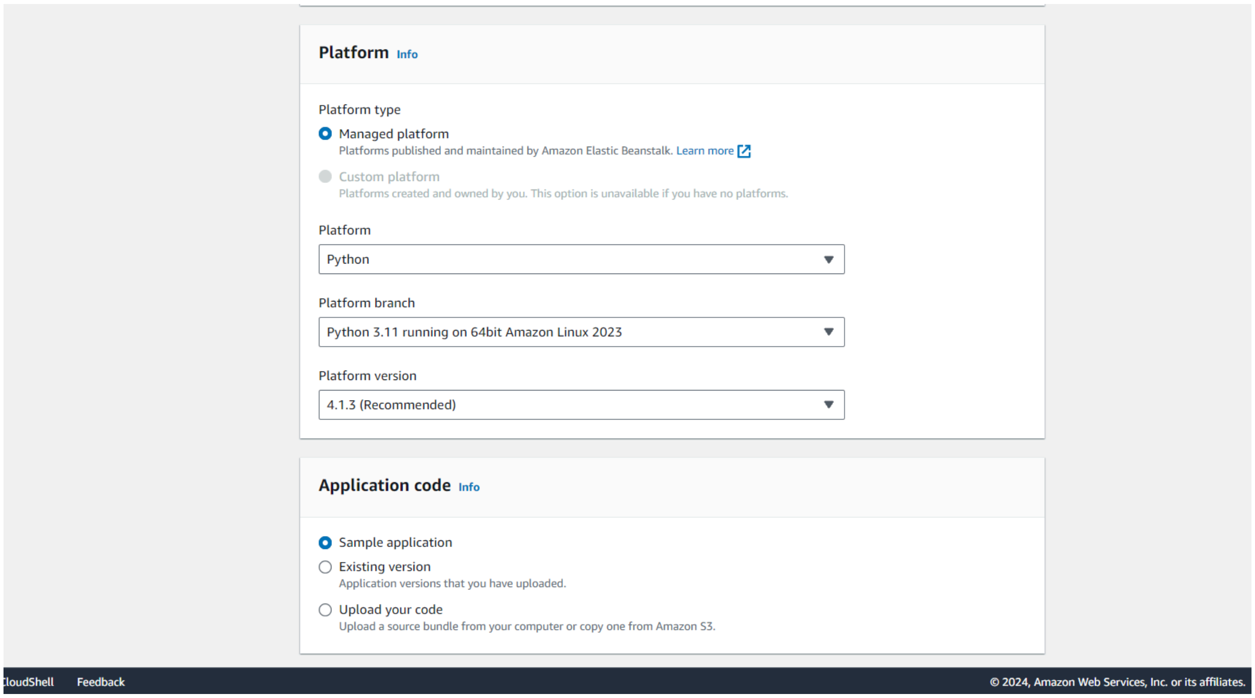
Application name
myapplication
Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

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

Environment name
Myapplication-env



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)

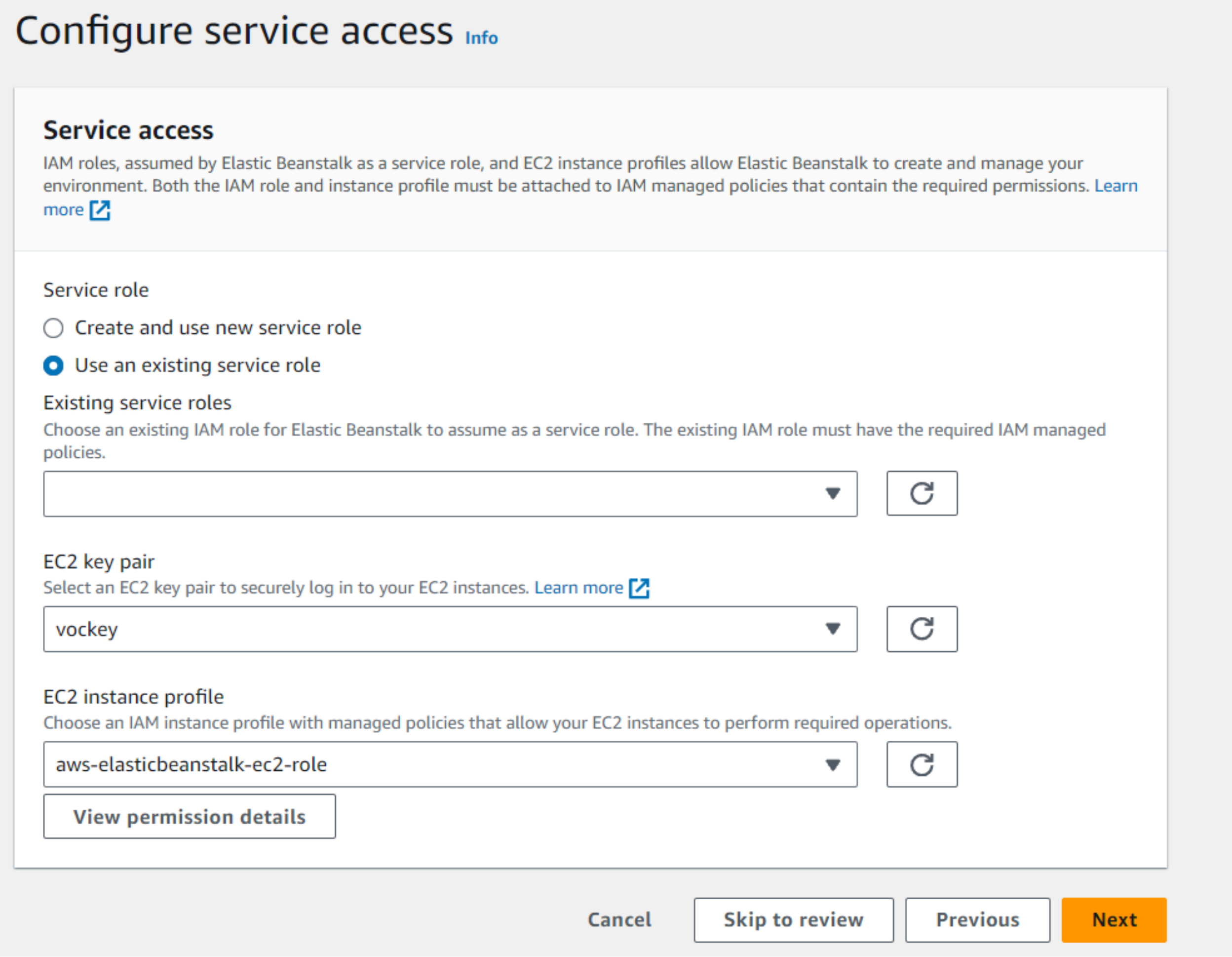
Application code Info

● 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.

Step 2 : add your Ec2 key pair and instance profile



Configure service access

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

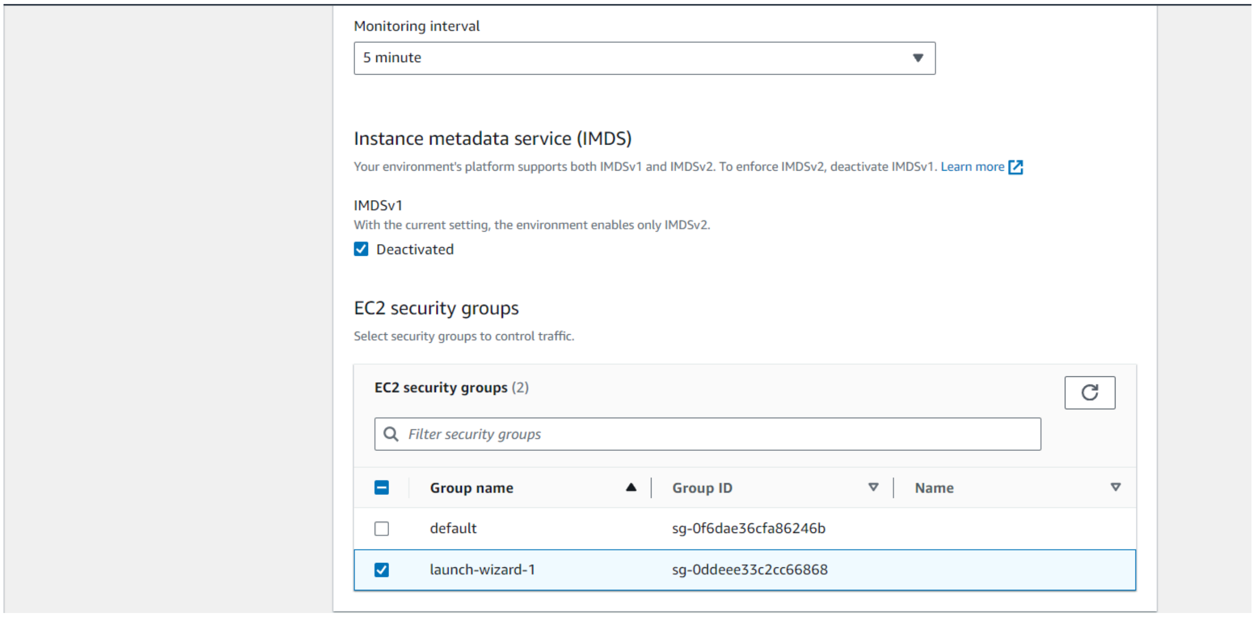
vockey

EC2 instance profile

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

aws-elasticbeanstalk-ec2-role

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

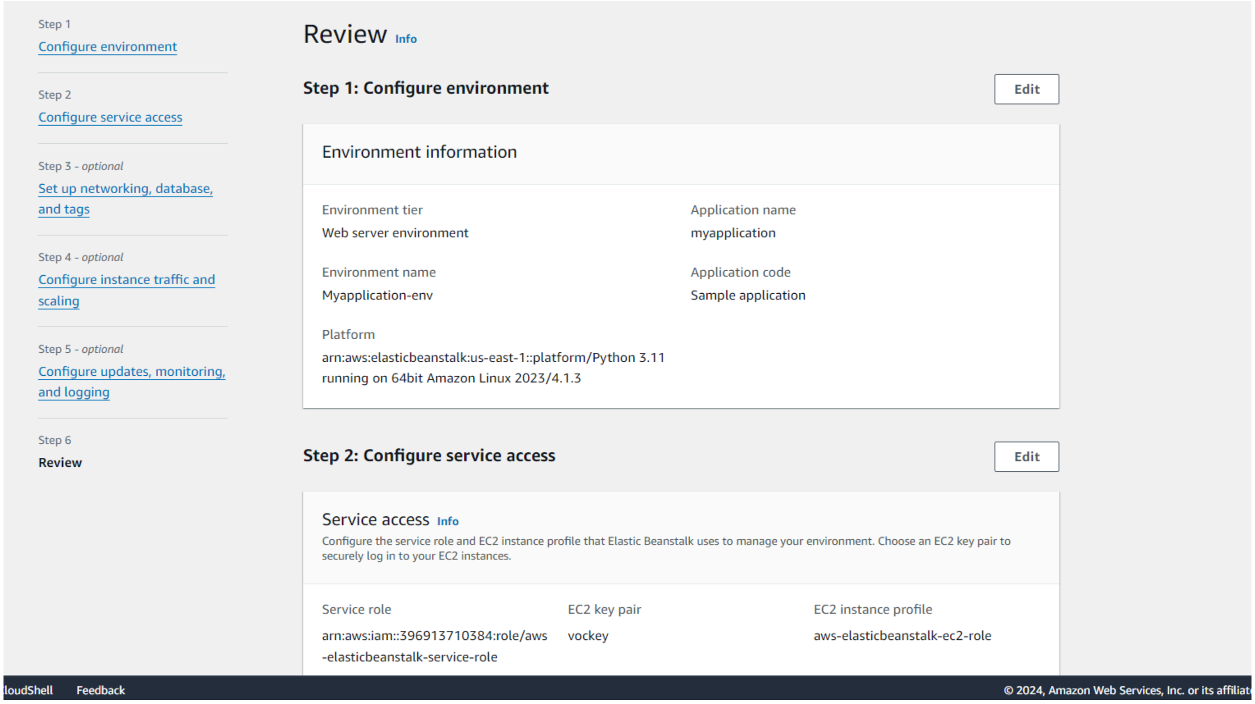
EC2 security groups

Select security groups to control traffic.

EC2 security groups (2)

Filter security groups

	Group name	Group ID	Name
□	default	sg-0f6dae36cfa86246b	
☑	launch-wizard-1	sg-0ddeee33c2cc66868	



Review Info

Step 1: Configure environment Edit

Environment information

Environment tier
Web server environment

Application name
myapplication

Environment name
Myapplication-env

Application code
Sample application

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

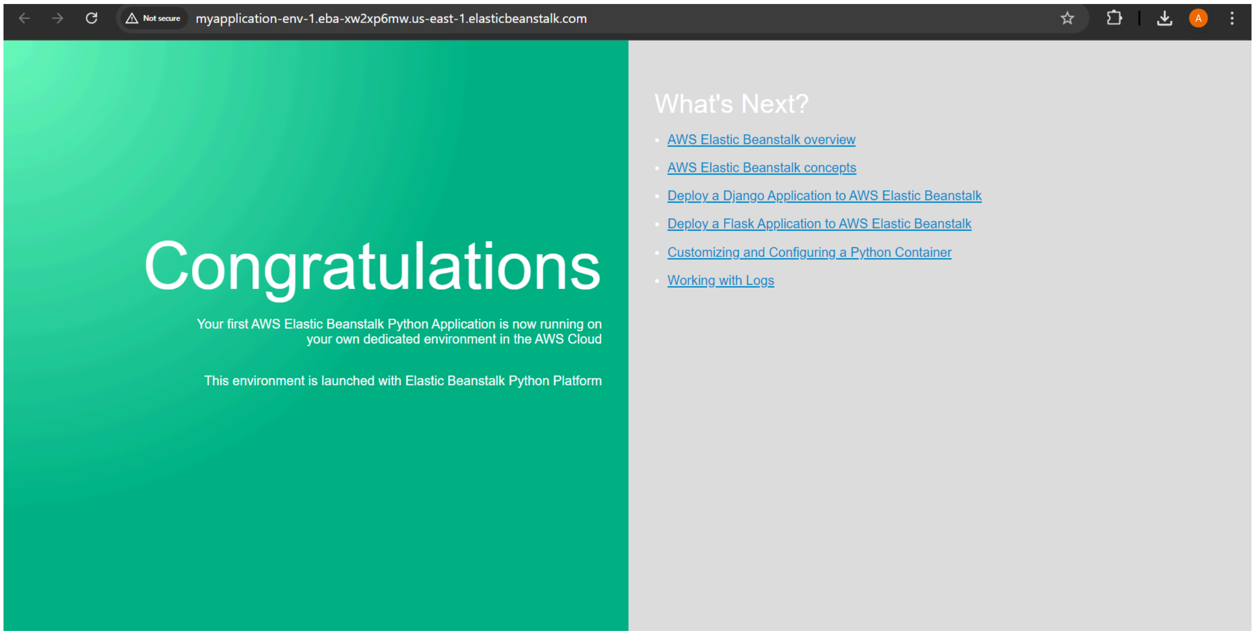
Step 2: Configure service access Edit

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	EC2 key pair	EC2 instance profile
arn:aws:iam::396913710384:role/aws-elasticbeanstalk-service-role	vockey	aws-elasticbeanstalk-ec2-role

Step 4 : Beanstalk environment is created



myapplication-env-1.eba-xw2xp6mw.us-east-1.elasticbeanstalk.com

Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

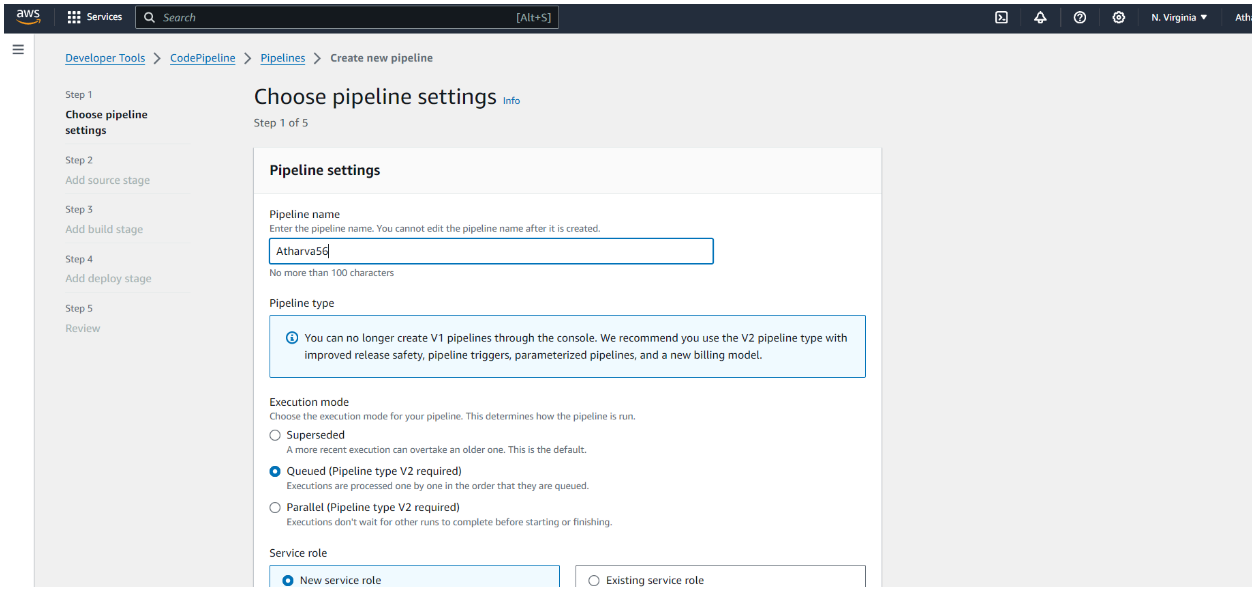
This environment is launched with Elastic Beanstalk Python Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy a Django Application to AWS Elastic Beanstalk
- Deploy a Flask Application to AWS Elastic Beanstalk
- Customizing and Configuring a Python Container
- Working with Logs

Pipeline Creation

Step 1 : click on create pipeline and give name



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

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.

Atharva56

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.

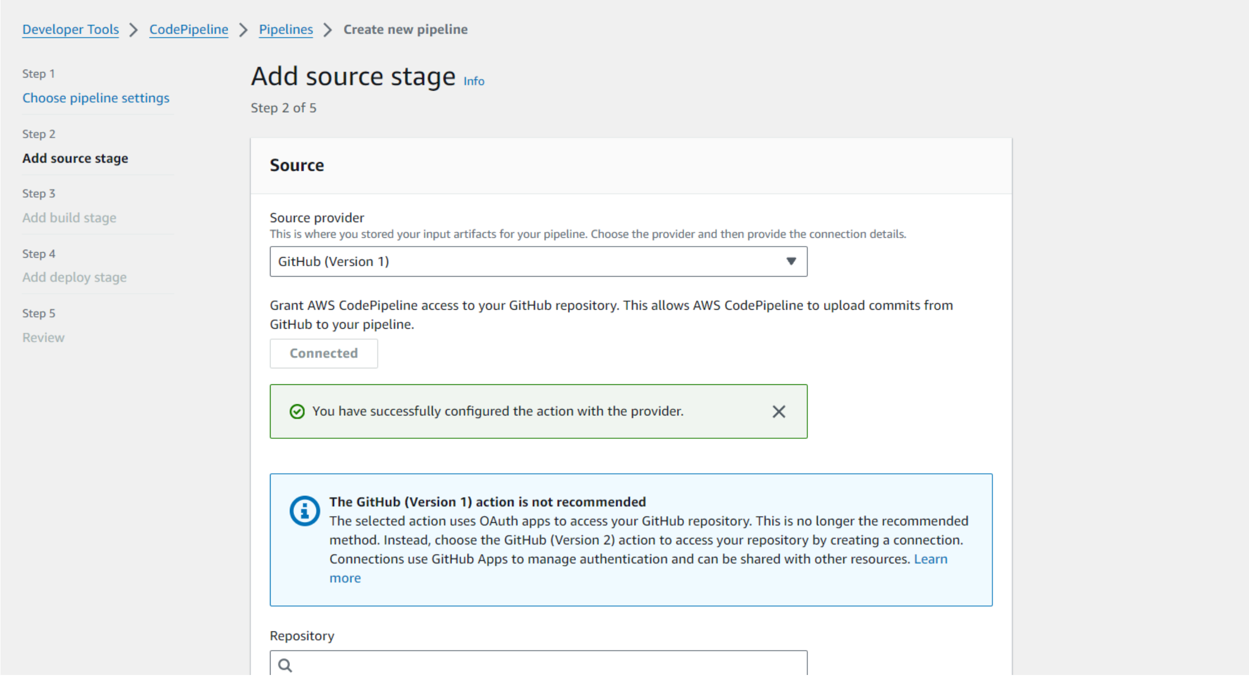
● Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

○ Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

Service role

● New service role ○ Existing service role

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



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)

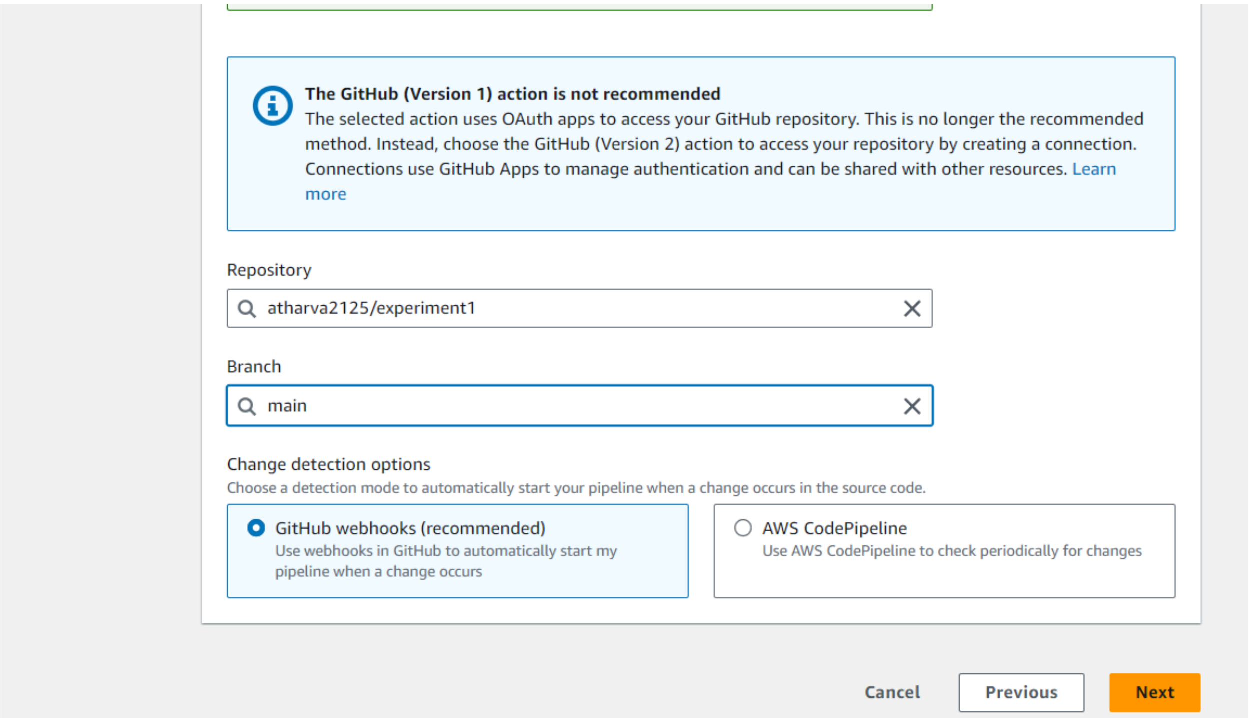
Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

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



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

atharva2125/experiment1

Branch

main

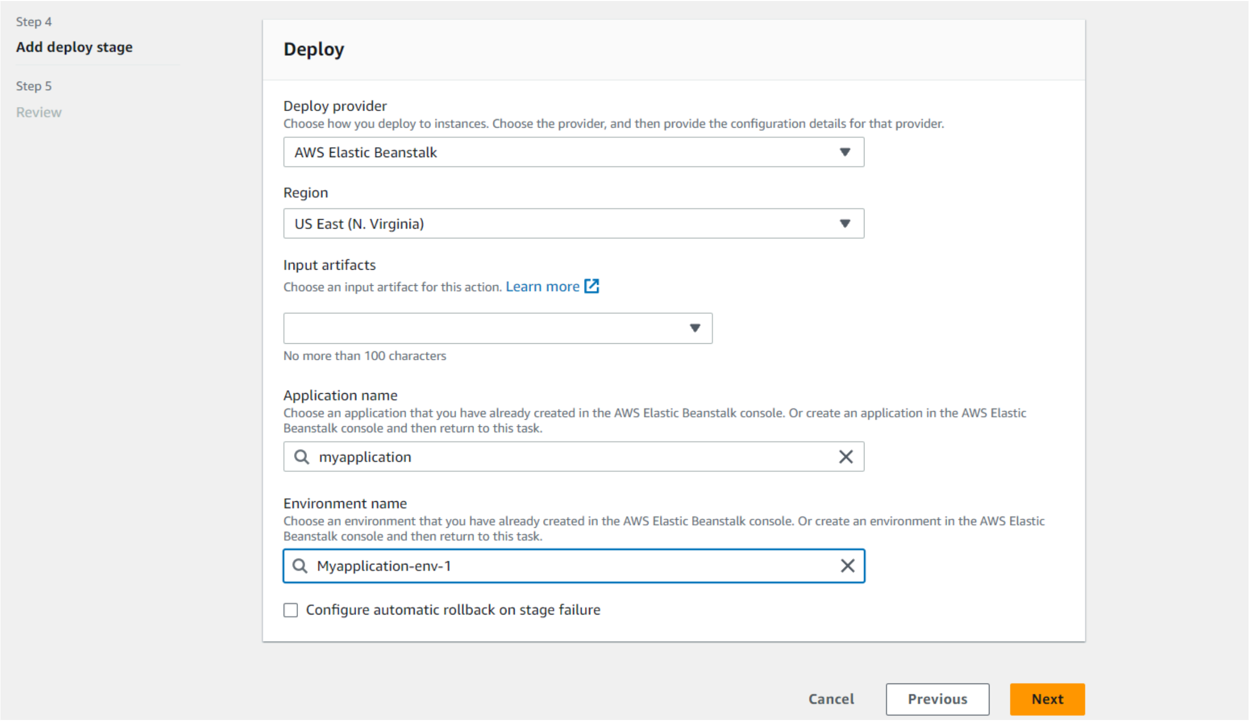
Change detection options

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



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.

myapplication

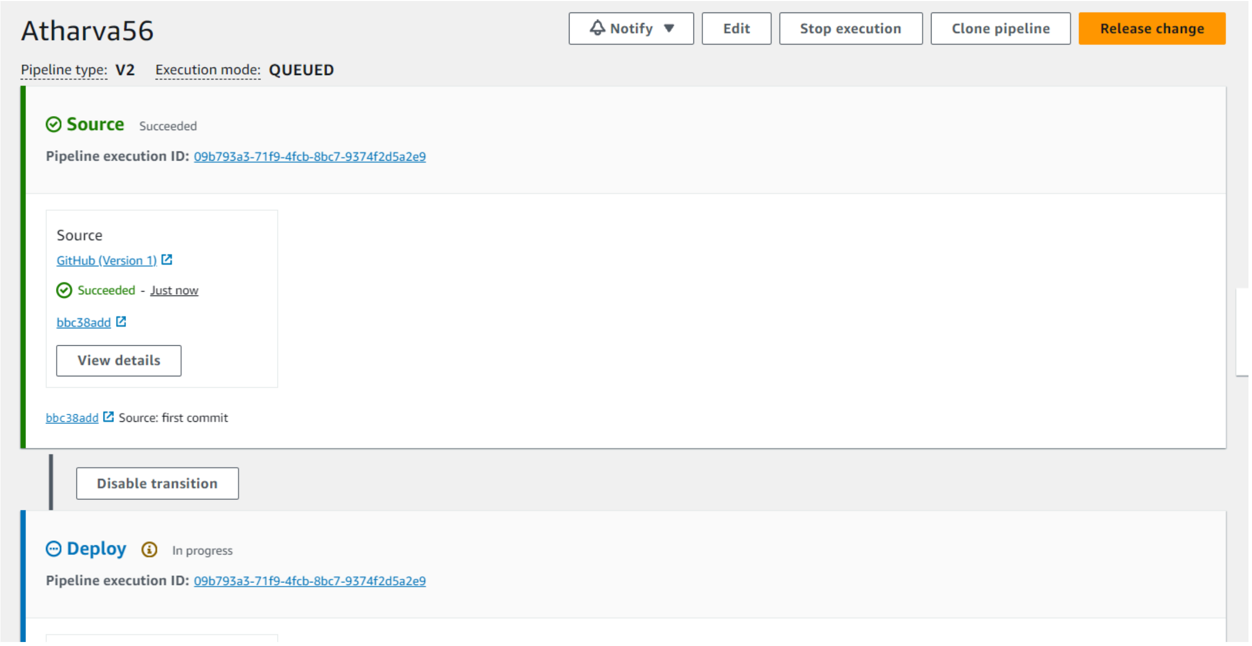
Environment name

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.

Myapplication-env-1

☐ Configure automatic rollback on stage failure

Step 5 : view the pipeline build and deployment



Atharva56

Pipeline type: **V2** Execution mode: **QUEUED**

Source Succeeded

Pipeline execution ID: 09b793a3-71f9-4fcb-8bc7-9374f2d5a2e9

Source

GitHub (Version 1)

Succeeded - Just now

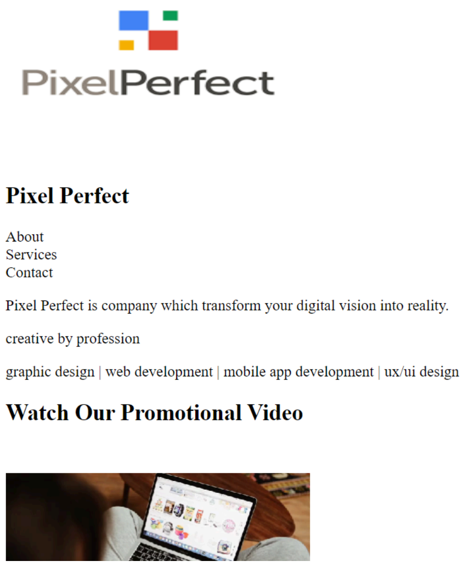
bbc38add

bbc38add Source: first commit

Deploy In progress

Pipeline execution ID: 09b793a3-71f9-4fcb-8bc7-9374f2d5a2e9

Step 6 : Check the deployed website at beanstalk link



PixelPerfect

Pixel Perfect

About
Services
Contact

Pixel Perfect is company which transform your digital vision into reality.

creative by profession

graphic design | web development | mobile app development | ux/ui design

Watch Our Promotional Video