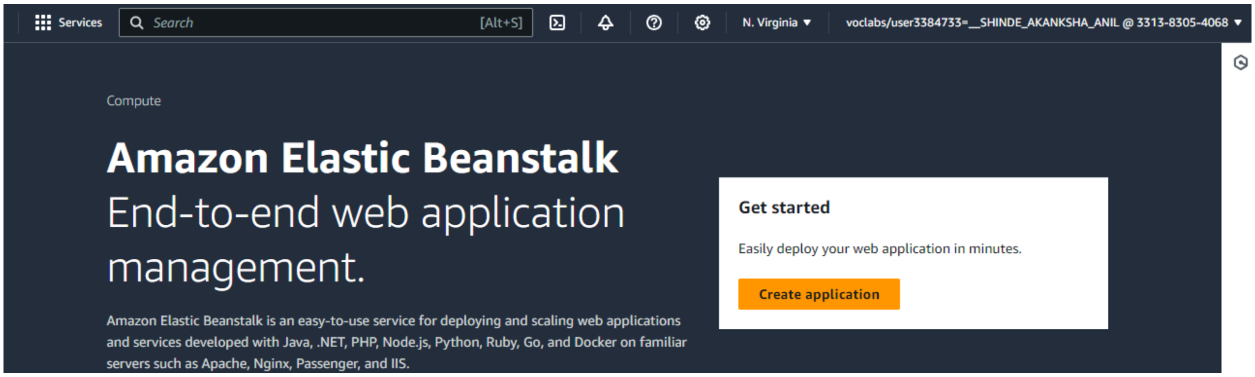
Name: Akanksha Shinde Class: D15C Roll No: 53

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.

1. Login to your AWS account and search for Elastic Beanstalk in the search box.


Services
Search
[Alt+S]
N. Virginia ▾
voclabs/user3384733=__SHINDE_AKANKSHA_ANIL @ 3313-8305-4068 ▾

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

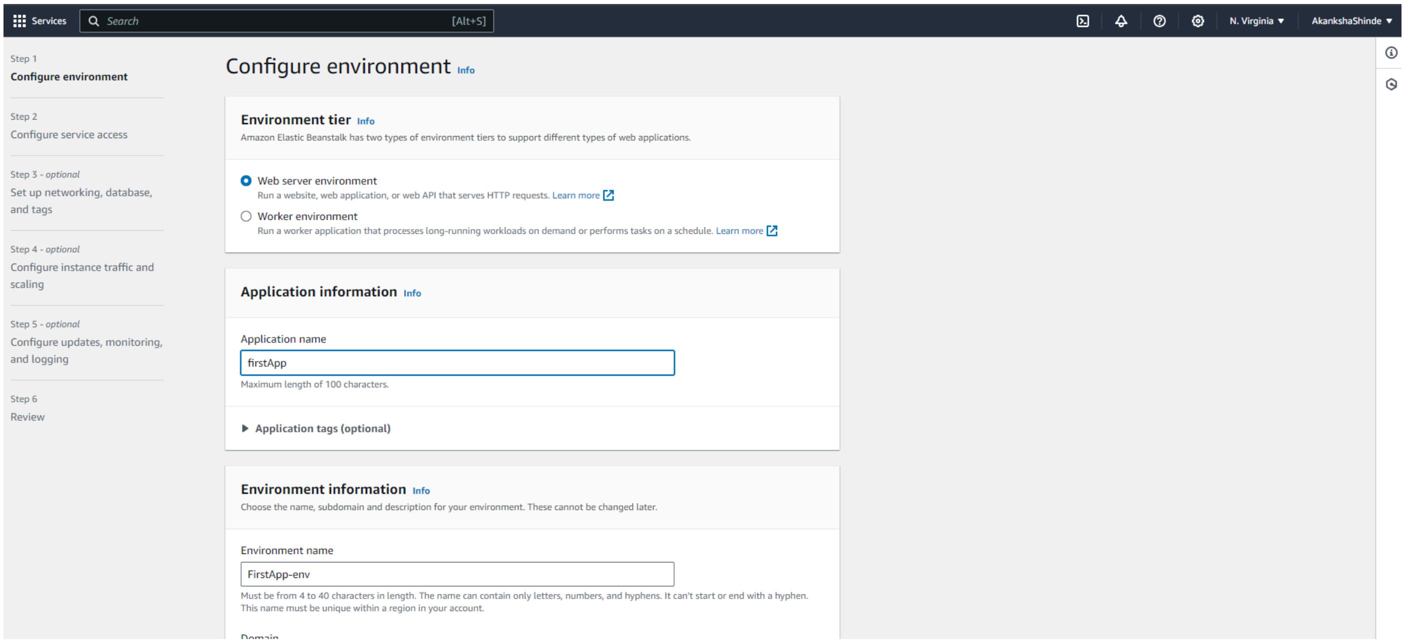
Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create application

2. Open up Elastic Beanstalk and name your web app.



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: firstApp

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.

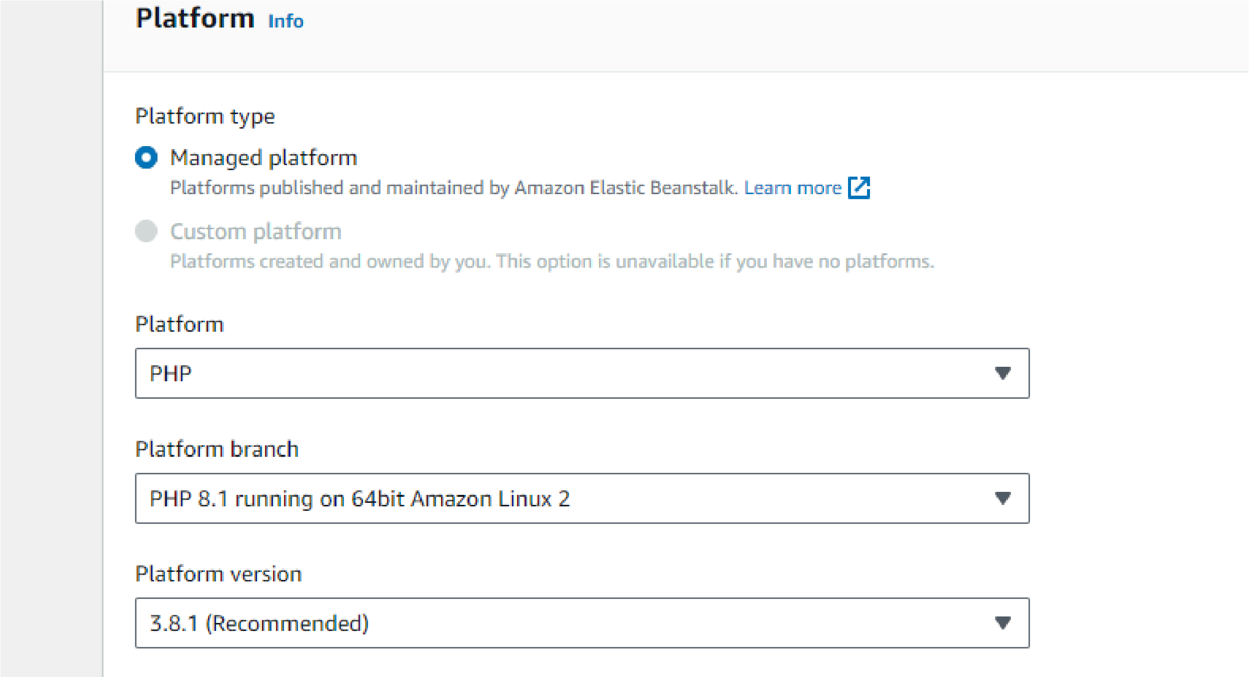
Environment name: FirstApp-env

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

Name: Akanksha Shinde Class: D15C Roll No: 53

3. Choose PHP from the drop-down menu and then click Create Application.



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.

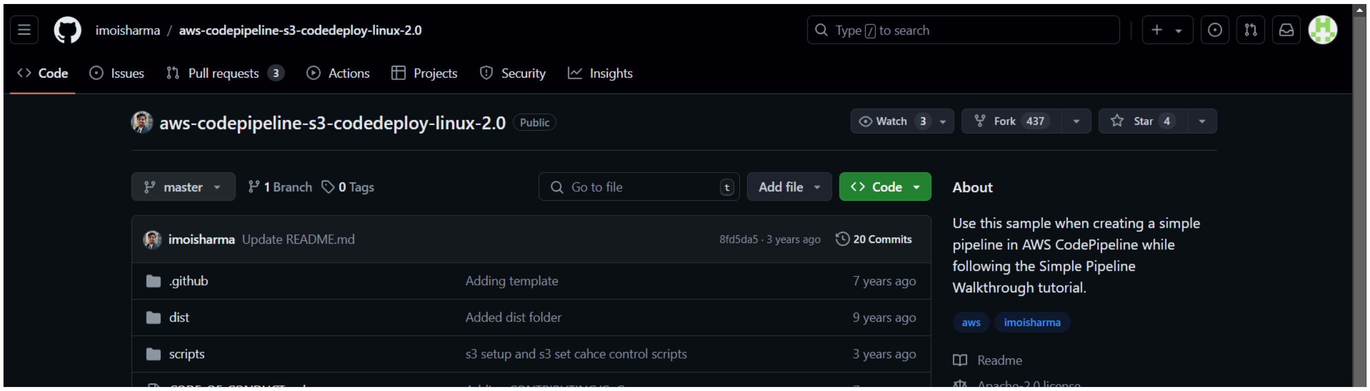
Platform: PHP

Platform branch: PHP 8.1 running on 64bit Amazon Linux 2

Platform version: 3.8.1 (Recommended)

4. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

Step 2: Get a copy of your sample code



aws-codepipeline-s3-codedeploy-linux-2.0 Public

imoisharma Update README.md 8fd5da5 · 3 years ago 20 Commits

.github Adding template 7 years ago

dist Added dist folder 9 years ago

scripts s3 setup and s3 set cahce control scripts 3 years ago

About

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.

aws imoisharma

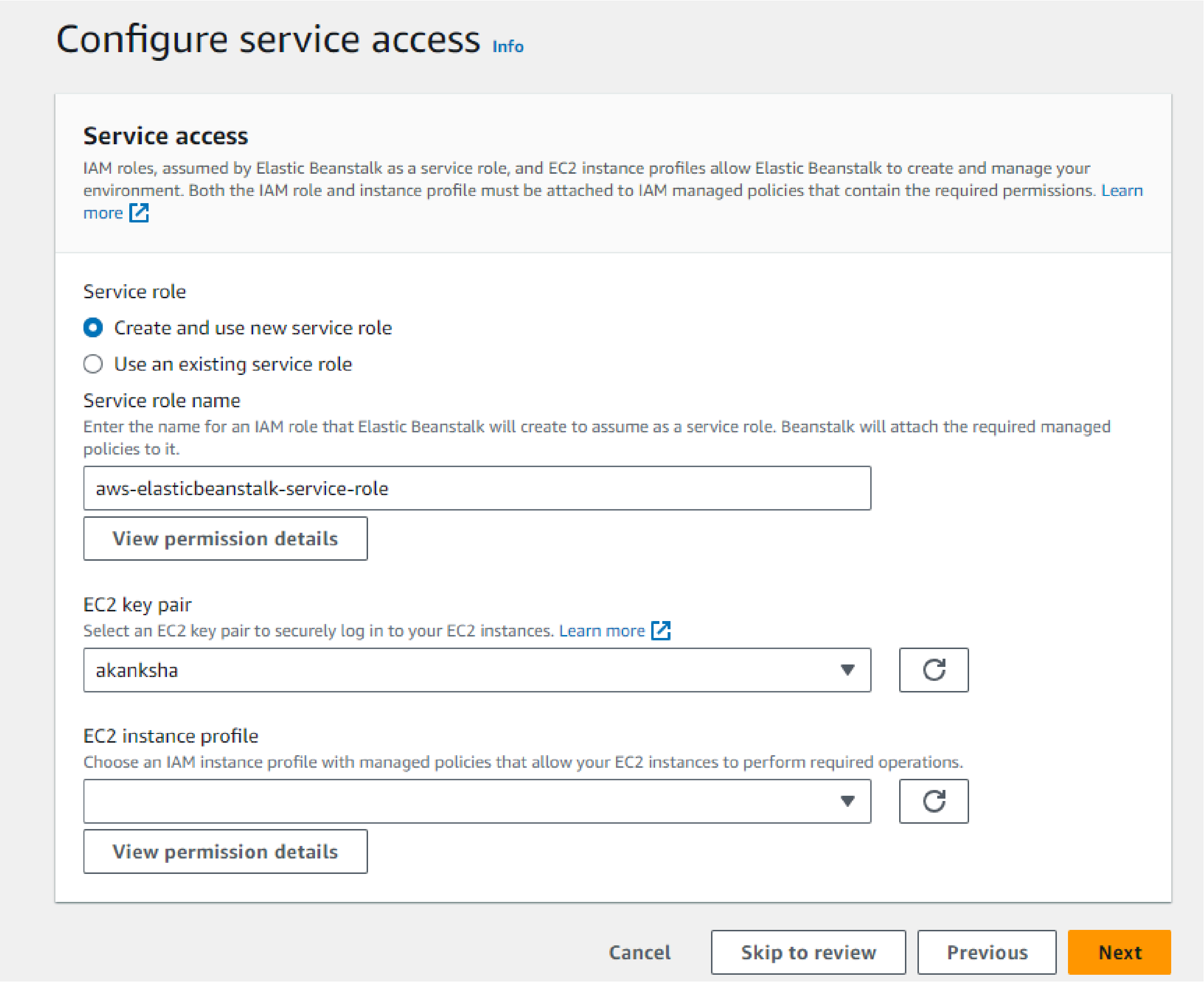
Readme

Apache-2.0 license

In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.
For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

Name: Akanksha Shinde Class: D15C Roll No: 53

Go to the repository shared above and simply fork it.



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

Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

View permission details

EC2 key pair

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

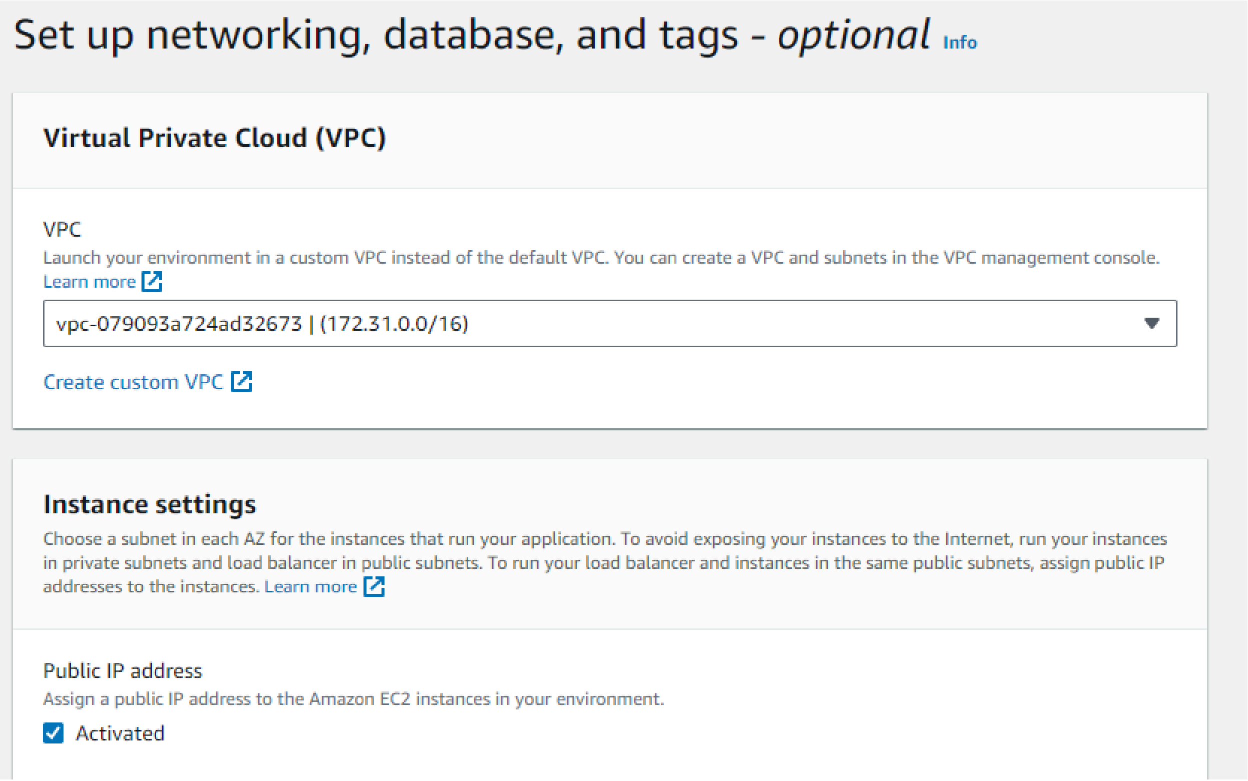
akanksha

EC2 instance profile

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

View permission details

Name: Akanksha Shinde Class: D15C Roll No: 53



Set up networking, database, and tags - *optional* Info

Virtual Private Cloud (VPC)

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-079093a724ad32673 | (172.31.0.0/16)

Create custom VPC

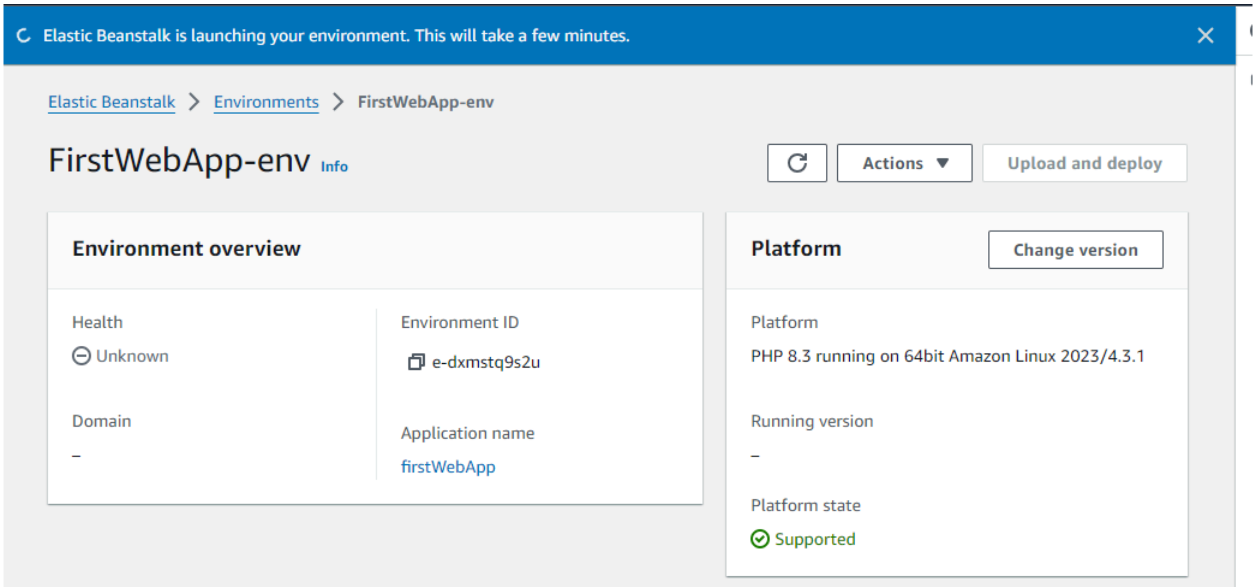
Instance settings

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



Elastic Beanstalk is launching your environment. This will take a few minutes.

Elastic Beanstalk > Environments > FirstWebApp-env

FirstWebApp-env Info

Environment overview

Health: Unknown

Environment ID: e-dxmstq9s2u

Domain: –

Application name: firstWebApp

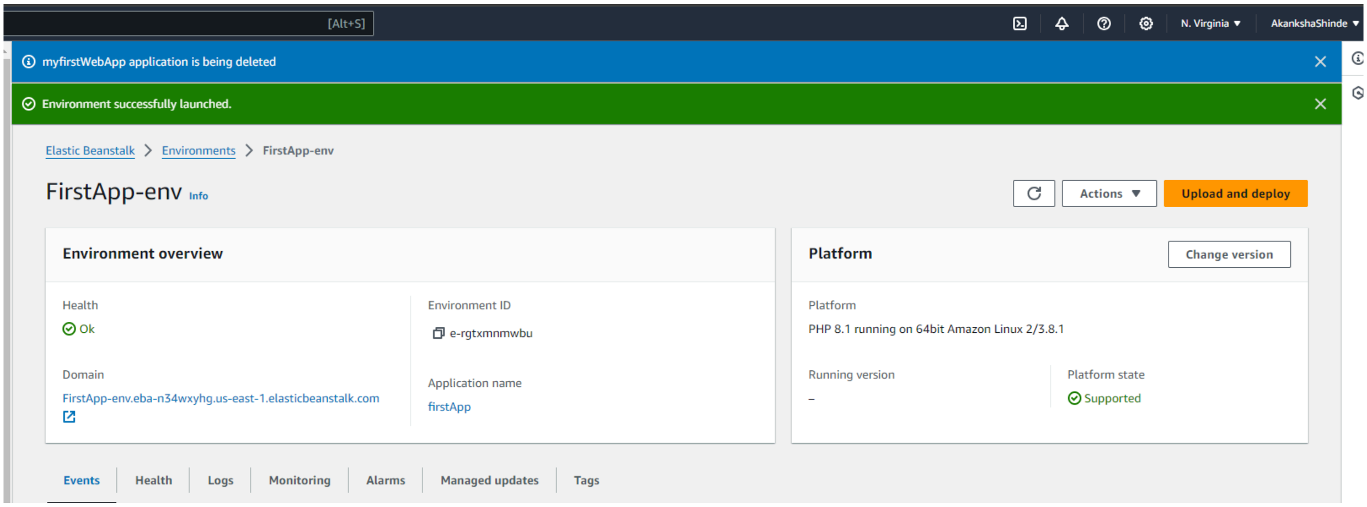
Platform

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

Running version: –

Platform state: Supported

Name: Akanksha Shinde Class: D15C Roll No: 53



myfirstWebApp application is being deleted

Environment successfully launched.

Elastic Beanstalk > Environments > FirstApp-env

FirstApp-env Info

Actions ▾ | Upload and deploy

Environment overview

Health: Ok

Environment ID: e-rgtxmnmwbu

Domain: FirstApp-env.eba-n34wxyhg.us-east-1.elasticbeanstalk.com

Application name: firstApp

Platform

Change version

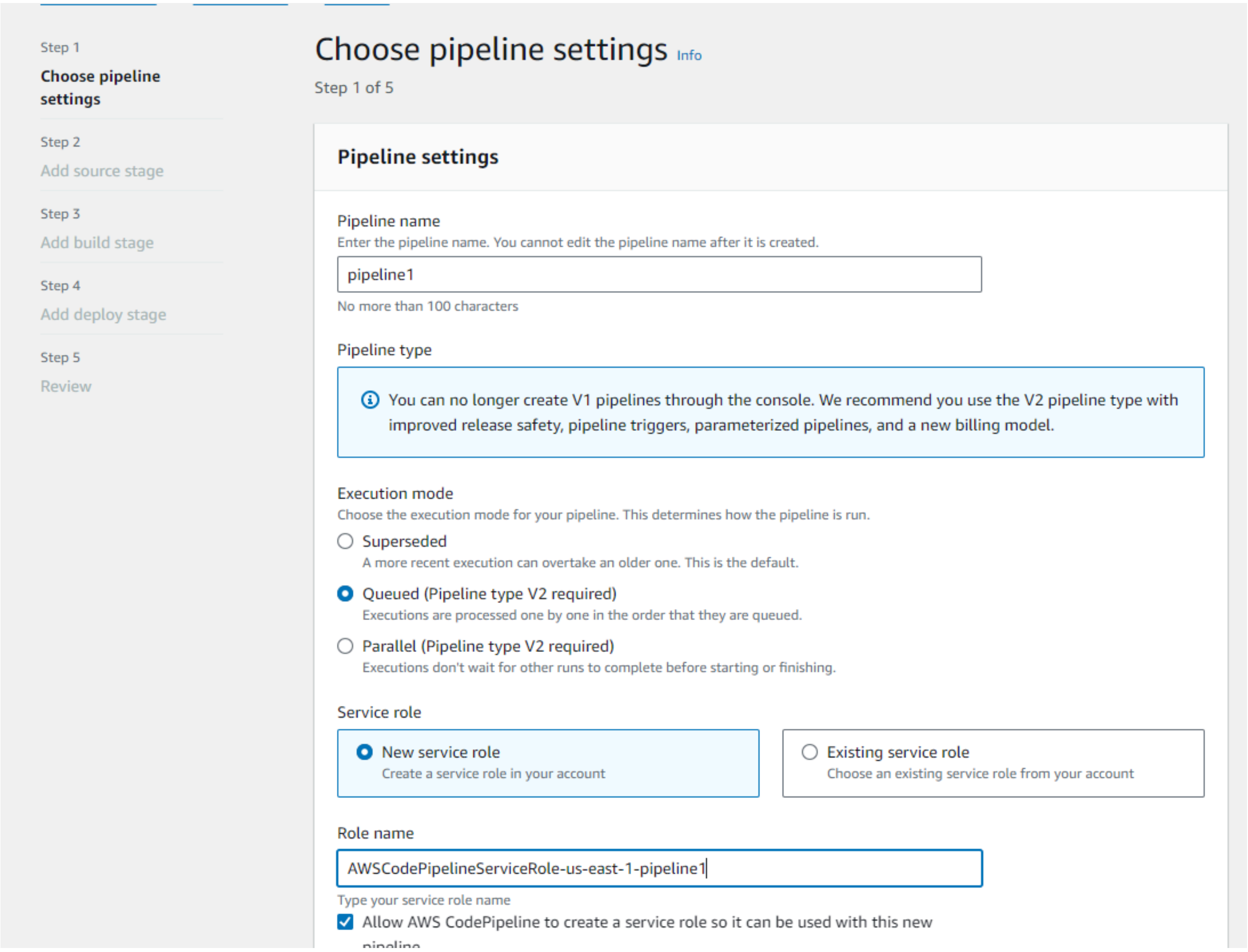
Platform: PHP 8.1 running on 64bit Amazon Linux 2/3.8.1

Running version: –

Platform state: Supported

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

Now to create pipeline,
1. Go to AWS Developer Tools -> CodePipeline and create a new Pipeline. Fill in the initial settings first.


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.

pipeline1

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
Create a service role in your account

○ Existing service role
Choose an existing service role from your account

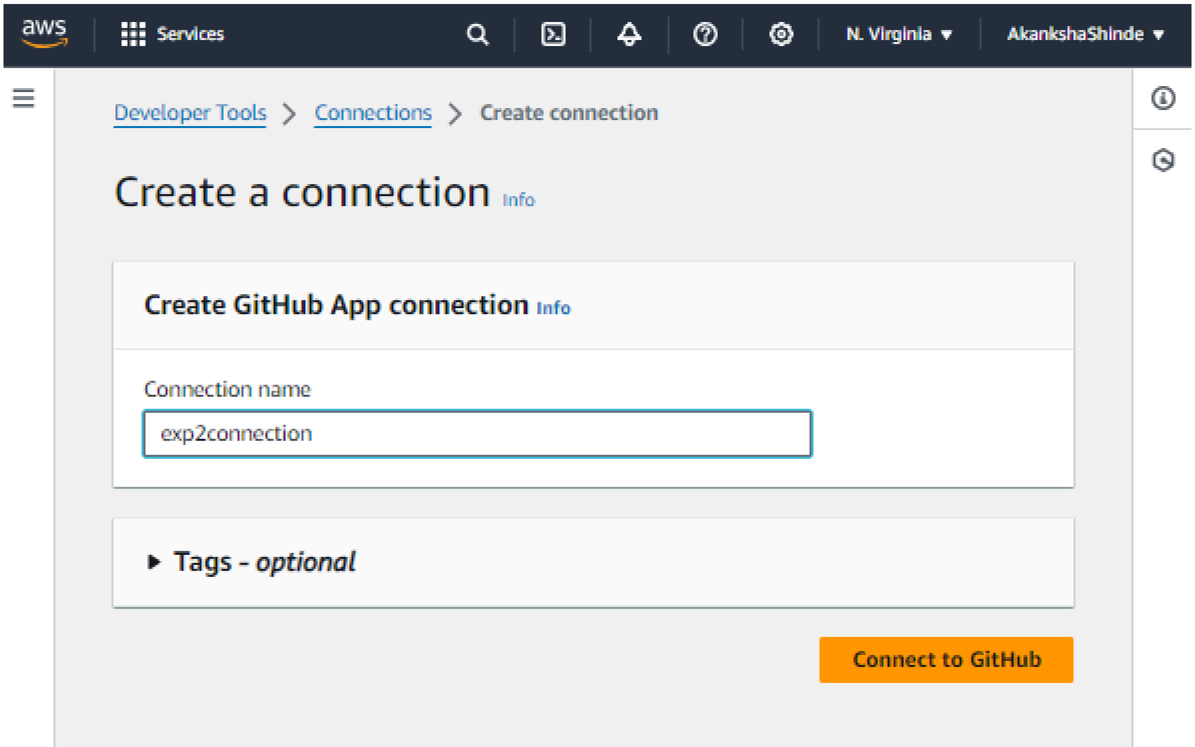
Role name

AWSCodePipelineServiceRole-us-east-1-pipeline1

Type your service role name

☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Name: Akanksha Shinde Class: D15C Roll No: 53



Create a connection Info

Create GitHub App connection Info

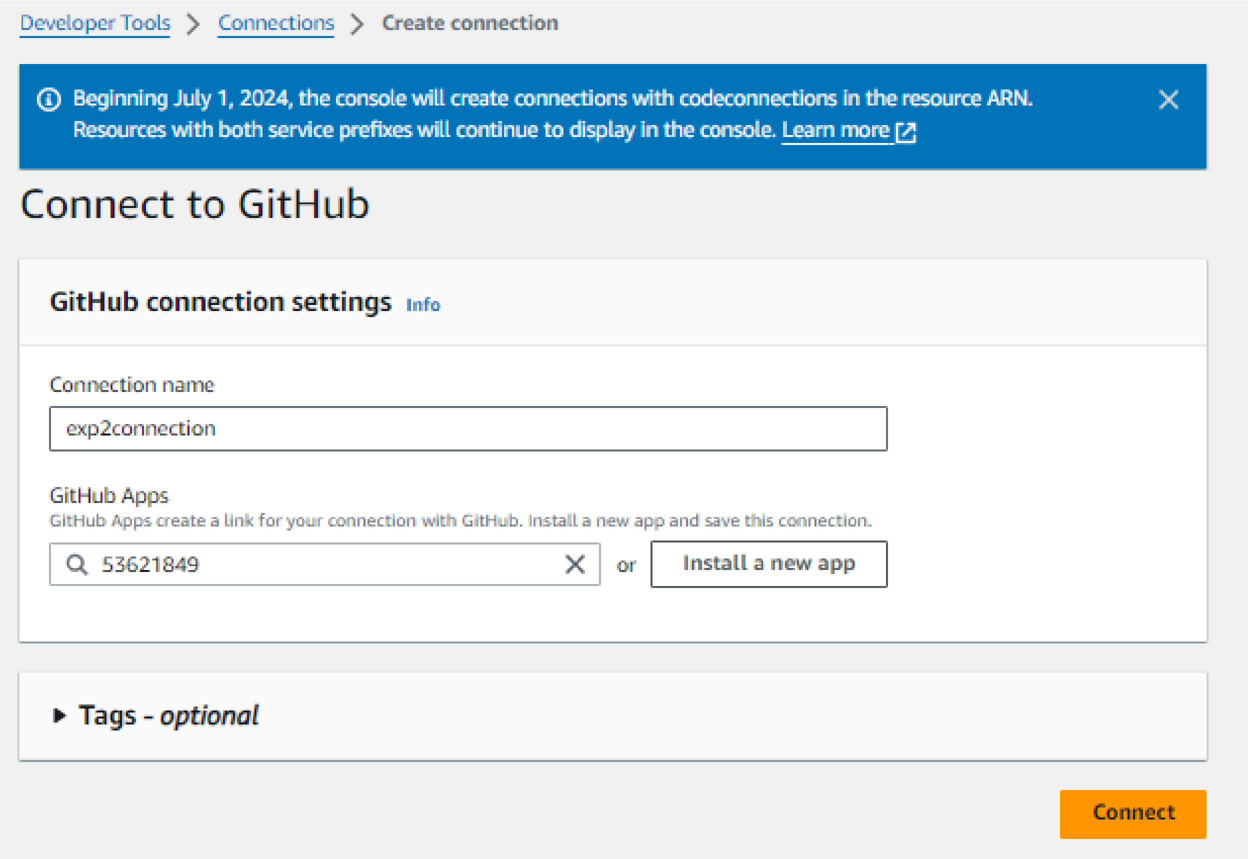
Connection name

exp2connection

▸ Tags - *optional*

2. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

Name: Akanksha Shinde Class: D15C Roll No: 53



Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. Learn more

Connect to GitHub

GitHub connection settings Info

Connection name

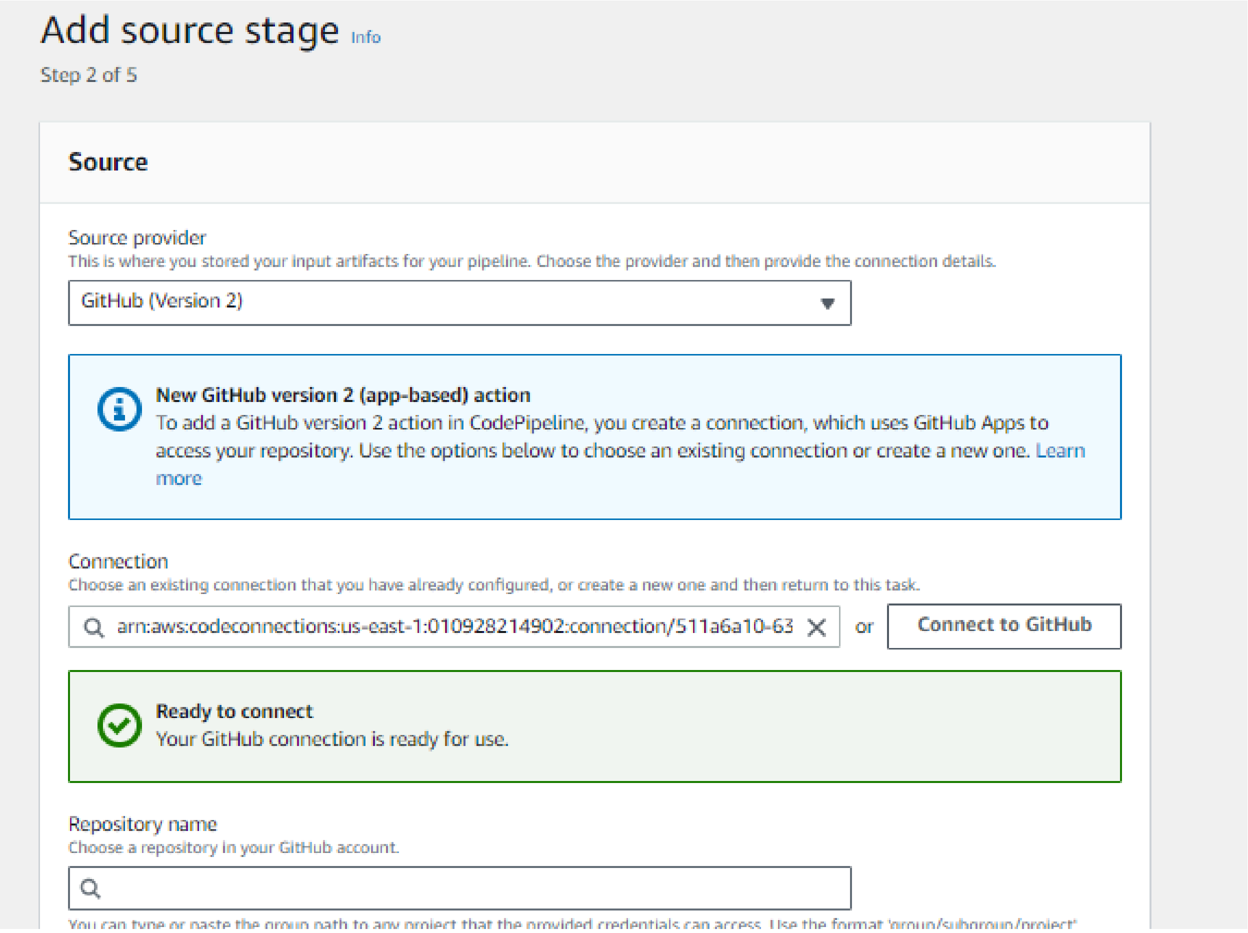
exp2connection

GitHub Apps

GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.

53621849

Name: Akanksha Shinde Class: D15C Roll No: 53



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 2) ▼

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

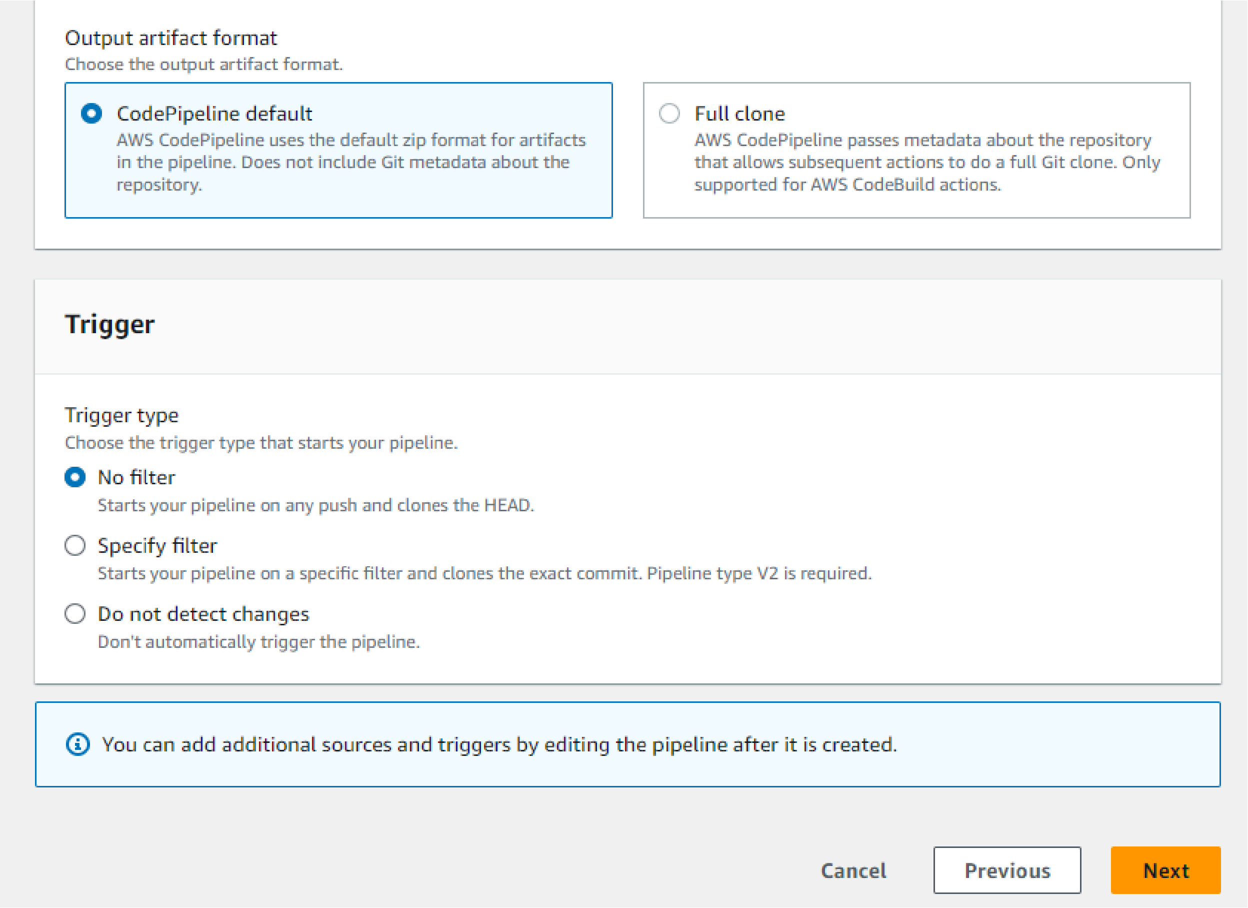
arn:aws:codeconnections:us-east-1:010928214902:connection/511a6a10-63 ✕ or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'

Name: Akanksha Shinde Class: D15C Roll No: 53



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.

Trigger

Trigger type

Choose the trigger type that starts your pipeline.

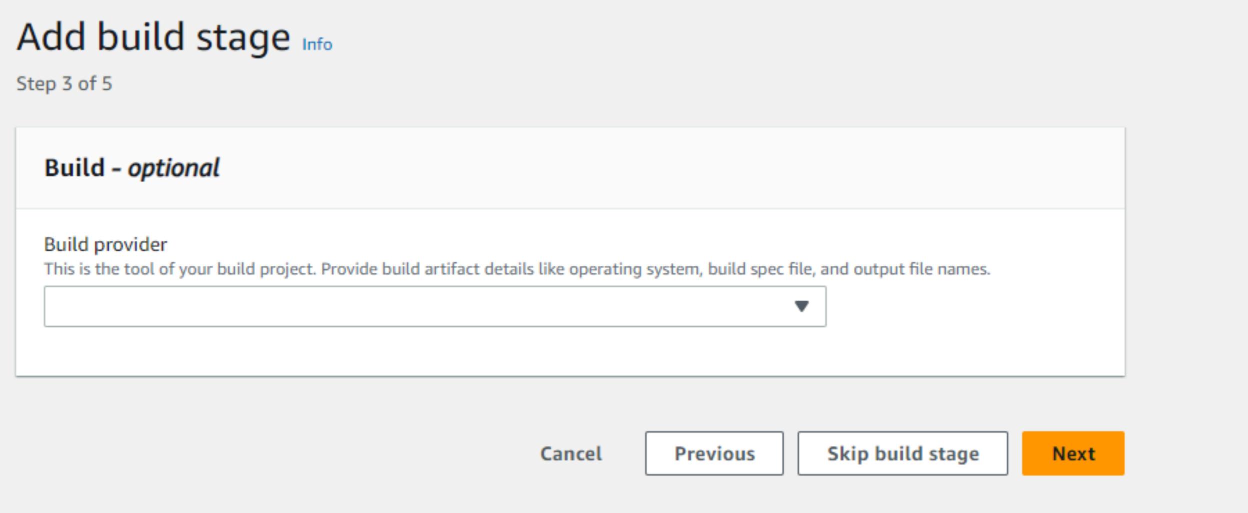
● No filter
Starts your pipeline on any push and clones the HEAD.

○ Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

○ Do not detect changes
Don't automatically trigger the pipeline.

You can add additional sources and triggers by editing the pipeline after it is created.

3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.



Add build stage Info

Step 3 of 5

Build - *optional*

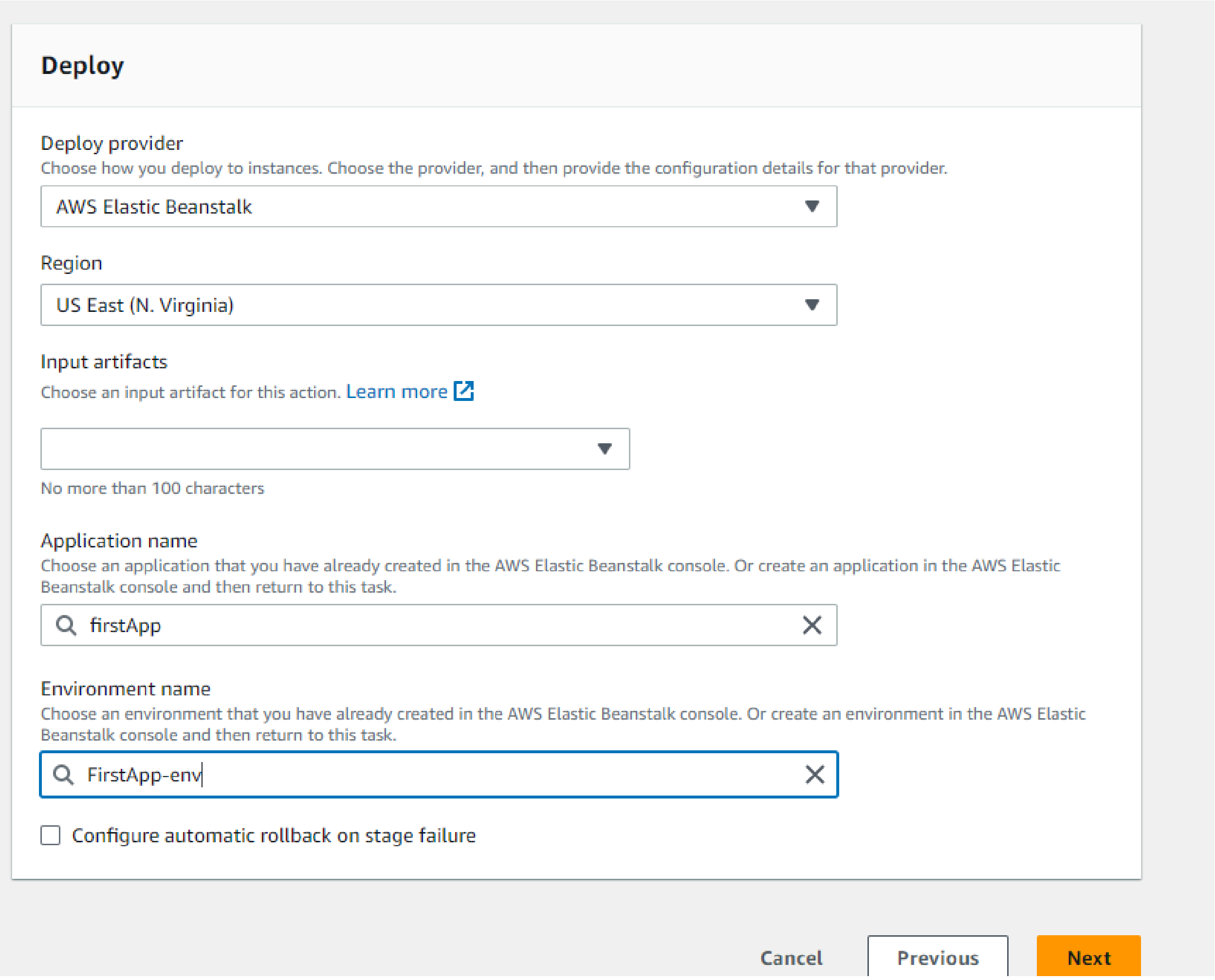
Build provider

This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Name: Akanksha Shinde Class: D15C Roll No: 53

Step 4: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.



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.

firstApp

Environment name

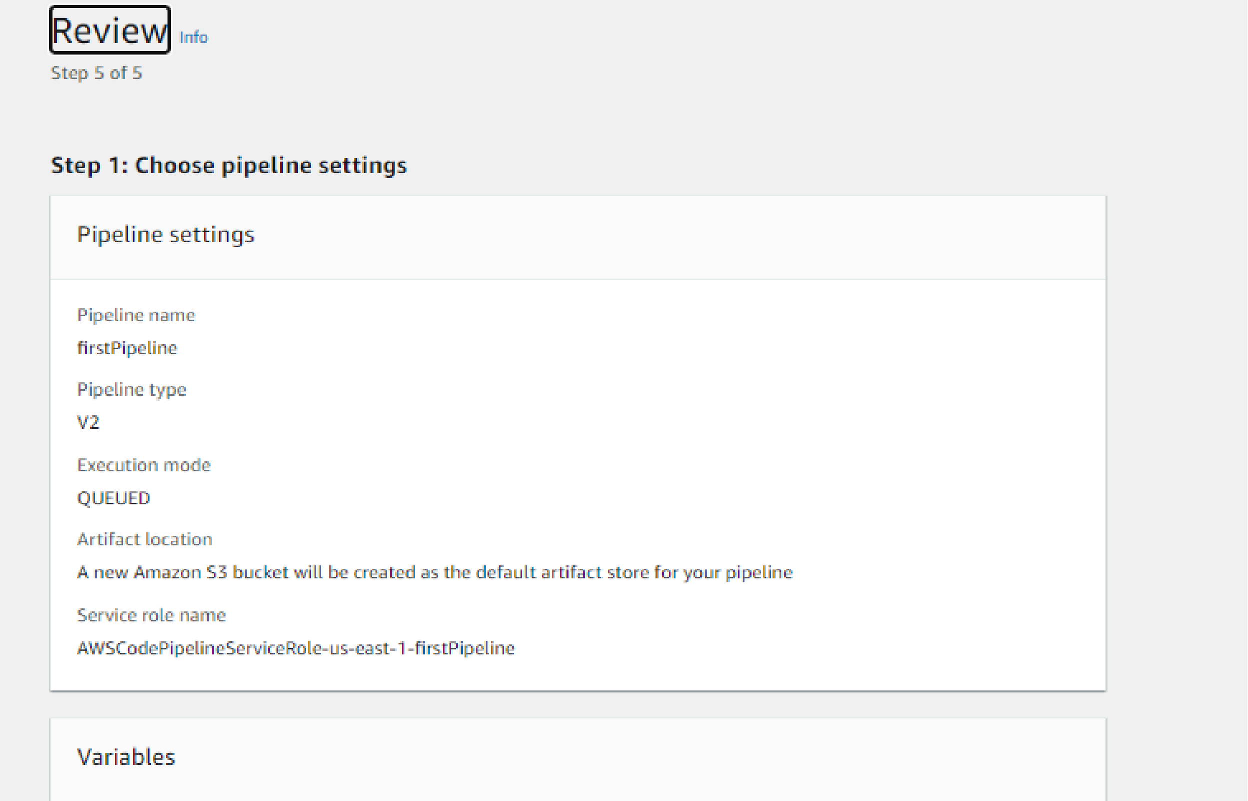
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

FirstApp-env

□ Configure automatic rollback on stage failure

Name: Akanksha Shinde Class: D15C Roll No: 53

2. Review all the settings and click on create pipeline



Step 5 of 5

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
firstPipeline

Pipeline type
V2

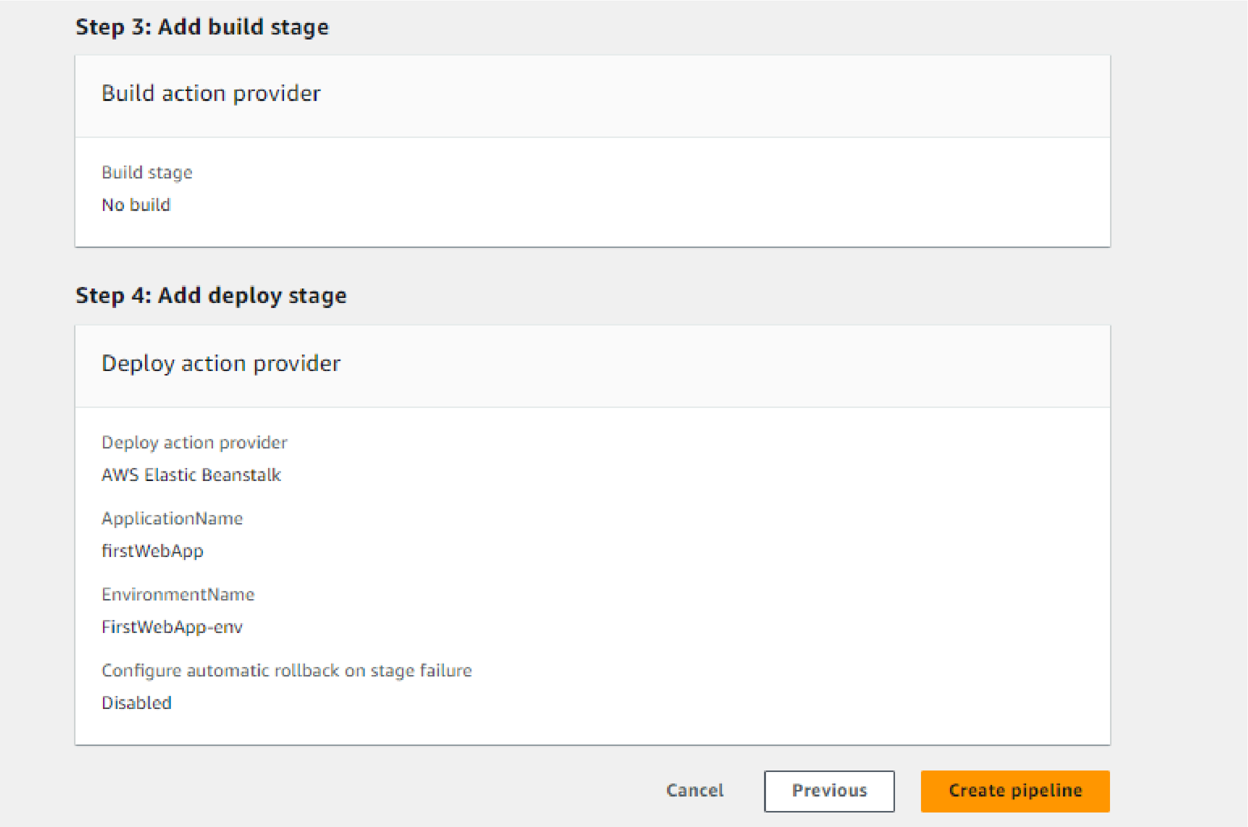
Execution mode
QUEUED

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-us-east-1-firstPipeline

Variables

Name: Akanksha Shinde Class: D15C Roll No: 53


Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

Deploy action provider

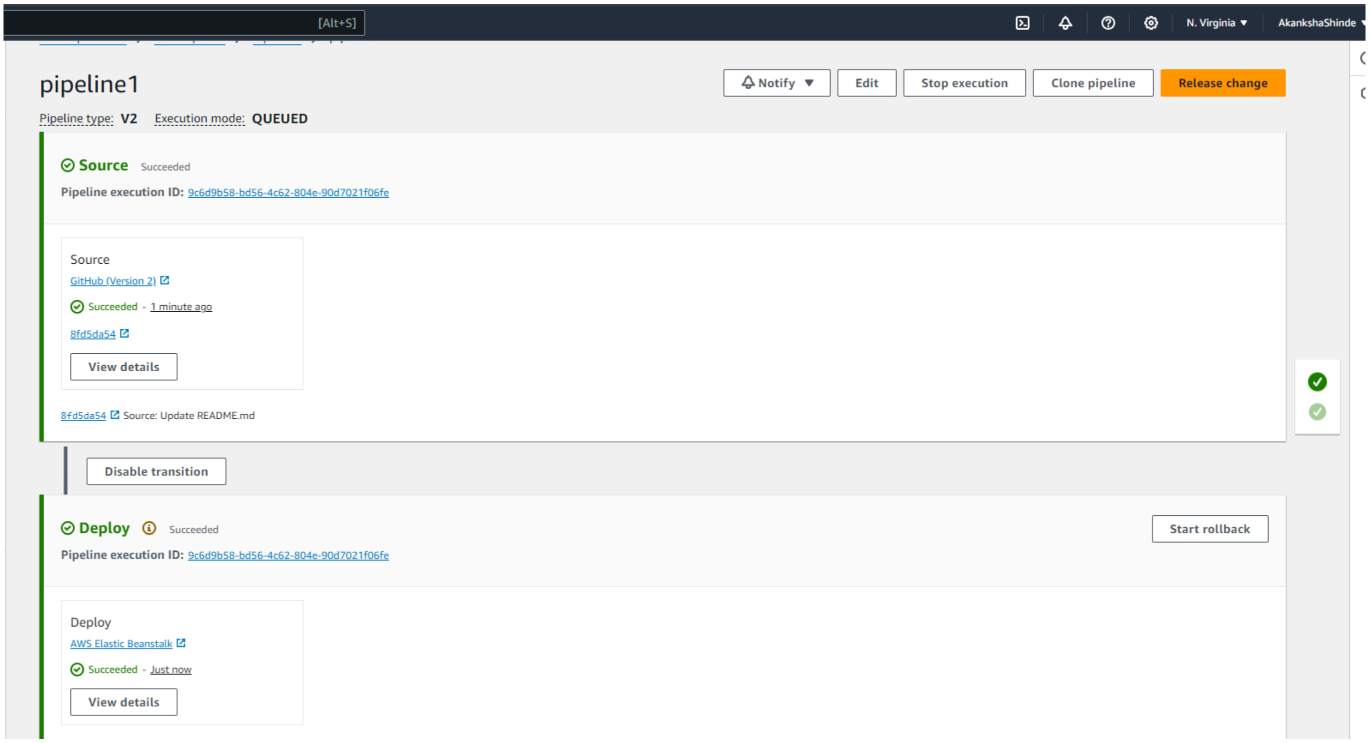
Deploy action provider
AWS Elastic Beanstalk

ApplicationName
firstWebApp

EnvironmentName
FirstWebApp-env

Configure automatic rollback on stage failure
Disabled

Finally you will be able to see this screen where you can infer the code has been deployed successfully.



pipeline1

Pipeline type: V2   Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: 9c6d9b58-bd56-4c62-804e-90d7021f06fe

Source
GitHub (Version 2)
Succeeded - 1 minute ago
8fd5da54

8fd5da54 Source: Update README.md

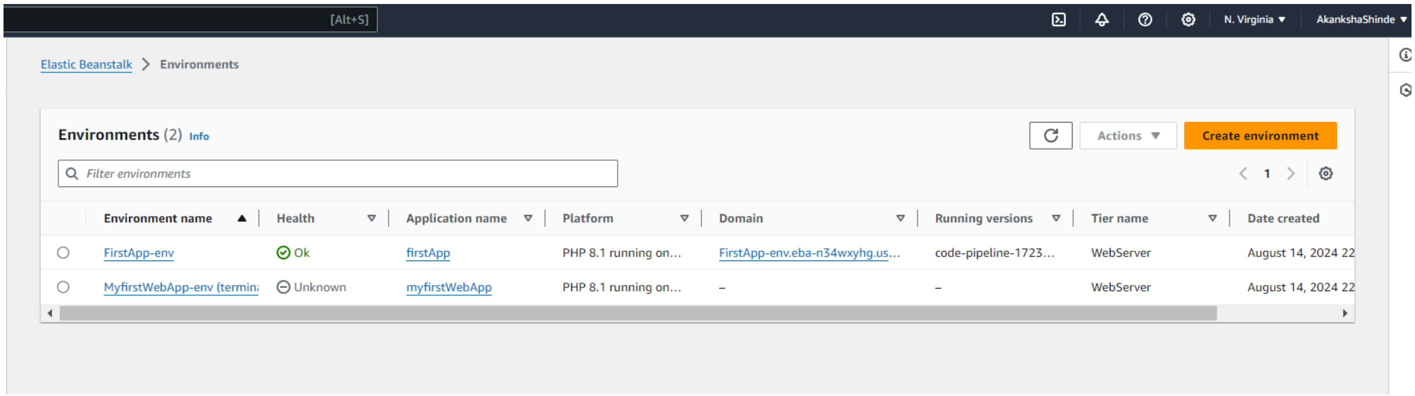
Deploy Succeeded

Pipeline execution ID: 9c6d9b58-bd56-4c62-804e-90d7021f06fe

Deploy
AWS Elastic Beanstalk
Succeeded - Just now

Name: Akanksha Shinde Class: D15C Roll No: 53

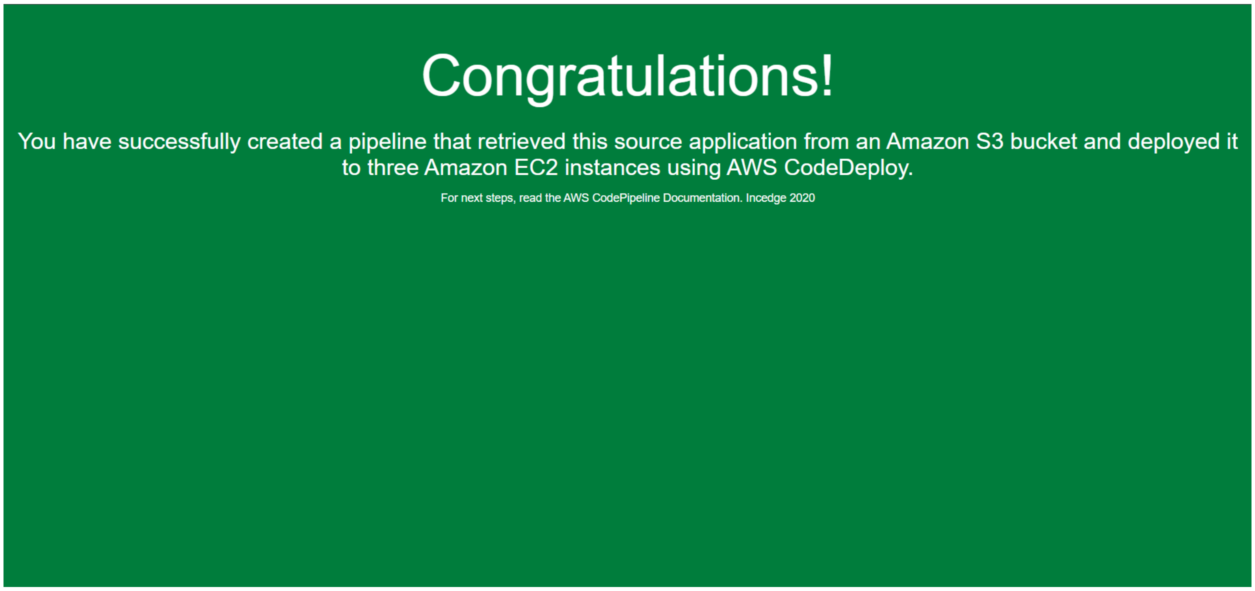
In a few minutes, we will have our pipeline created. Once we have the success message on the Deploy part, we can go ahead and check our URL provided in the EBS environment.



Environments (2) Info

	Environment name ▲	Health	Application name	Platform	Domain	Running versions	Tier name	Date created
○	FirstApp-env	Ok	firstApp	PHP 8.1 running on…	FirstApp-env.eba-n34wxyhg.us…	code-pipeline-1723…	WebServer	August 14, 2024 22
○	MyfirstWebApp-env (termin:	Unknown	myfirstWebApp	PHP 8.1 running on…	–	–	WebServer	August 14, 2024 22

This is the sample website we just created.



Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020

If you can see this, that means that you successfully created an automated software using CodePipeline.