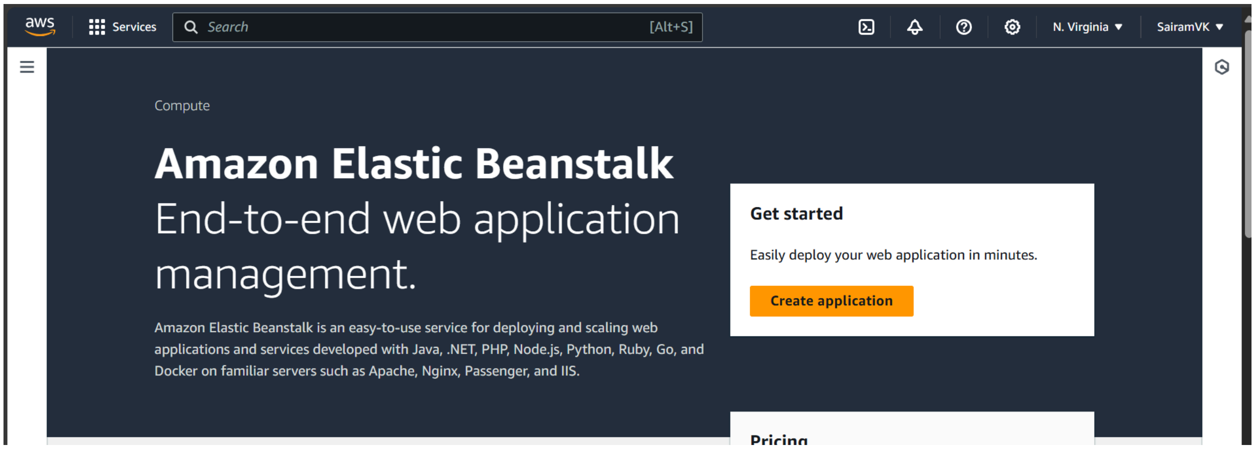
Name: Anish Kulkarni　　Roll No.: 29　　Class: D15C　　AY: 2024-25

Experiment No. 2

Aim: To build your application using AWS Codebuild and deploy on S3/SEBS using AWS CodePipeline, deploy a sample application on an EC2 instance using AWS CodeDeploy.

Steps:-

Step 1: Log into AWS and navigate to ‘Elastic Beanstalk’. Then, click on ‘Create application’.



Compute

Amazon Elastic Beanstalk

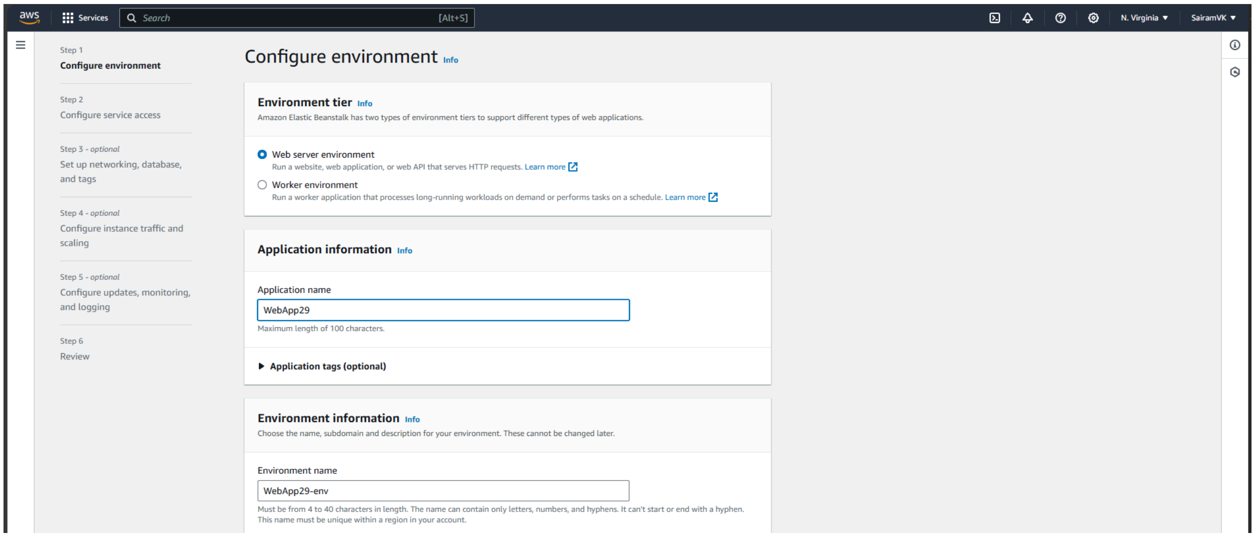
End-to-end web application management.

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.

Step 2: Give your environment a name.



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: WebApp29

Maximum length of 100 characters.

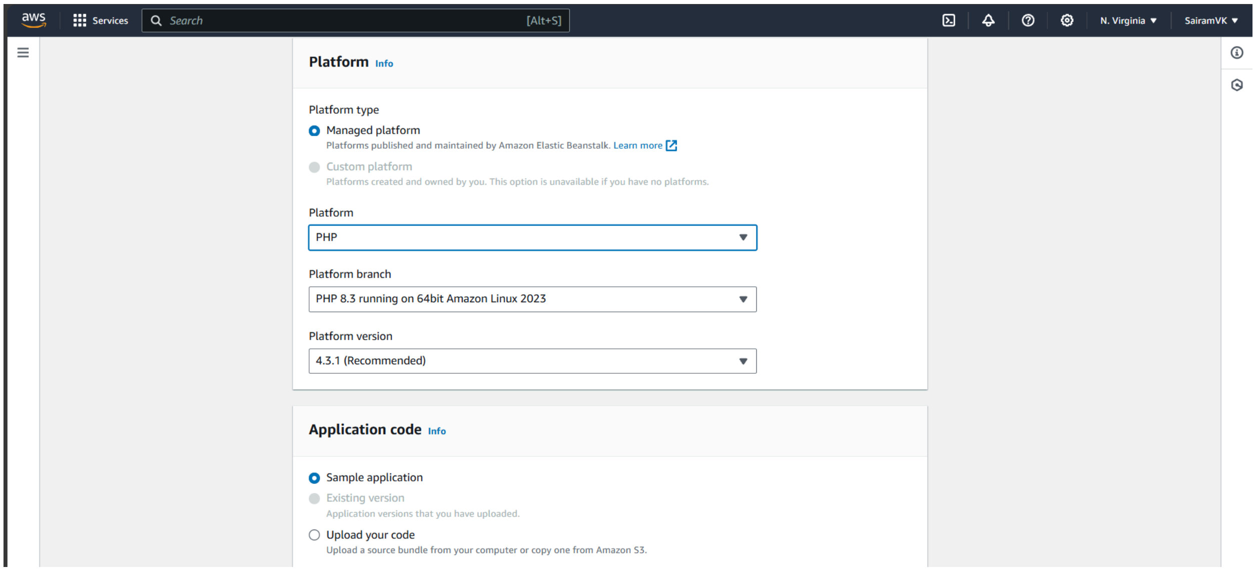
Environment information

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

Environment name: WebApp29-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.

Step 3: In the ‘Platform’ drop-down box, choose PHP. Keep all other settings as default and click on ‘Next’.



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

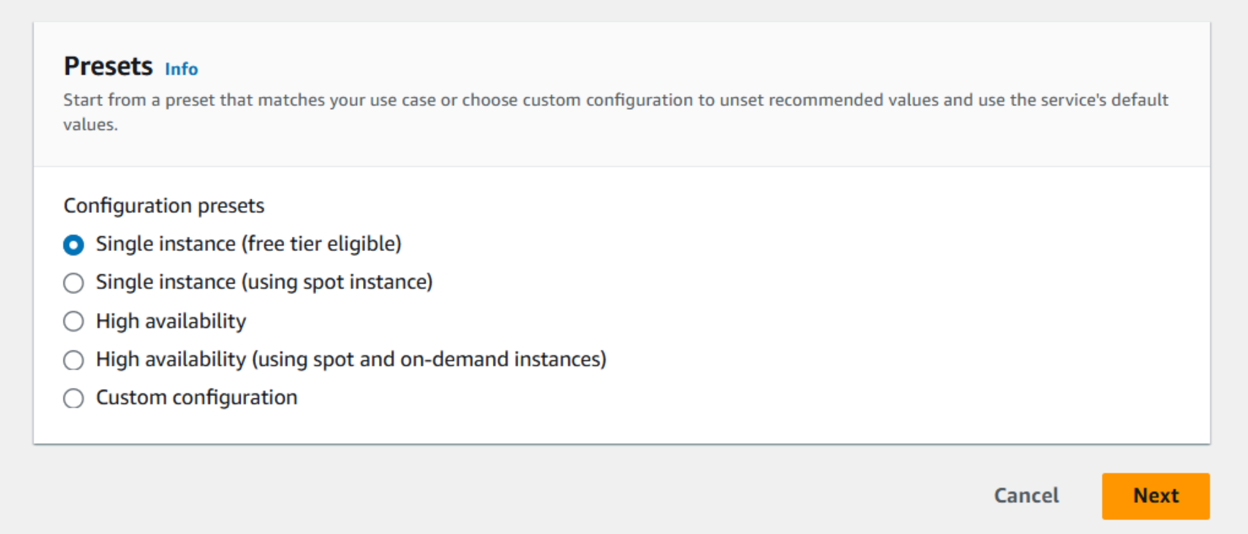
Platform version: 4.3.1 (Recommended)

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)

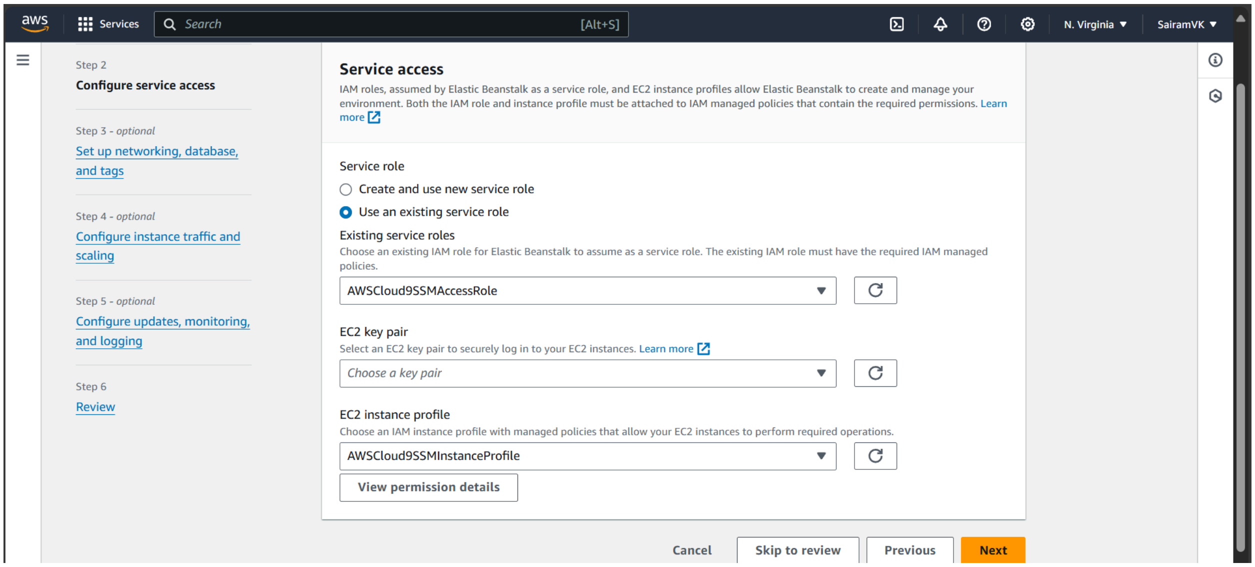
○ Single instance (using spot instance)

○ High availability

○ High availability (using spot and on-demand instances)

○ Custom configuration

Step 4: In the ‘Service access’ section, choose ‘AWSCloud9AAMAccessRole’ in ‘Existing service roles’ and ‘AWSCloud9SSMInstanceProfile’ in ‘EC2 instance profile’.



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

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

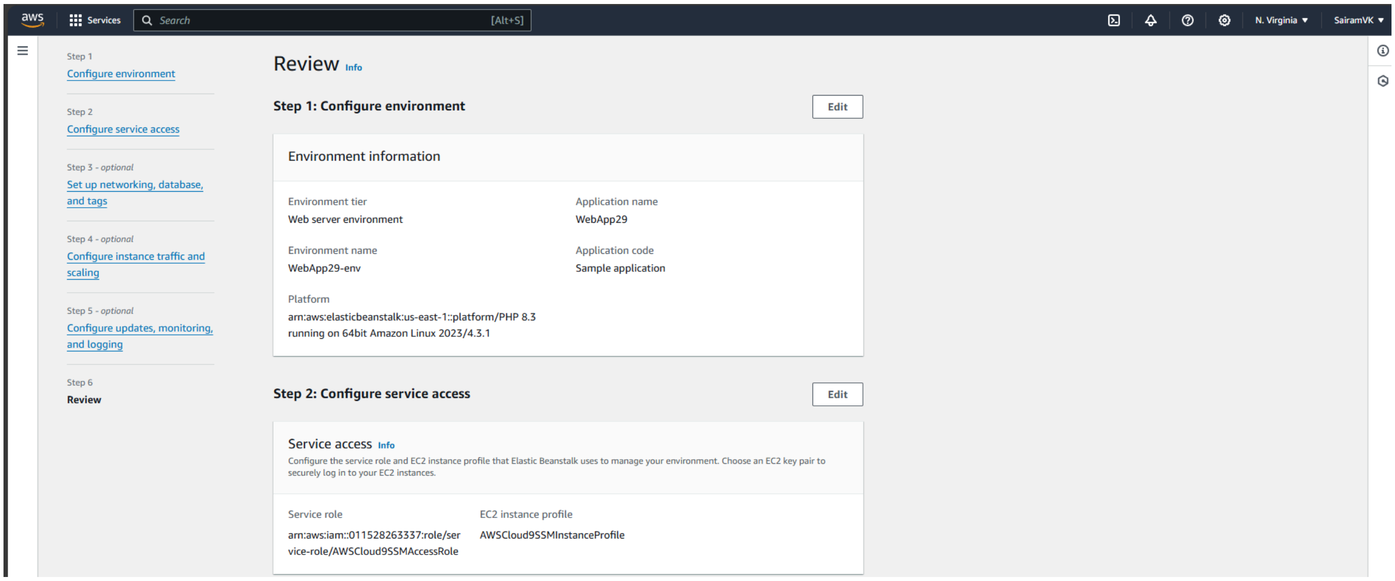
○ Create and use new service role
● Use an existing service role

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.
AWSCloud9SSMAccessRole ▼

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.
AWSCloud9SSMInstanceProfile ▼

Step 5: Review all the environment settings and click on ‘Submit’.



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

Review Info

Step 1: Configure environment Edit

Environment information

Environment tier
Web server environment

Application name
WebApp29

Environment name
WebApp29-env

Application code
Sample application

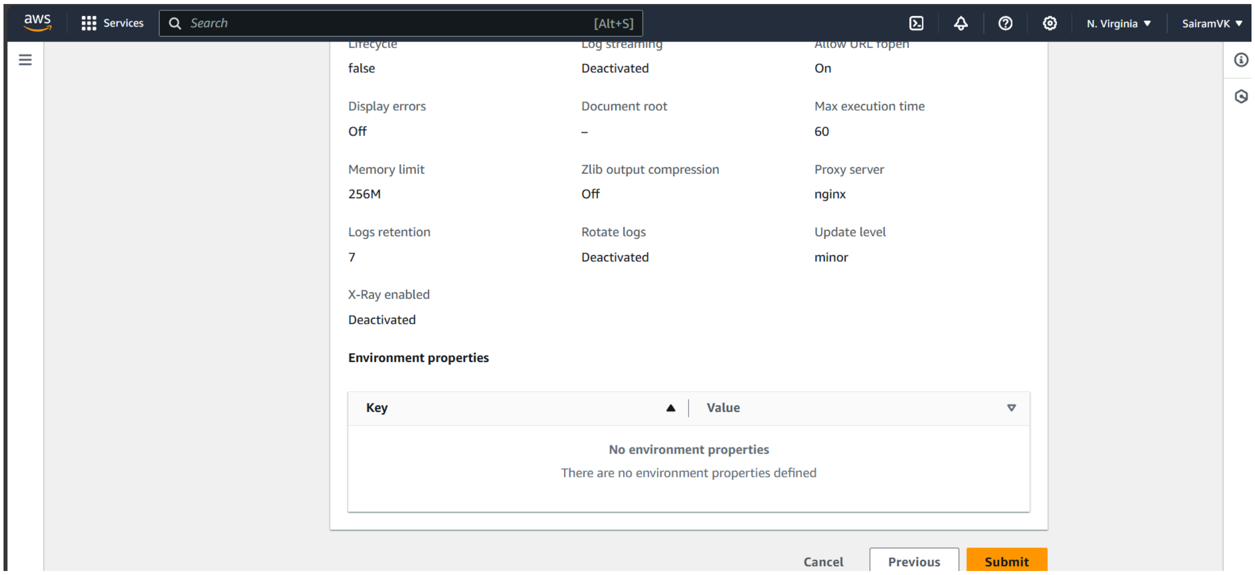
Platform
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1

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::011528263337:role/service-role/AWSCloud9SSMAccessRole

EC2 instance profile
AWSCloud9SSMInstanceProfile



Lifecycle
false

Log streaming
Deactivated

Allow URL fopen
On

Display errors
Off

Document root
–

Max execution time
60

Memory limit
256M

Zlib output compression
Off

Proxy server
nginx

Logs retention
7

Rotate logs
Deactivated

Update level
minor

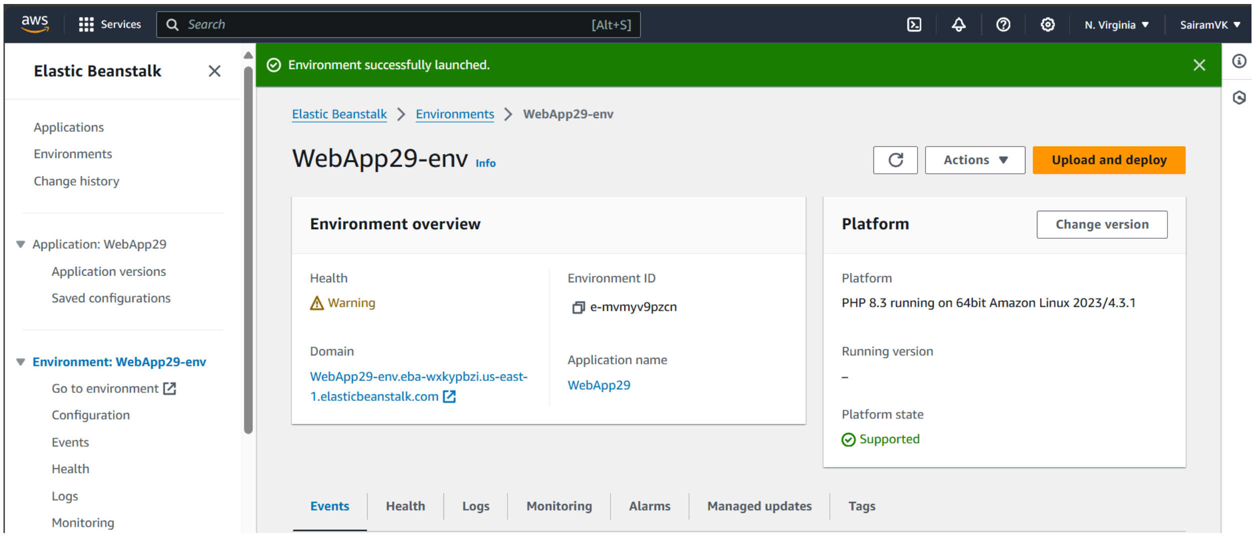
X-Ray enabled
Deactivated

Environment properties

Key	Value

No environment properties

There are no environment properties defined



Environment successfully launched.

WebApp29-env

Environment overview

Health
⚠ Warning

Environment ID
e-mvmyv9pzcn

Domain
WebApp29-env.eba-wxkypbzi.us-east-1.elasticbeanstalk.com

Application name
WebApp29

Platform

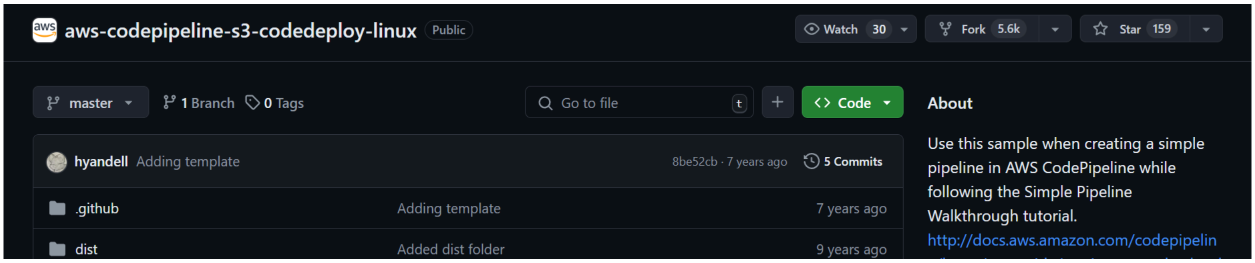
Platform
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1

Running version
–

Platform state
✓ Supported

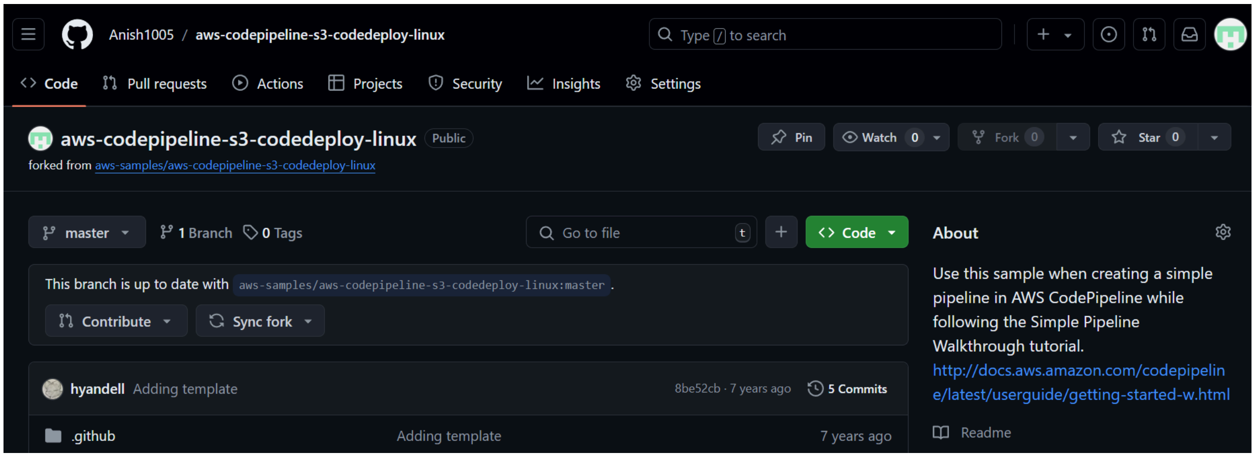
The environment gets created.

Step 6: Go to the github link below and fork the repository into your personal github in order to get the sample code for deploying a file on AWS CodePipeline.
https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux



Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.
http://docs.aws.amazon.com/codepipelin

hyandell Adding template	8be52cb · 7 years ago	5 Commits
.github	Adding template	7 years ago
dist	Added dist folder	9 years ago



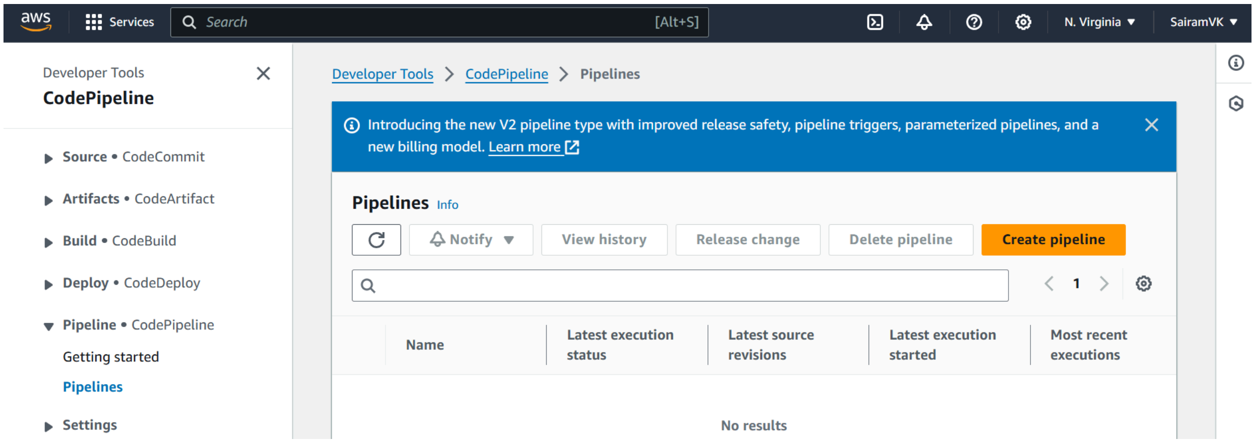
forked from aws-samples/aws-codepipeline-s3-codedeploy-linux

This branch is up to date with `aws-samples/aws-codepipeline-s3-codedeploy-linux:master`.

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.
http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html

hyandell Adding template	8be52cb · 7 years ago	5 Commits
.github	Adding template	7 years ago

Step 7: Navigate to CodePipeline and click on ‘Create Pipeline’.

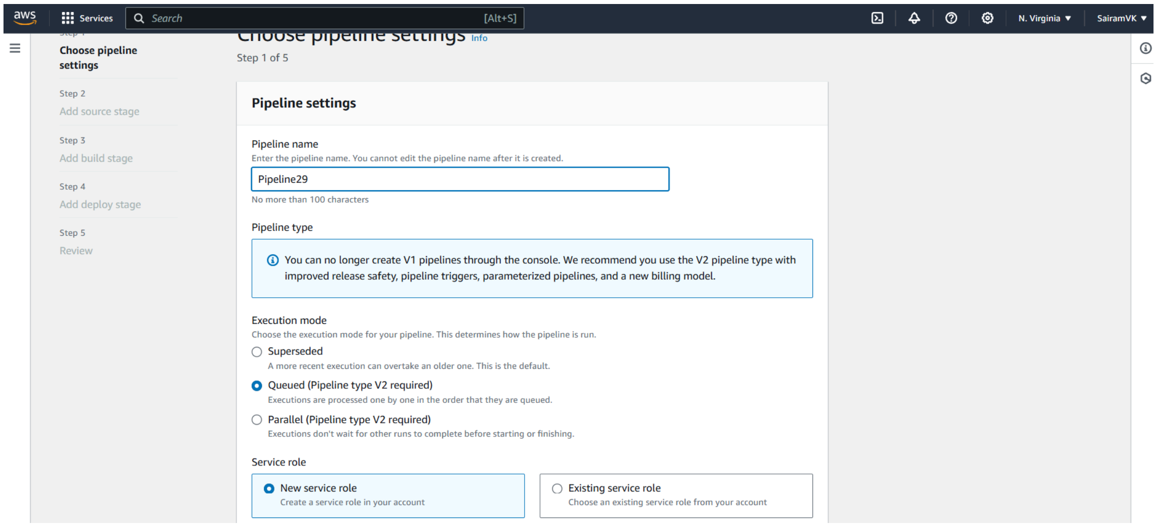


Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more

Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions

No results

Step 8: Give your pipeline a name. A new service role is also created with the name of the pipeline.



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.
Pipeline29
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.

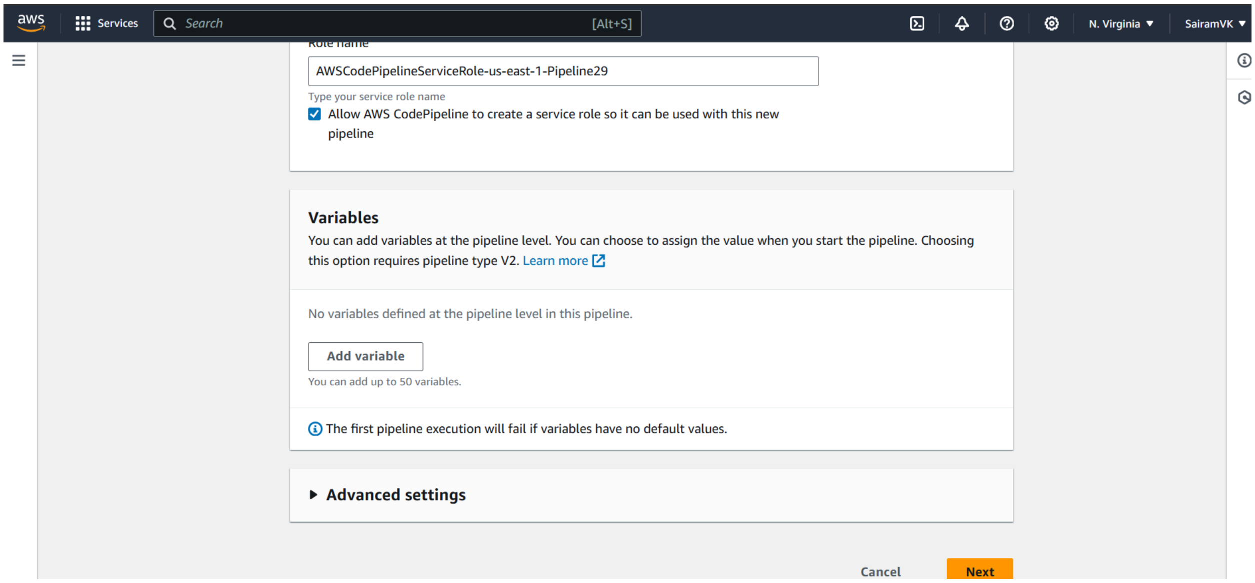
● Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

○ 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-Pipeline29
Type your service role name

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

Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. Learn more

No variables defined at the pipeline level in this pipeline.

Add variable

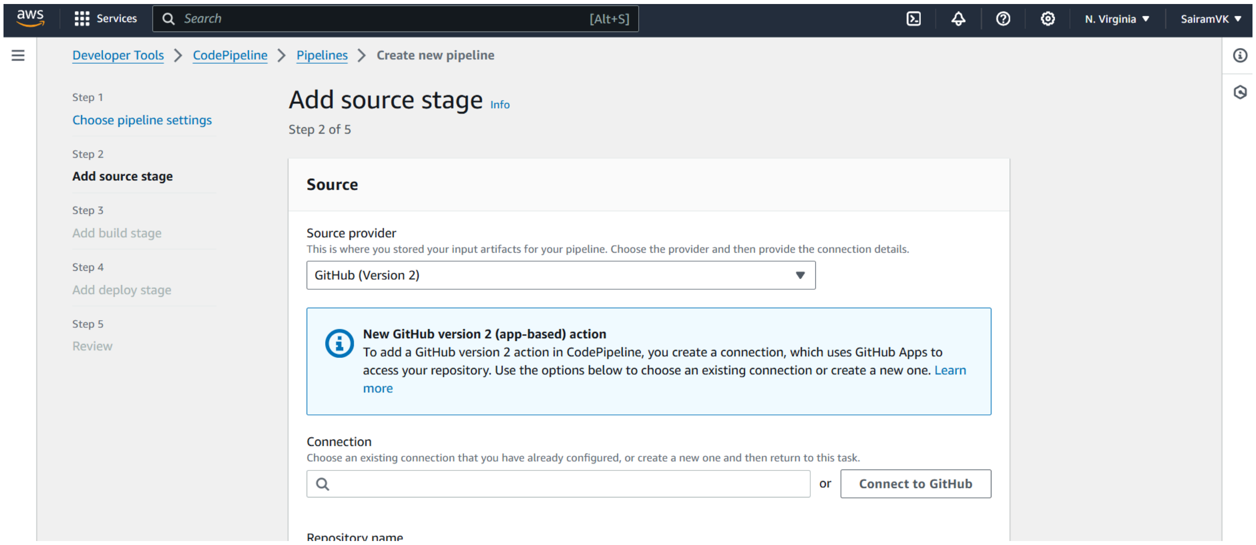
You can add up to 50 variables.

The first pipeline execution will fail if variables have no default values.

▶ **Advanced settings**

Cancel Next

Step 9: Select Github (Version 2) as source provider and click on ‘Connect to Github’ to connect the pipeline to your Github.



Add source stage

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

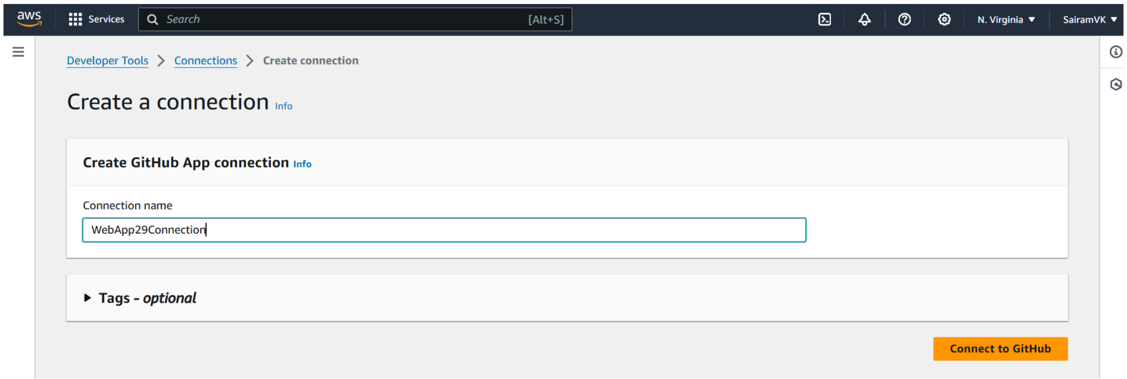
Connection

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

or

Repository name

Step 10: Give your connection a name and click on ‘Connect to Github’. Then, either provide a link for connection or install the app if it is your first time.

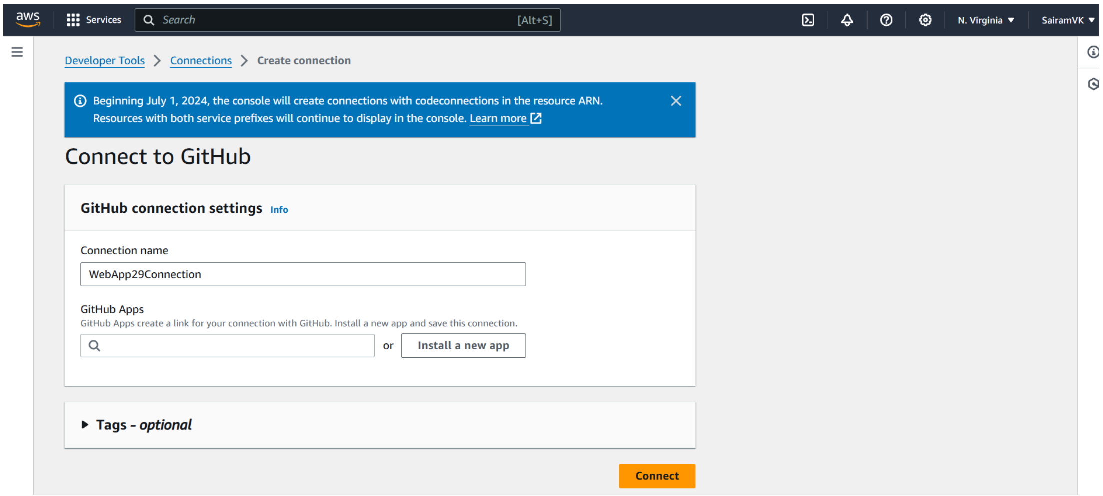


Create a connection

Create GitHub App connection

Connection name

WebApp29Connection



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

Connection name

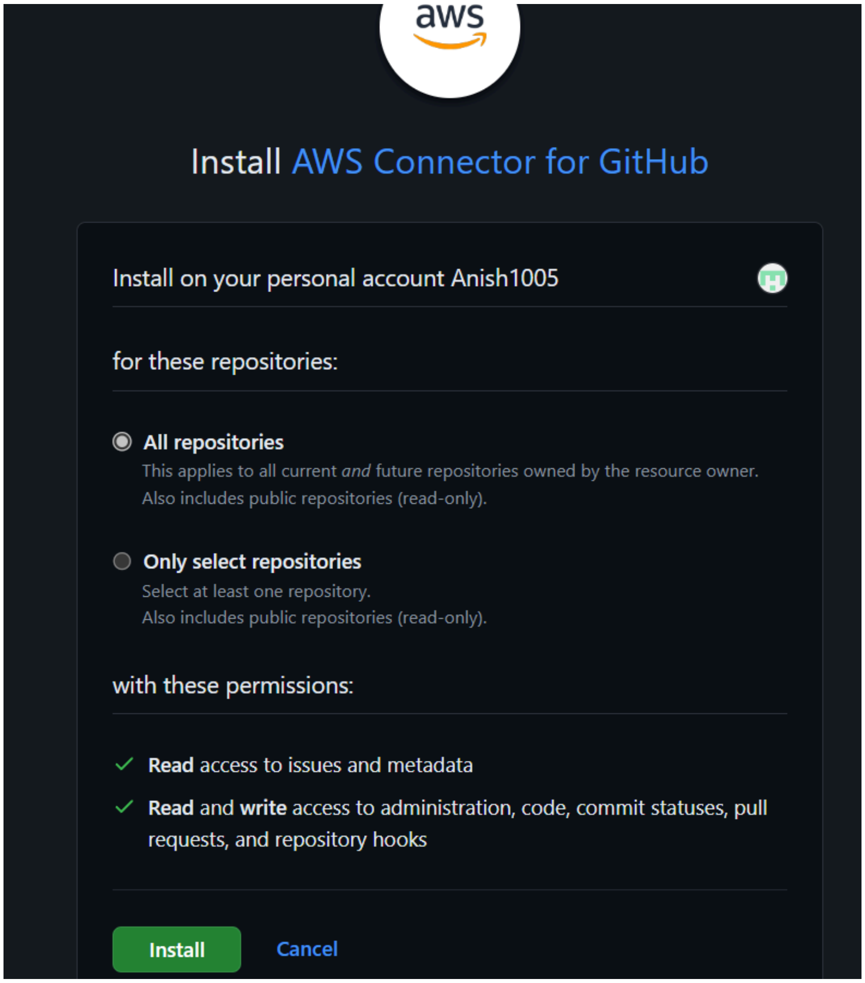
WebApp29Connection

GitHub Apps

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

or

Step 11: Install AWS Connector for Github.



Install AWS Connector for GitHub

Install on your personal account Anish1005

for these repositories:

● **All repositories**
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

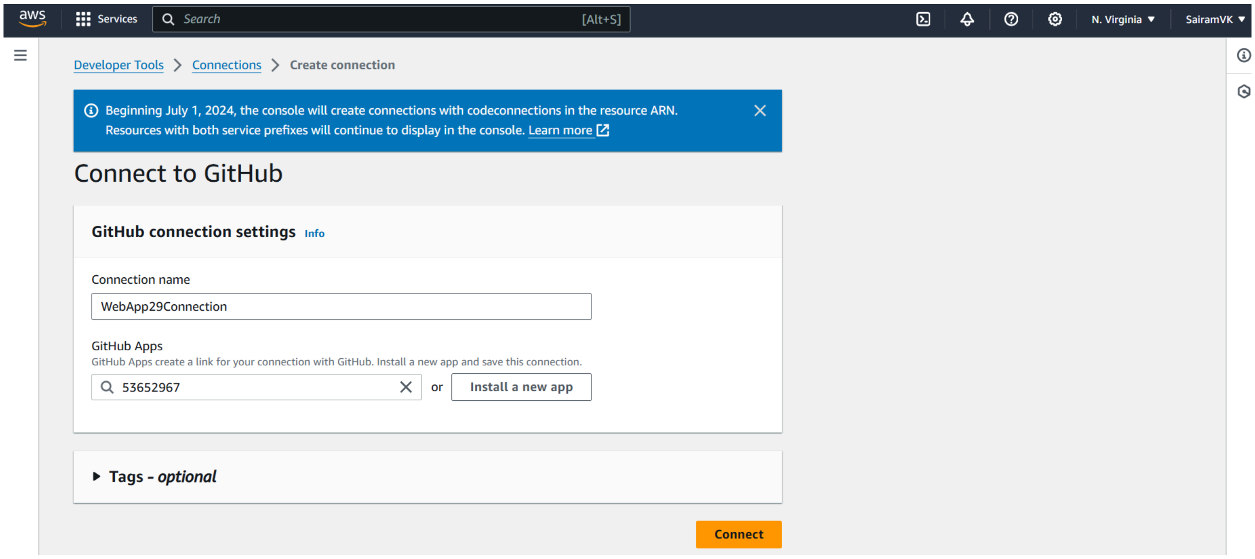
○ **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

with these permissions:

✓ **Read** access to issues and metadata

✓ **Read** and **write** access to administration, code, commit statuses, pull requests, and repository hooks

Step 12: Once installation is done, the text field is filled automatically. Click on 'Connect'.



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

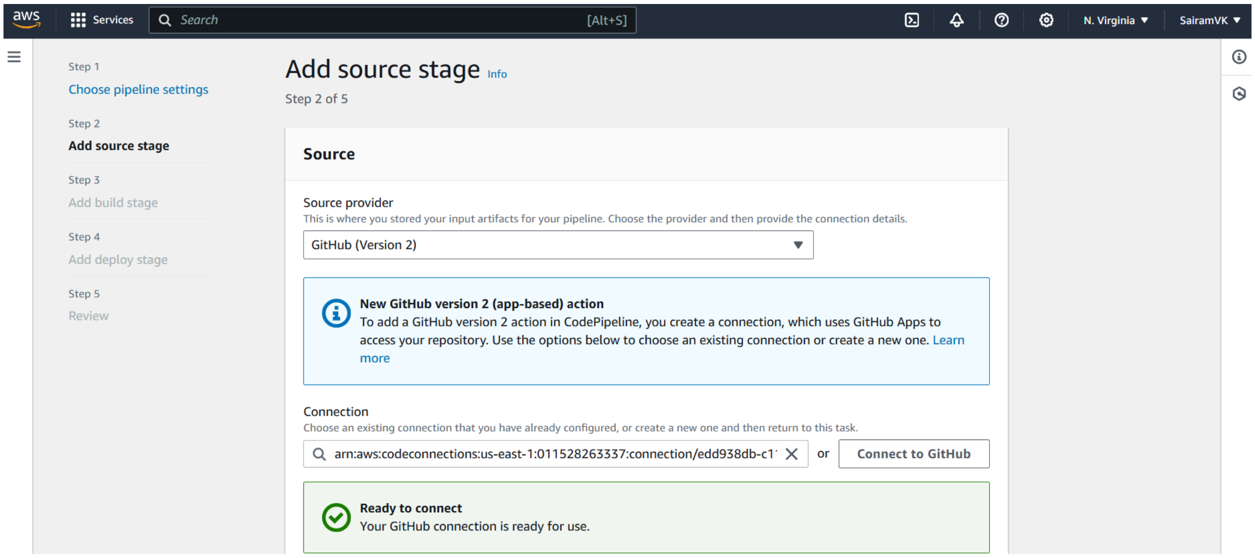
Connect to GitHub

GitHub connection settings Info

Connection name
WebApp29Connection

GitHub Apps
GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.
53652967 or Install a new app

▸ **Tags - *optional***



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 2)

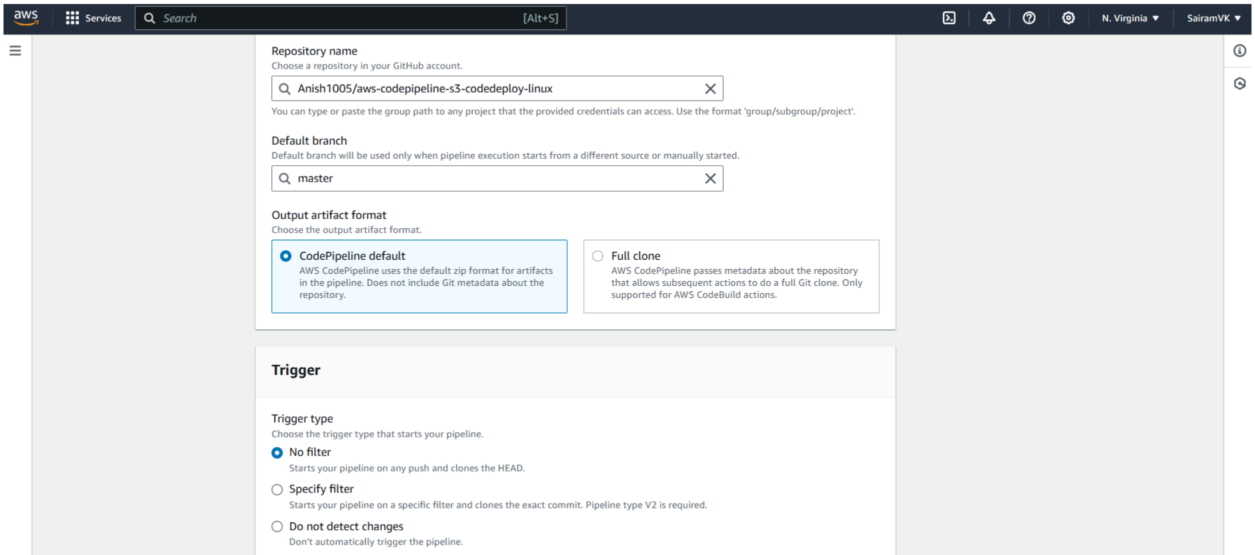
New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

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-1:011528263337:connection/edd938db-c1 or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

AWS shows that the Github connection is ready to use.

Step 13: Select the repository that you forked the sample code to be deployed to and choose the branch on which the files are present (‘master’ is set as default). Also set the trigger type as ‘no filter’.



Repository name
Choose a repository in your GitHub account.
Anish1005/aws-codepipeline-s3-codedeploy-linux
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.
master

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

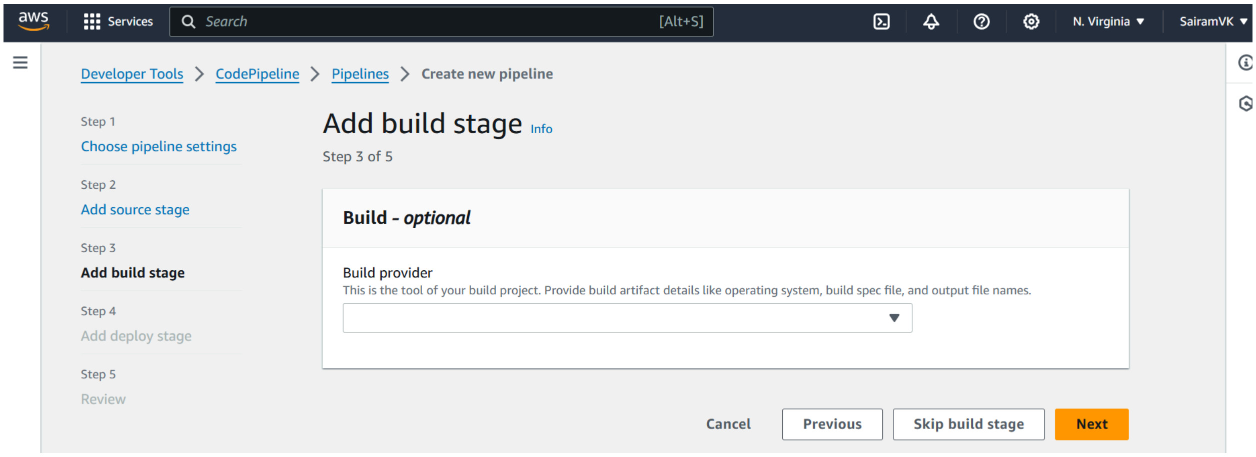
Trigger type
Choose the trigger type that starts your pipeline.

● 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.

Step 14: Skip the build stage and go to deploy stage.



Add build stage Info

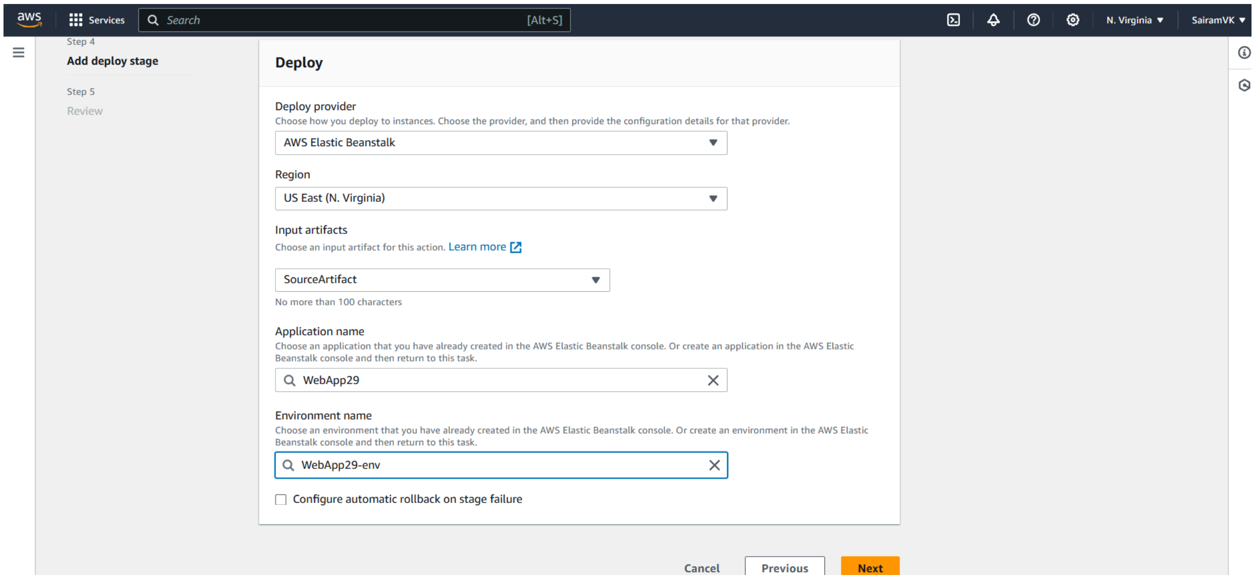
Step 3 of 5

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.

Step 15: Choose 'AWS Elastic Beanstalk' as deploy provider and input artifacts as 'SourceArtifact'. Then, enter the names of your application and environment.



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

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.

WebApp29

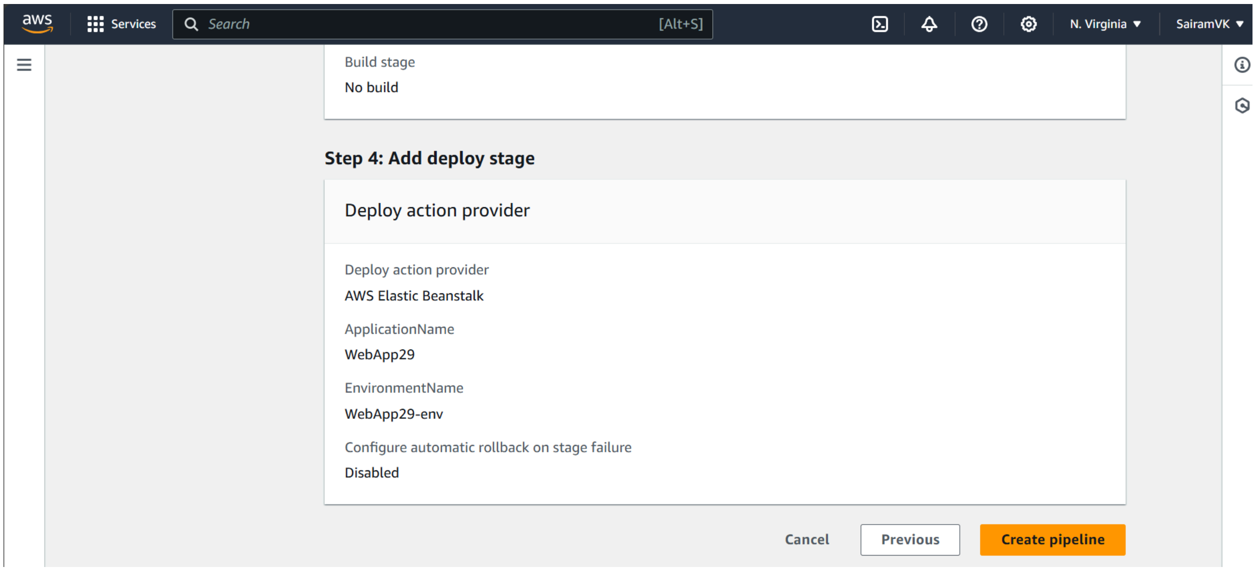
Environment name

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.

WebApp29-env

□ Configure automatic rollback on stage failure

Step 16: Review all pipeline information and click on ‘Create pipeline’.



Build stage
No build

Step 4: Add deploy stage

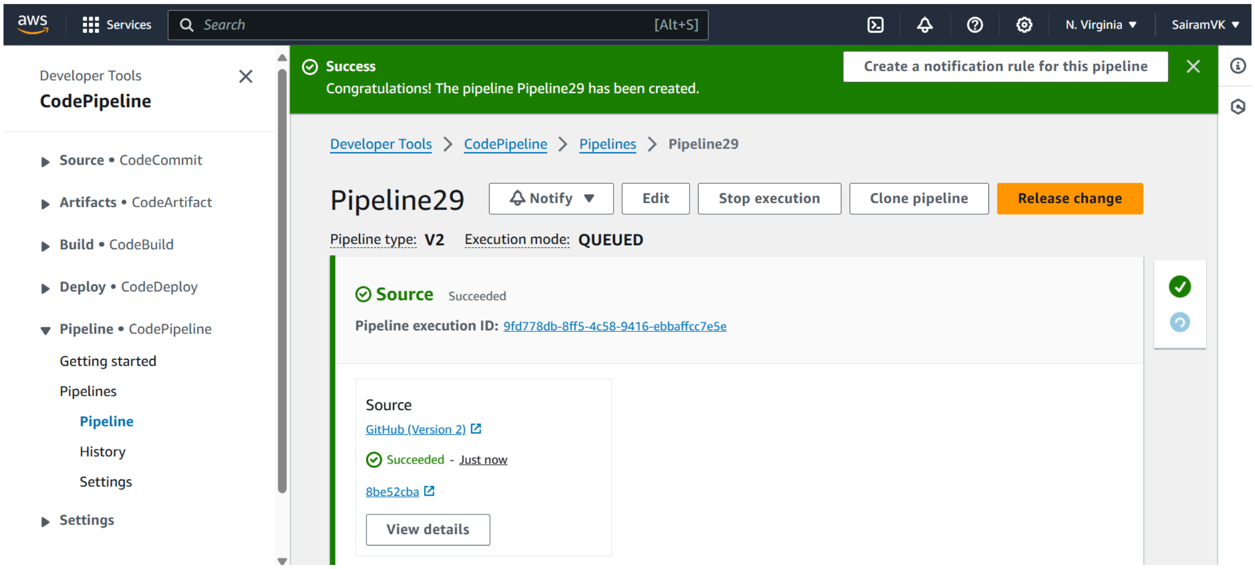
Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
WebApp29

EnvironmentName
WebApp29-env

Configure automatic rollback on stage failure
Disabled



Success
Congratulations! The pipeline Pipeline29 has been created.

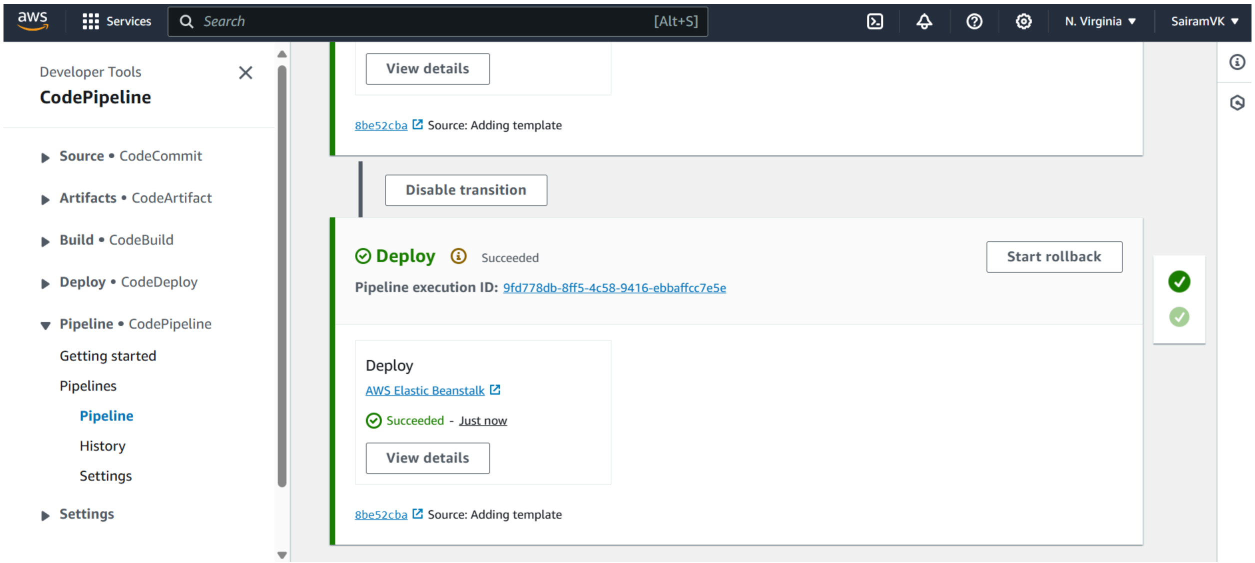
Developer Tools > CodePipeline > Pipelines > Pipeline29

Pipeline29

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

Source Succeeded
Pipeline execution ID: 9fd778db-8ff5-4c58-9416-ebbaffcc7e5e

Source
GitHub (Version 2)
Succeeded - Just now
8be52cba



8be52cba Source: Adding template

Deploy Succeeded

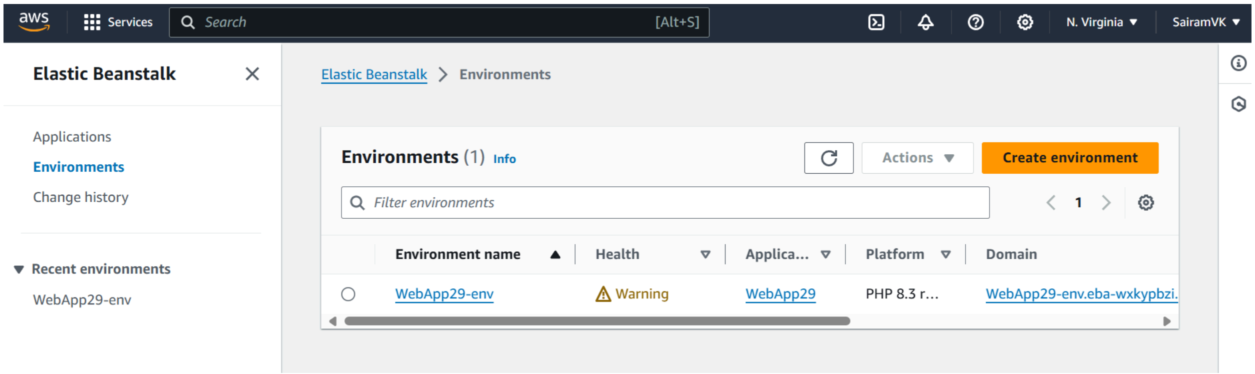
Pipeline execution ID: 9fd778db-8ff5-4c58-9416-ebbaffcc7e5e

Deploy
AWS Elastic Beanstalk
Succeeded - Just now

8be52cba Source: Adding template

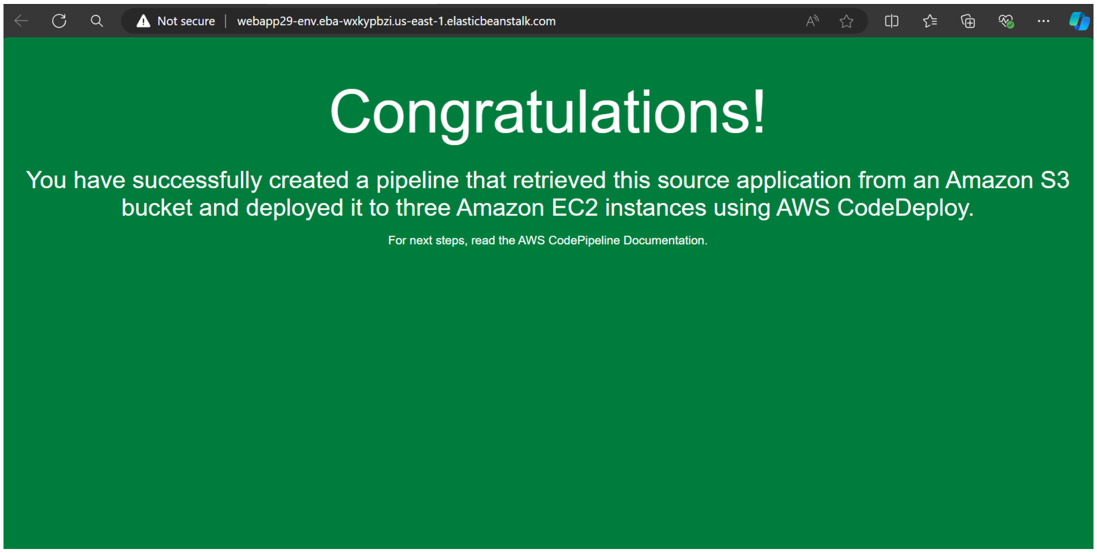
The pipeline is created and deployment is complete.

Step 17: Clicking on ‘AWS Elastic Beanstalk’ under ‘Deploy’ redirects you to the environments screen of Elastic Beanstalk. Click on the link under ‘Domain’.



	Environment name	Health	Applica...	Platform	Domain
○	WebApp29-env	⚠ Warning	WebApp29	PHP 8.3 r...	WebApp29-env.eba-wxkypbzi.

Step 18: The sample website is displayed showing that the website was successfully hosted.

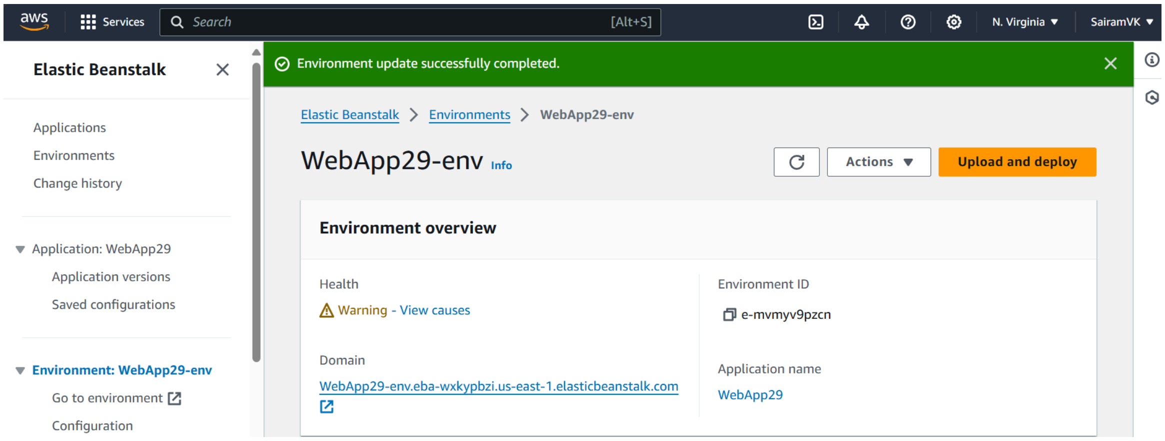


Congratulations!

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.

Step 19: Making changes to the index file in the github will update the website and changes made to the website are visible.



Environment update successfully completed.

WebApp29-env

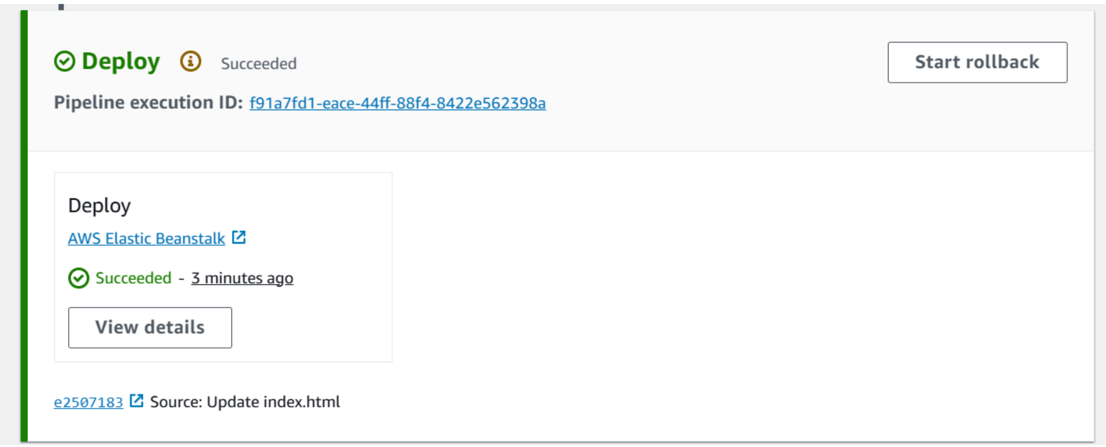
Environment overview

Health: Warning - View causes

Environment ID: e-mvmyv9pzcn

Domain: WebApp29-env.eba-wxkypbzi.us-east-1.elasticbeanstalk.com

Application name: WebApp29



Deploy Succeeded

Pipeline execution ID: f91a7fd1-eace-44ff-88f4-8422e562398a

Deploy

AWS Elastic Beanstalk

Succeeded - 3 minutes ago

e2507183 Source: Update index.html