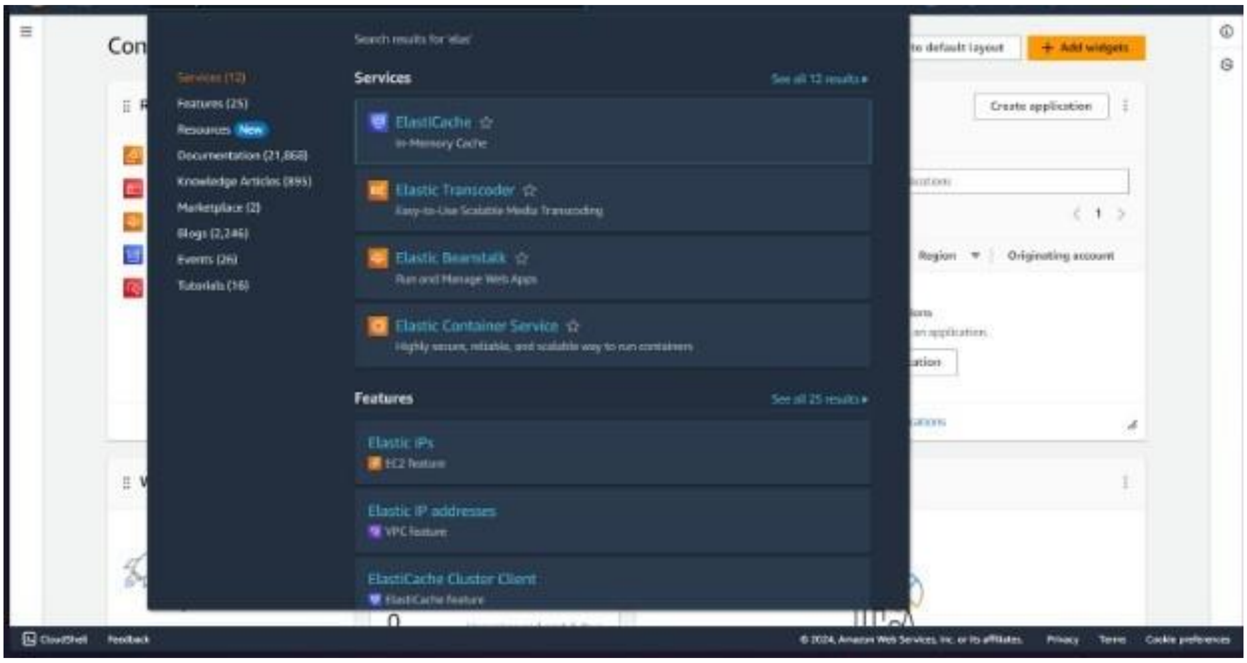
Aim**:** To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Step 1:Start by logging into your AWS console. Once you're in, use the search bar near the services section to look for "Elastic Beanstalk."



Search results for 'elas'

Services (12)
Features (25)
Resources New
Documentation (21,868)
Knowledge Articles (895)
Marketplace (2)
Blogs (2,246)
Events (26)
Tutorials (16)

Services — See all 12 results

ElastiCache — In-Memory Cache

Elastic Transcoder — Easy-to-Use Scalable Media Transcoding

Elastic Beanstalk — Run and Manage Web Apps

Elastic Container Service — Highly secure, reliable, and scalable way to run containers

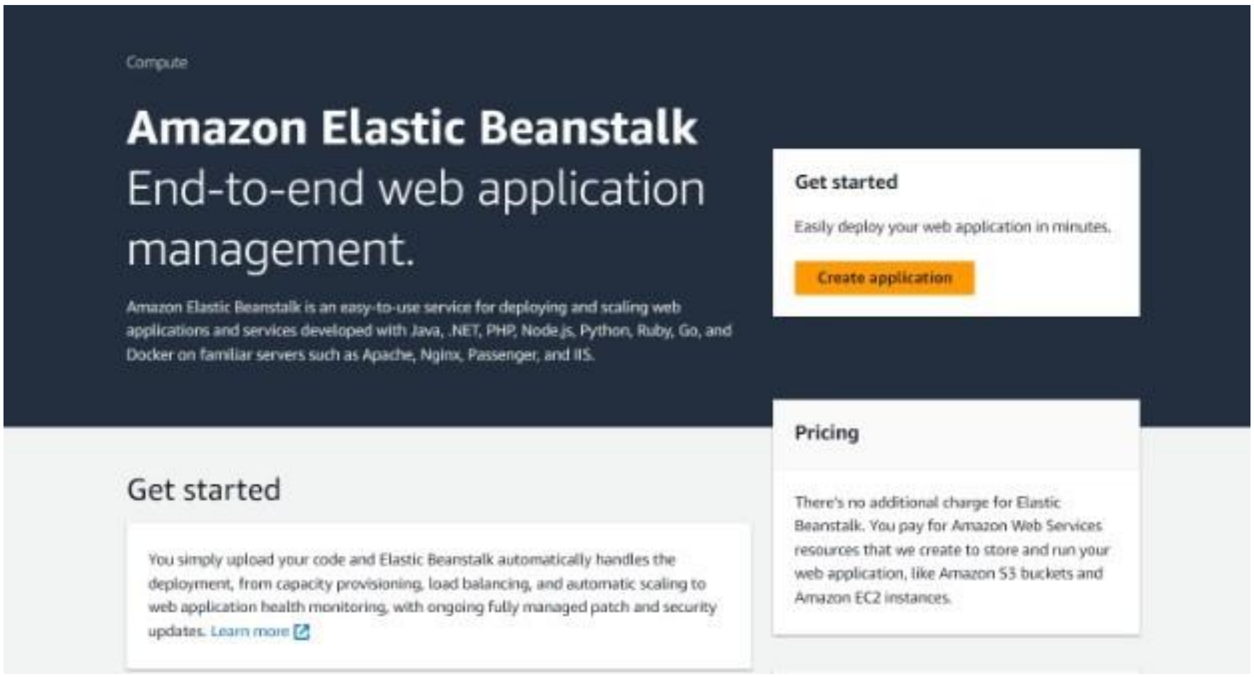
Features — See all 25 results

Elastic IPs — EC2 feature

Elastic IP addresses — VPC feature

ElastiCache Cluster Client — ElastiCache feature

Step 2:After opening Elastic Beanstalk, you'll see an option to "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.

Create application

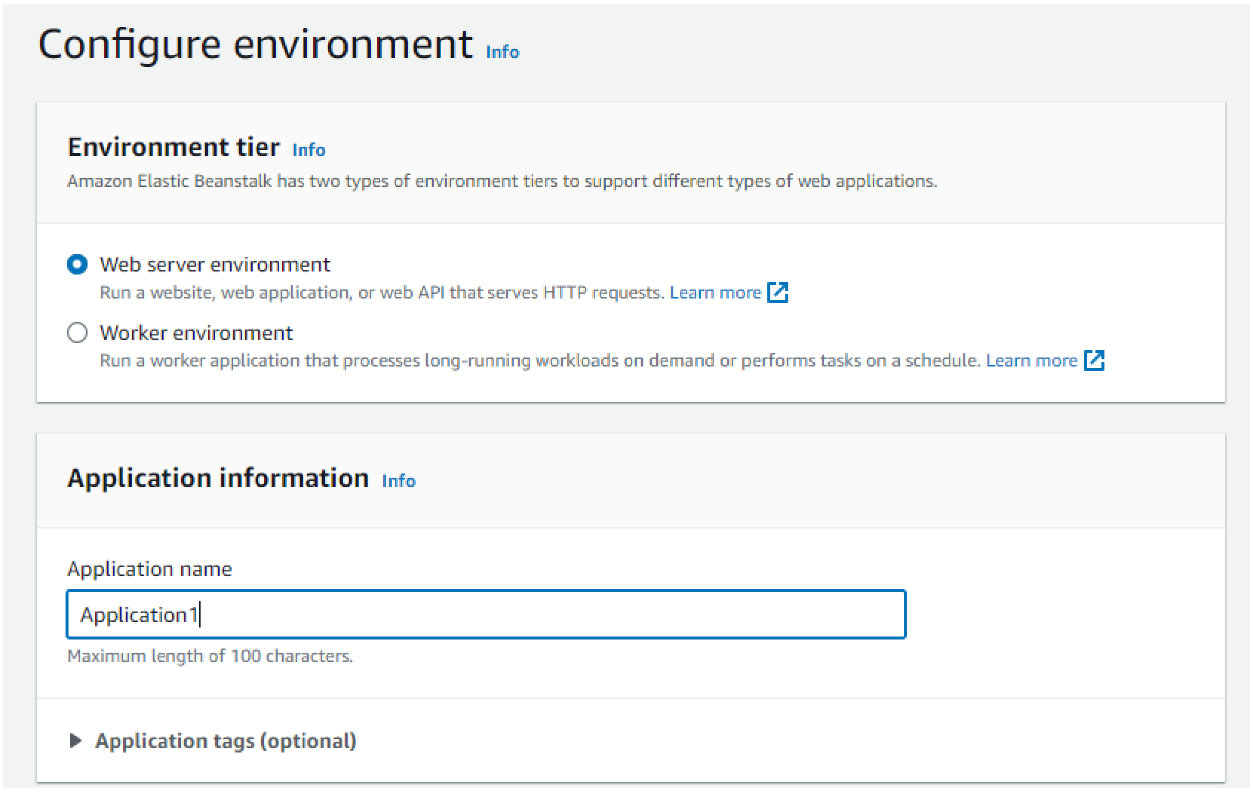
Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. Learn more

Step 3: Select the "Web Service Environment" option for the environment setup. Provide an appropriate name for the application.



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

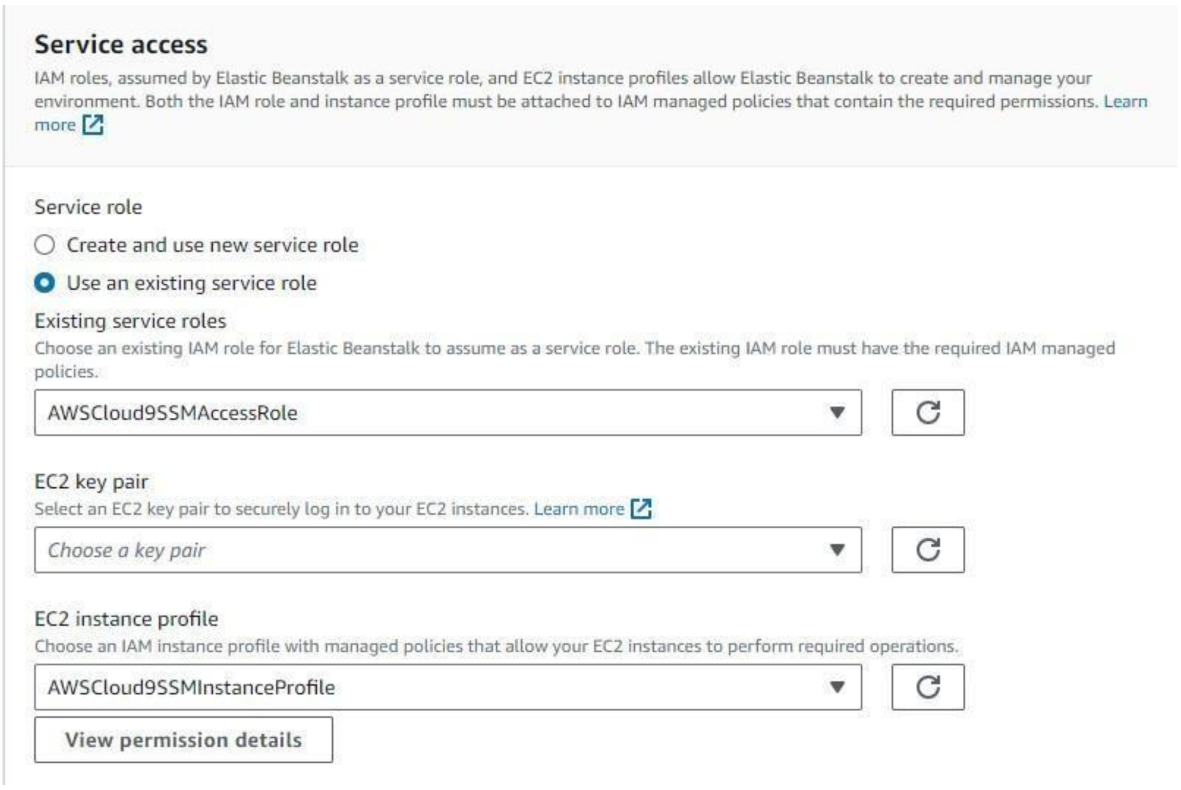
Application information Info

Application name: Application1

Maximum length of 100 characters.

▶ Application tags (optional)

Step 4: In the "Service Access" section, choose "Use an existing service role" to utilize pre-existing permissions.



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

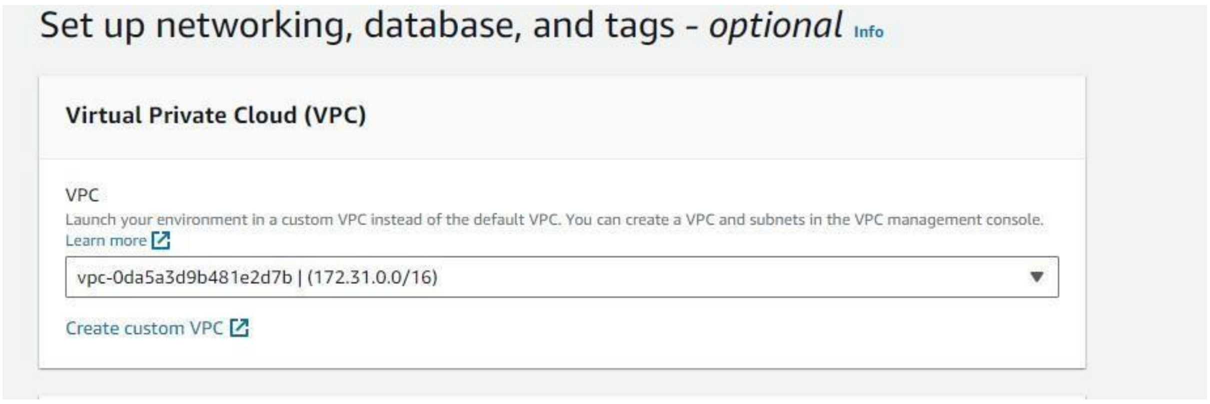
EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. Learn more

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: In the VPC (Virtual Private Cloud) dropdown, select `vpc-0da5a3d9b481e2d7b (172.31.0.0/16)` to specify the network for the environment.



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

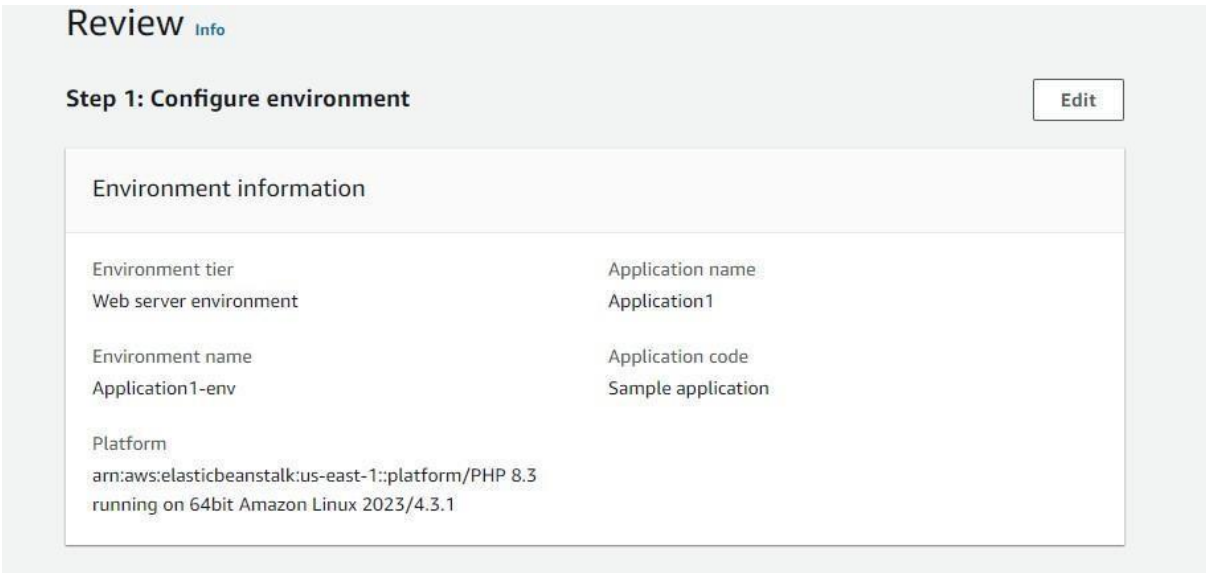
Virtual Private Cloud (VPC)

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-0da5a3d9b481e2d7b | (172.31.0.0/16)

Create custom VPC

Step 6: Click on the "Review" button to double-check all selections before proceeding.



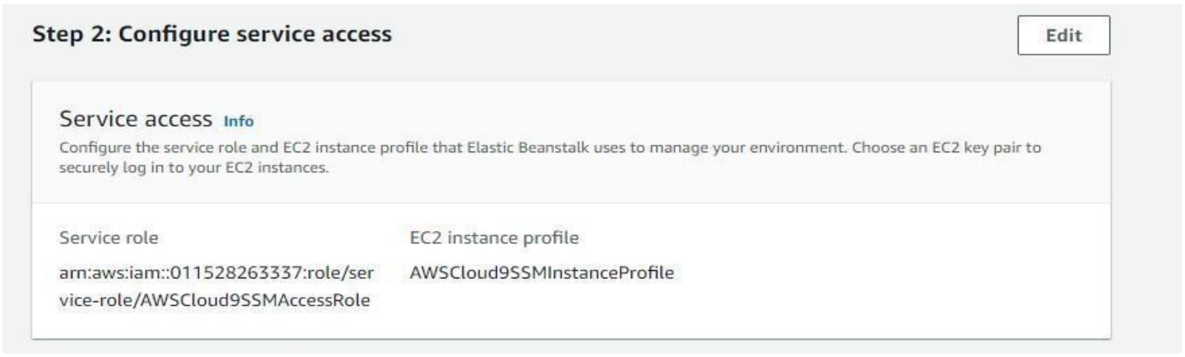
Review Info

Step 1: Configure environment

Environment information

Environment tier Web server environment	Application name Application1
Environment name Application1-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	

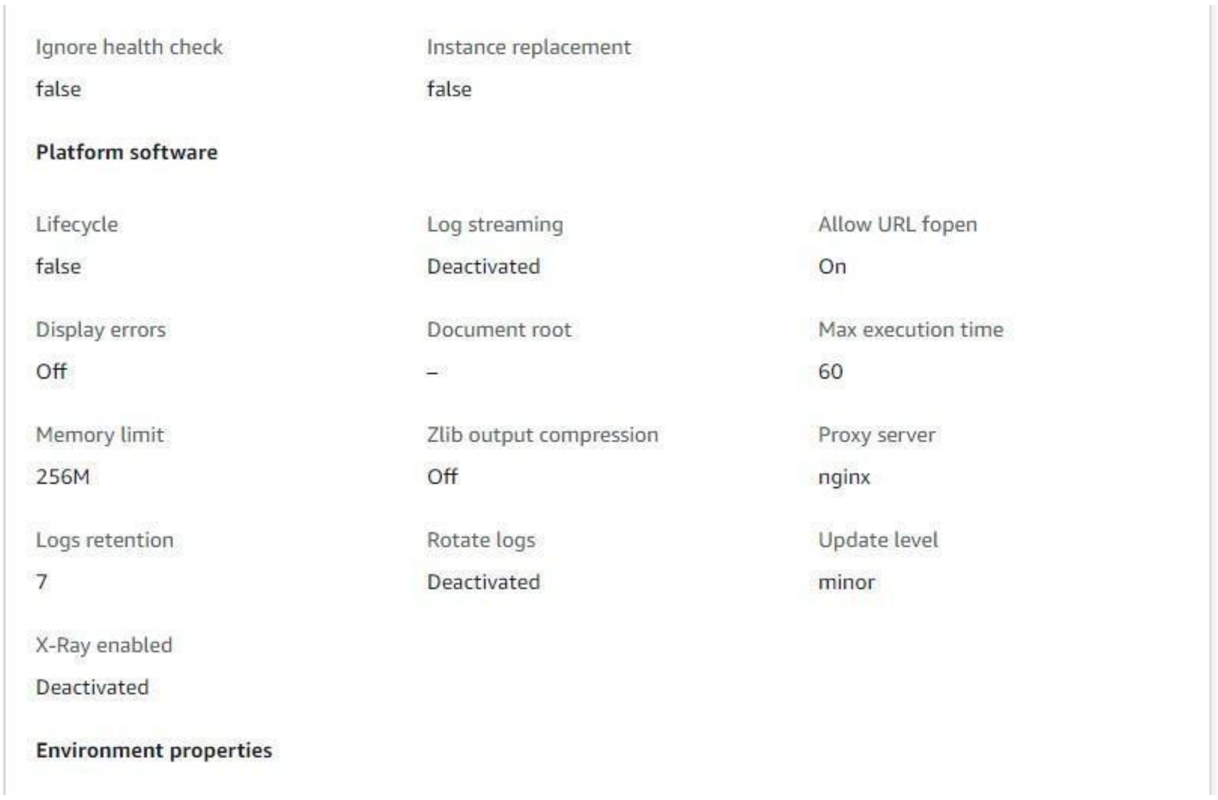
Step 7: After reviewing, click "Configure Service Access" to finalize the settings, ensuring the environment has the correct permissions.



Step 2: Configure service access

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



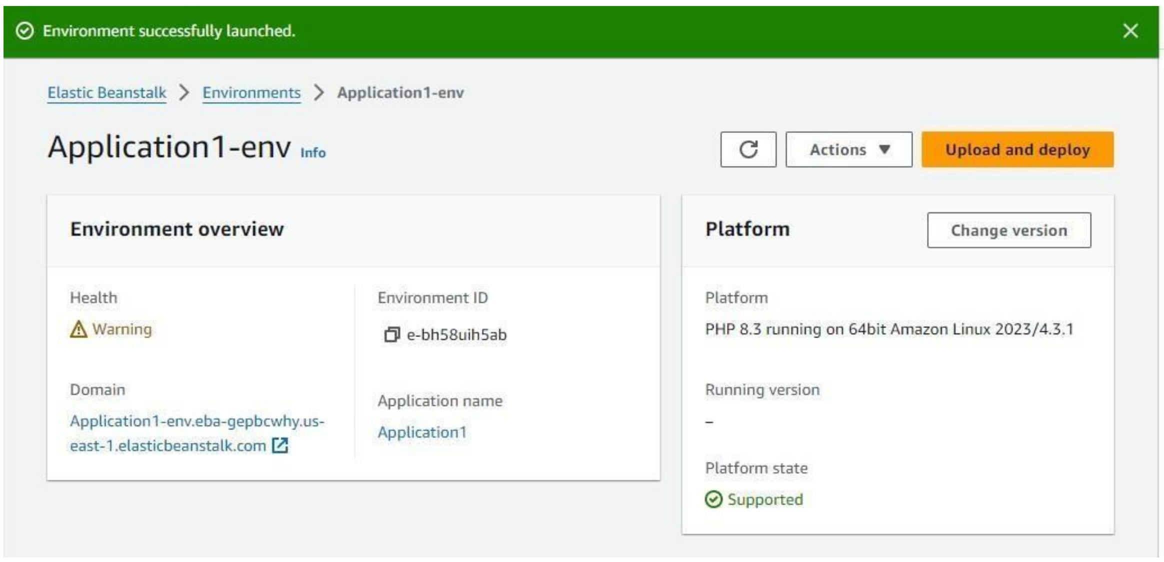
Ignore health check	Instance replacement	
false	false	

Platform software

Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	Document root	Max execution time
Off	–	60
Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		

Environment properties

Step 8: The environment creation process is successfully set up.



Environment successfully launched.

Elastic Beanstalk > Environments > Application1-env

Application1-env Info

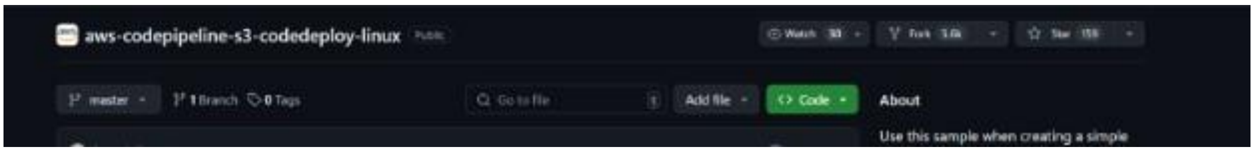
Environment overview

Health: Warning
Environment ID: e-bh58uih5ab
Domain: Application1-env.eba-gepbcwhy.us-east-1.elasticbeanstalk.com
Application name: Application1

Platform

Platform: PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1
Running version: –
Platform state: Supported

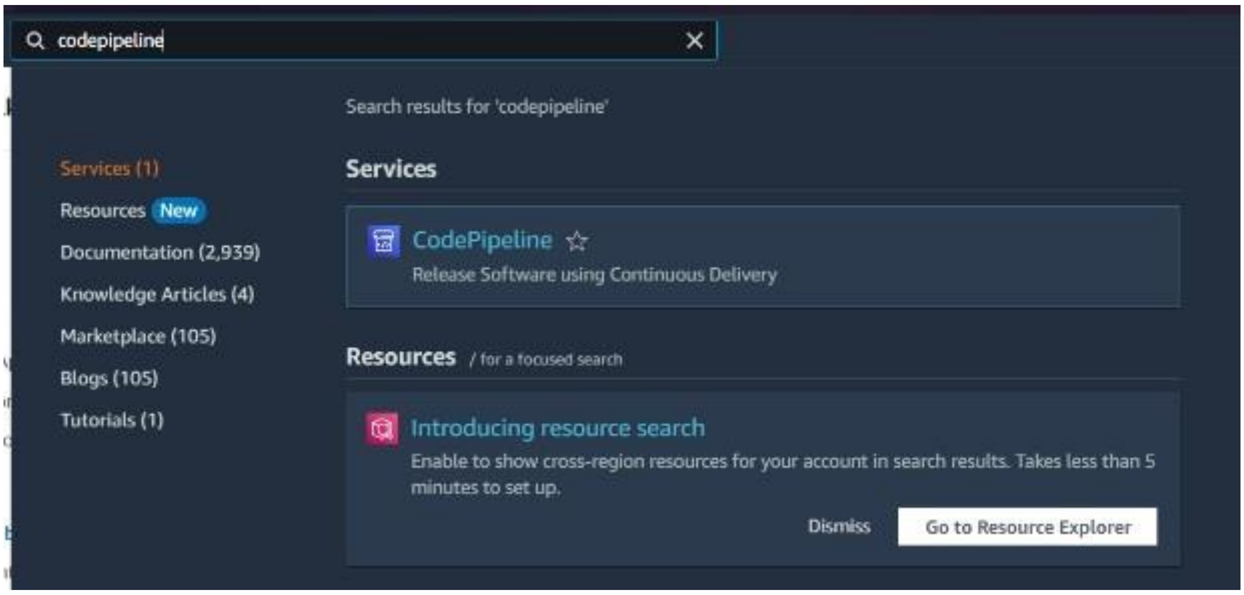
Step 9: Go to a GitHub account and use this link to find the necessary repository: https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux



Use this sample when creating a simple

Step 10: On the repository page, click the "Fork" button next to the code to create a copy of the repository in the GitHub account.

Step 11: Search for "CodePipeline," and click on it in the results.



Search results for 'codepipeline'

Services

CodePipeline

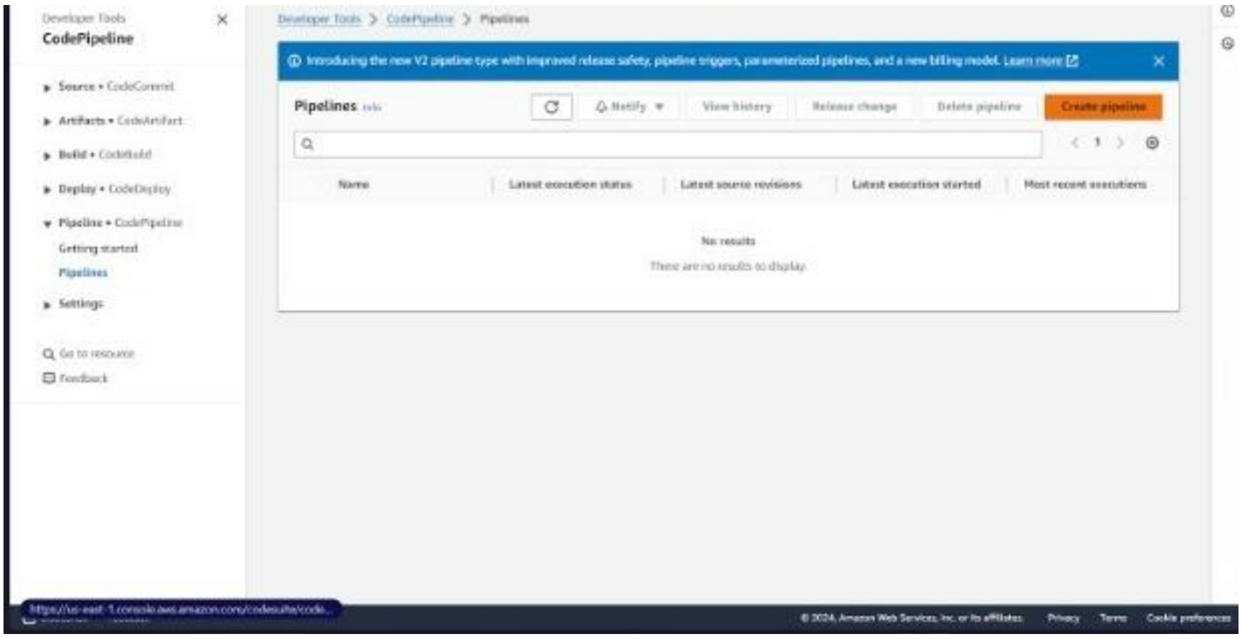
Release Software using Continuous Delivery

Resources / for a focused search

Introducing resource search

Enable to show cross-region resources for your account in search results. Takes less than 5 minutes to set up.

Step 12: In the CodePipeline interface, locate the "Create Pipeline" button and click it to start setting up the new 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

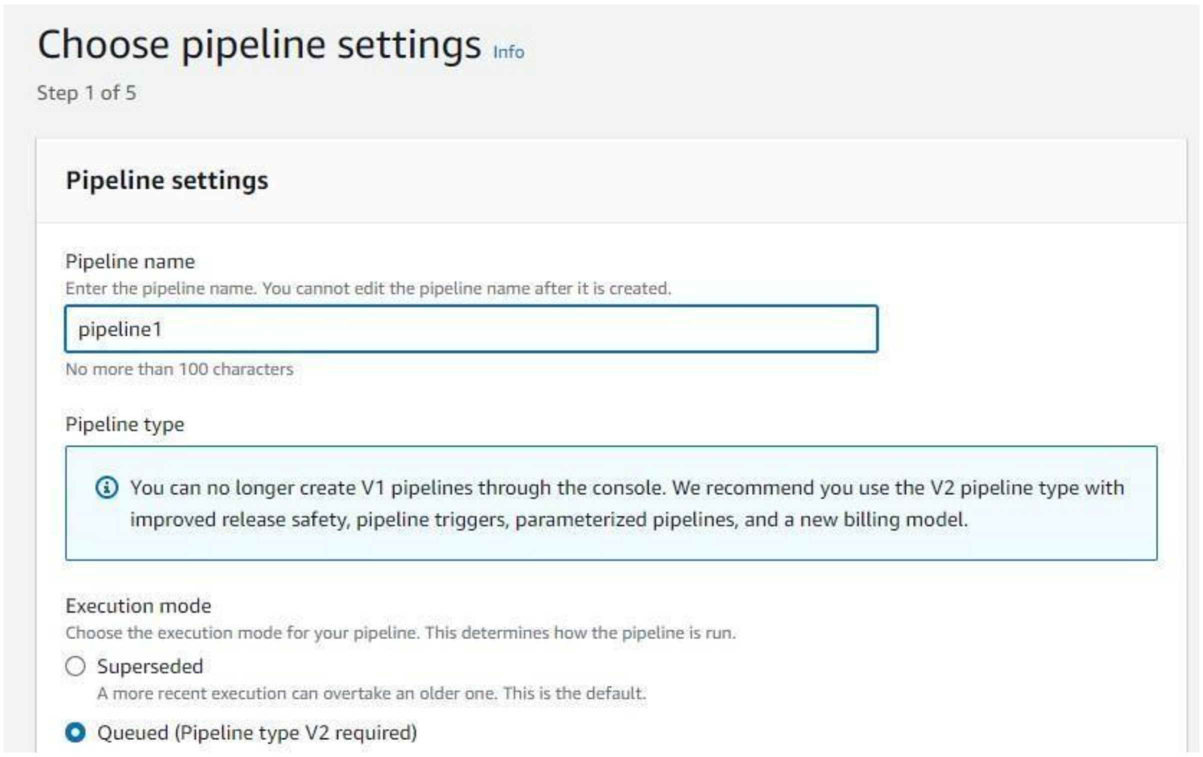
No results

There are no results to display.

https://us-east-1.console.aws.amazon.com/codesuite/code...

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 13: Name the pipeline, set the execution mode to Queued, and select New service role for the service role. AWS will generate a default role name.



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.

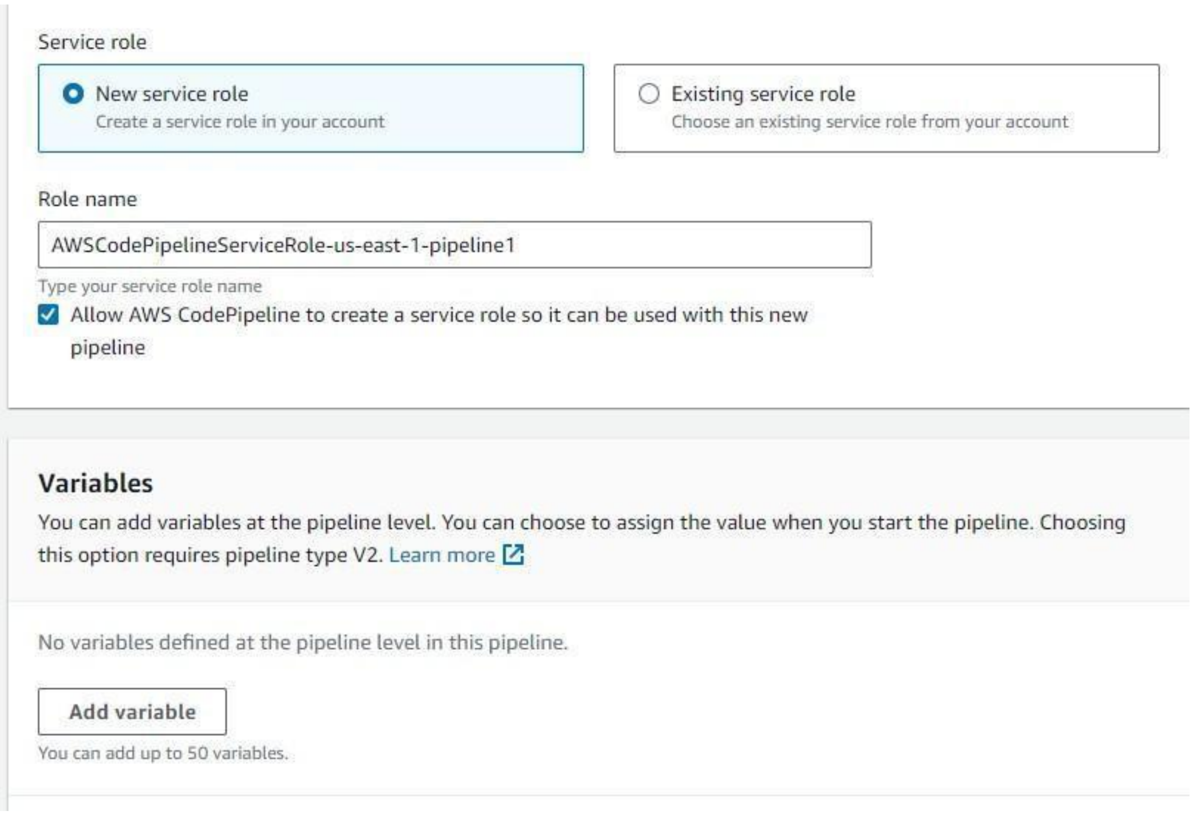
Execution mode

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)



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

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

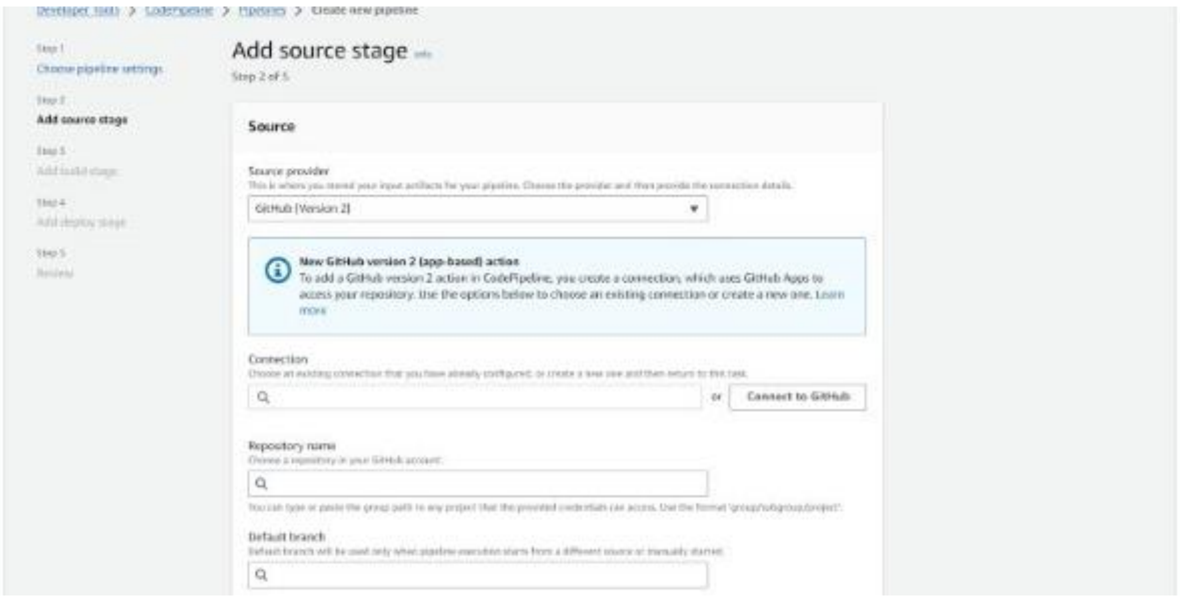
No variables defined at the pipeline level in this pipeline.

Add variable

You can add up to 50 variables.

Step 14: If specific variables are required for the pipeline, go to the Variables section and add them by clicking Add Variable.

Step 15: In the "Add Storage Stage" section, there is an option to connect the pipeline to GitHub. Click the "Connect to GitHub" button next to the connection field.



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 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)

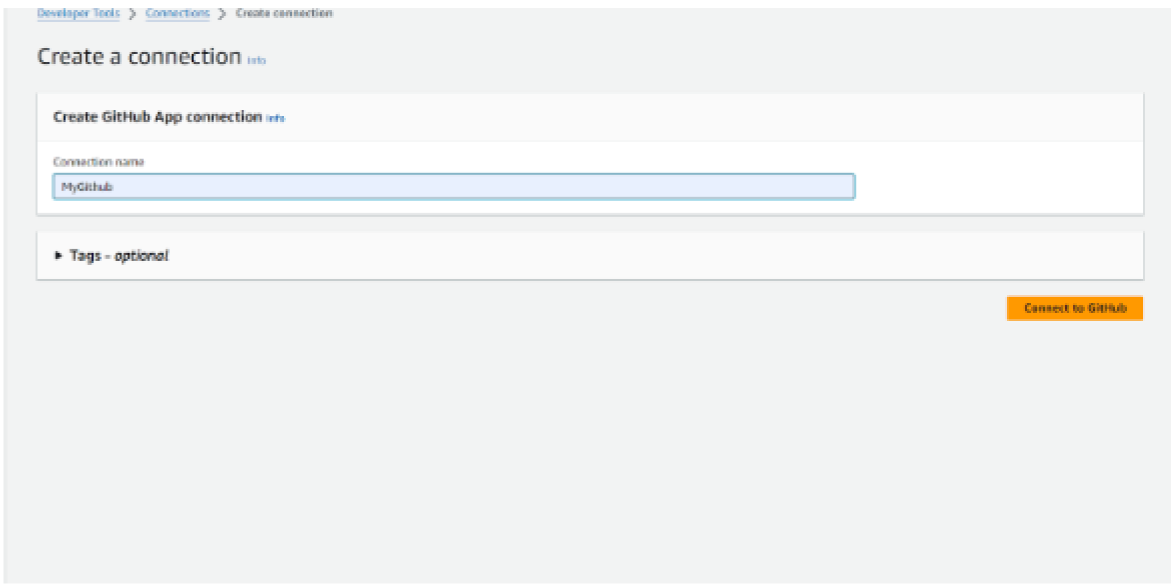
New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

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

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'.

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

Step 16: Enter a name for the connection when prompted, then click "Connect to GitHub" to link the pipeline with the GitHub account.



Developer Tools > Connections > Create connection

Create a connection Info

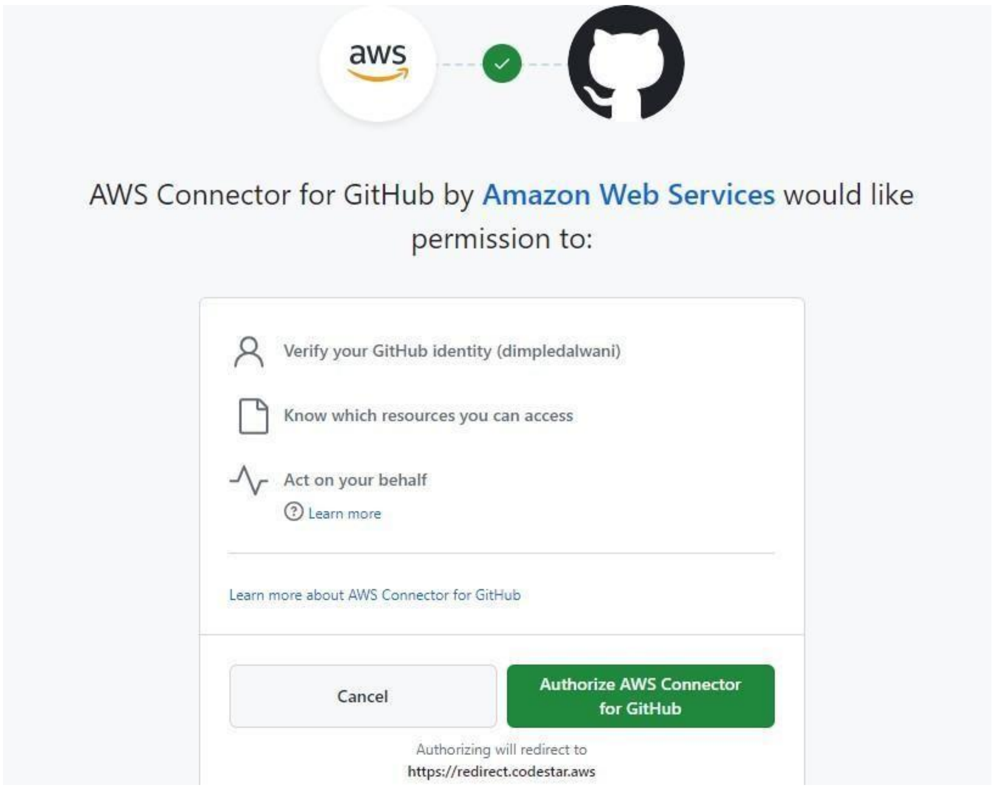
Create GitHub App connection Info

Connection name
MyGitHub

▸ Tags - *optional*

Connect to GitHub

Step 17: Authorization will be required on the next page. Click the "Authorize AWS Connector for GitHub" button to grant the necessary permissions.


AWS Connector for GitHub by Amazon Web Services would like permission to:

Verify your GitHub identity (dimpledalwani)

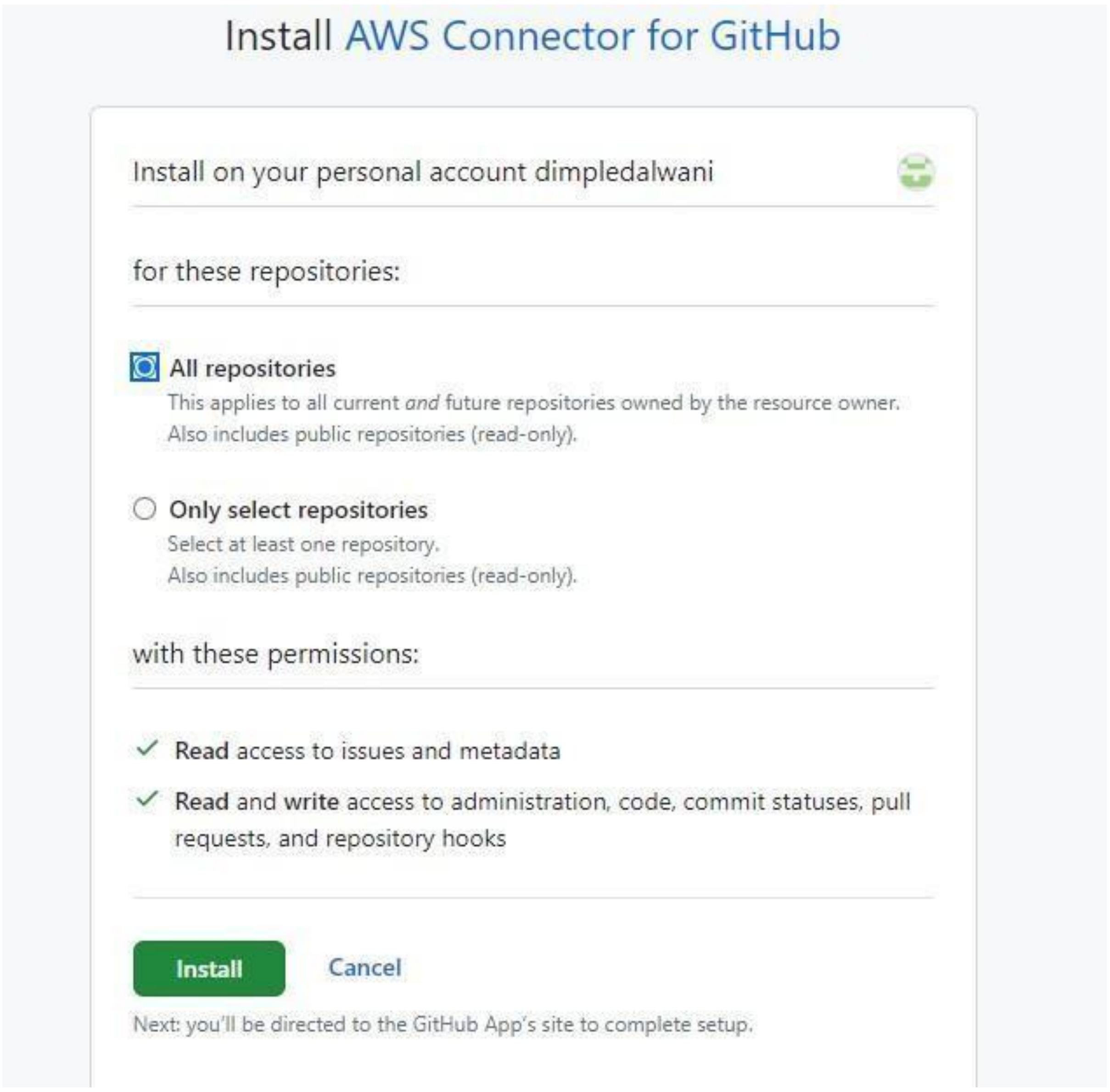
Know which resources you can access

Act on your behalf

Learn more
Learn more about AWS Connector for GitHub
Cancel
Authorize AWS Connector for GitHub

Authorizing will redirect to https://redirect.codestar.aws

Step 18: Select the "All repositories" option to grant access to all GitHub repositories, and click "Install" to finalize the connection.



Install AWS Connector for GitHub

Install on your personal account dimpledalwani

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).

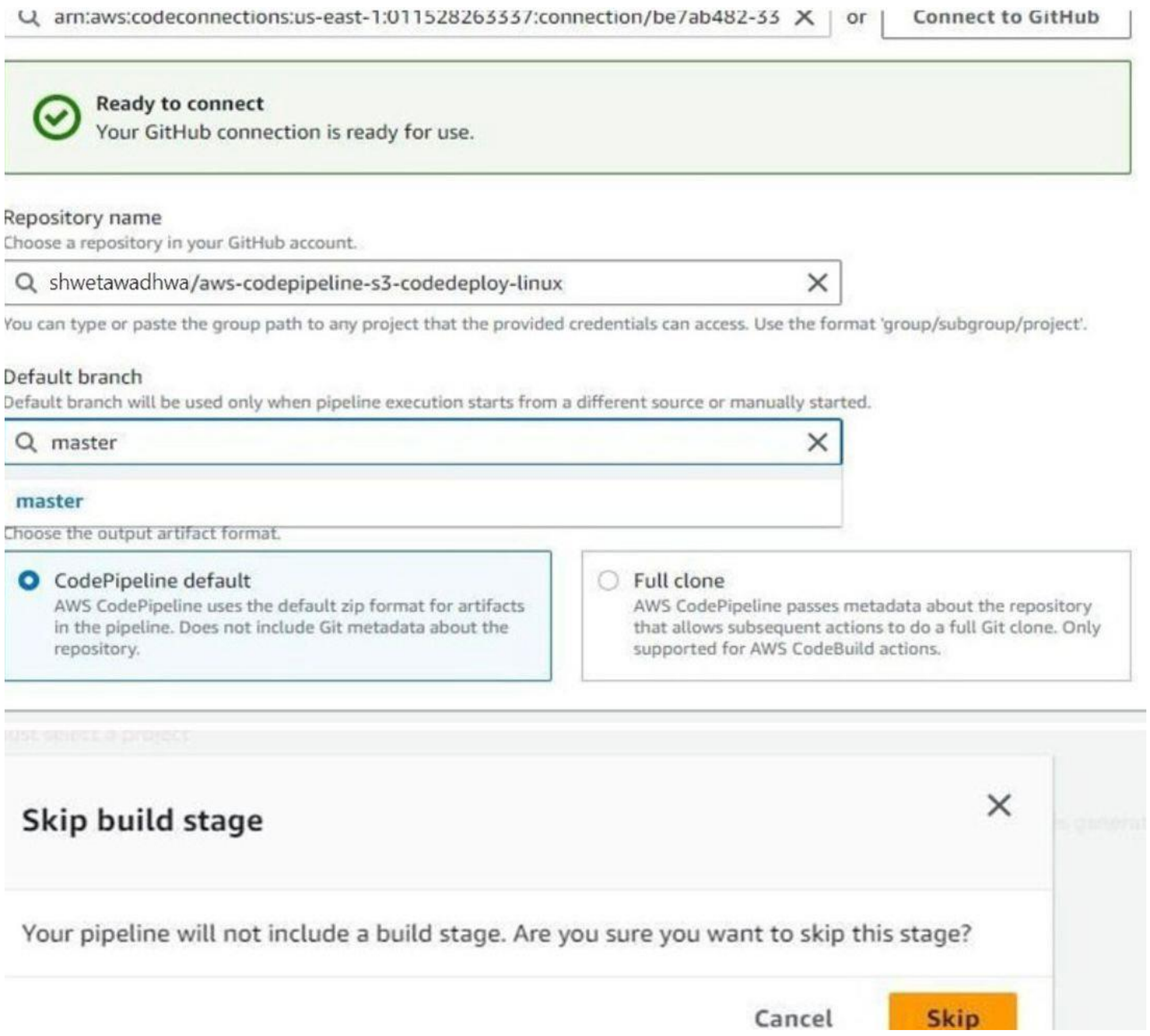
with these permissions:

✓ **Read** access to issues and metadata

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

Next: you'll be directed to the GitHub App's site to complete setup.

Step 19: In the pipeline setup, enter the name of the forked repository. Set the default branch to "master" and choose "CodePipeline default" as the configuration.


arn:aws:codeconnections:us-east-1:011528263337:connection/be7ab482-33 or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.
shwetawadhwa/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

master
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.

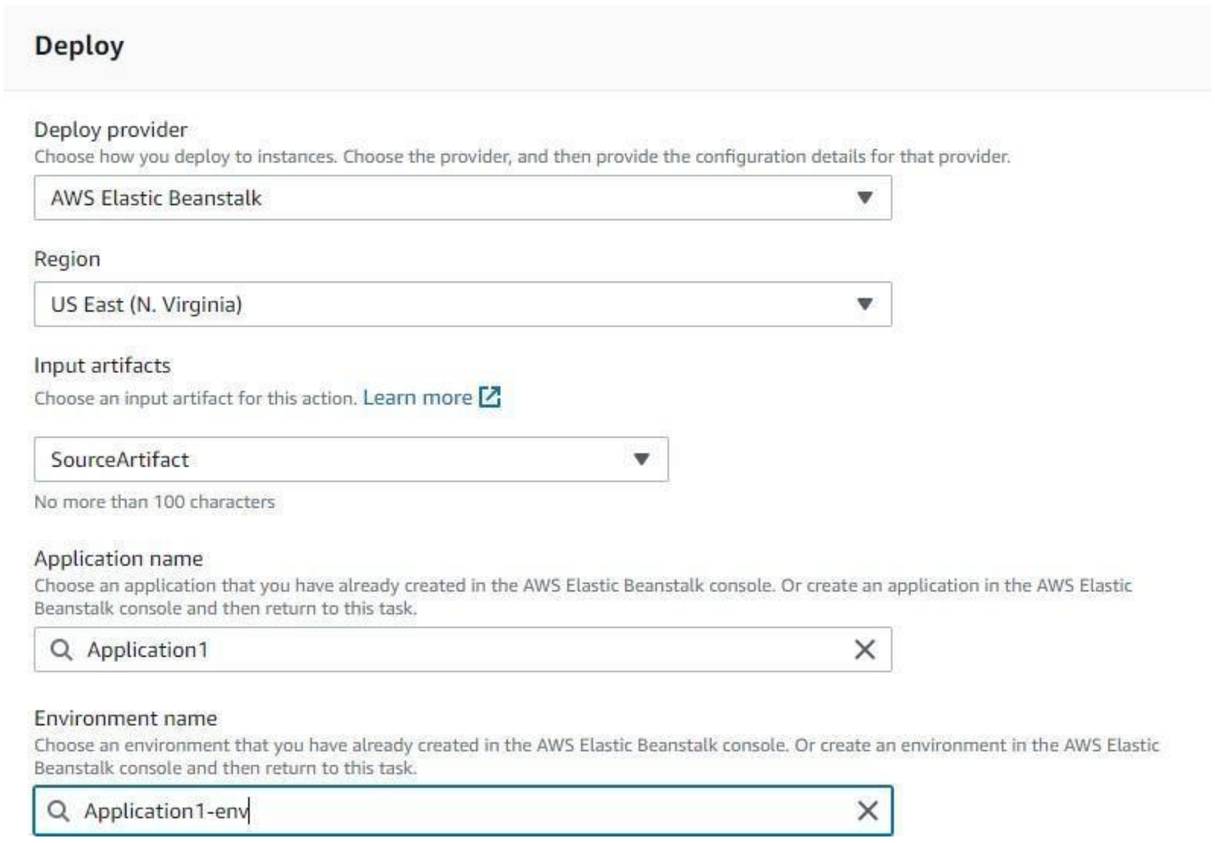
Skip build stage

Your pipeline will not include a build stage. Are you sure you want to skip this stage?

Cancel Skip

Step 20: The "Build" stage is optional and can be skipped if not needed.

Step 21: In the "Deploy" stage, enter the application name and environment name created earlier to link the pipeline to the application 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

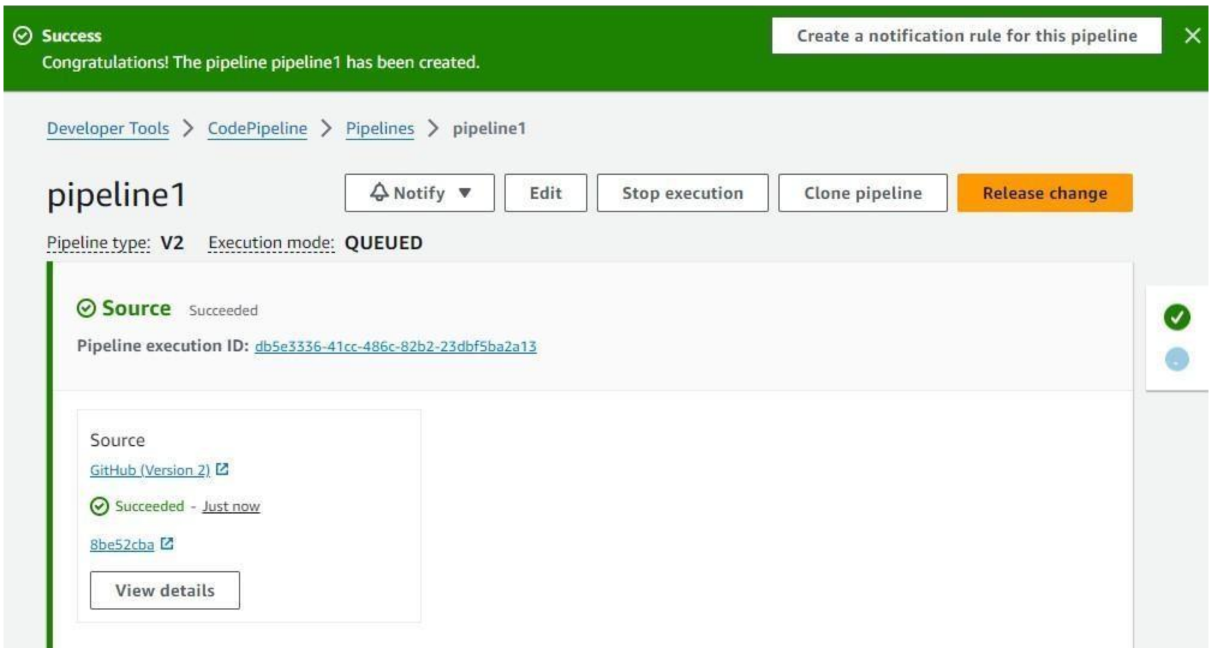
Region
US East (N. Virginia)

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.
Application1

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.
Application1-env

Step 22: The pipeline is successfully created.



Success
Congratulations! The pipeline pipeline1 has been created.

Create a notification rule for this pipeline
Developer Tools > CodePipeline > Pipelines > pipeline1
Notify
Edit
Stop execution
Clone pipeline
Release change

pipeline1

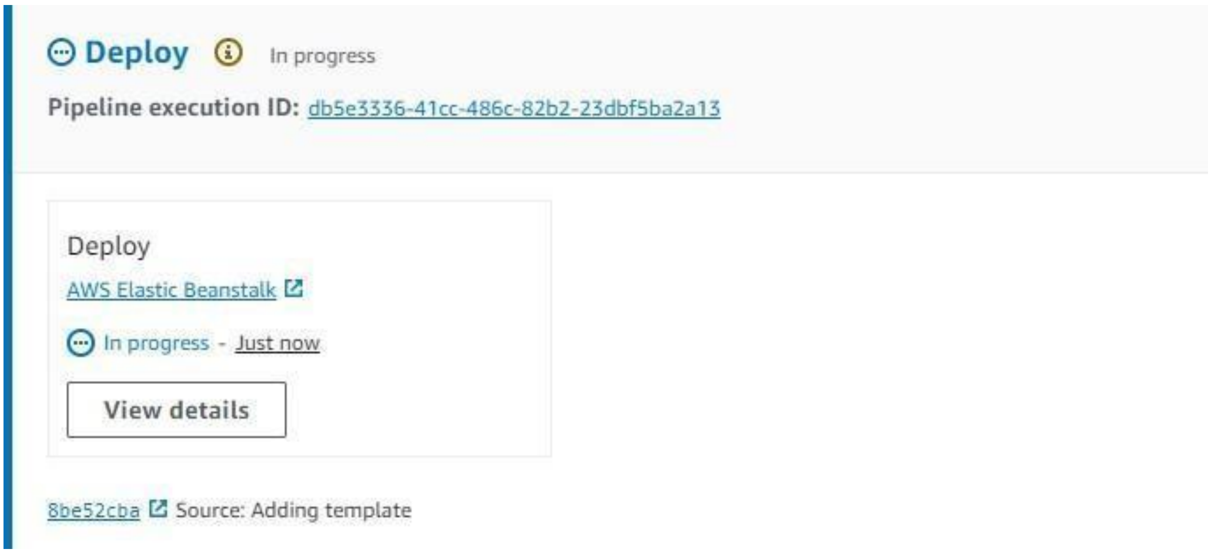
Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: db5e3336-41cc-486c-82b2-23dbf5ba2a13

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

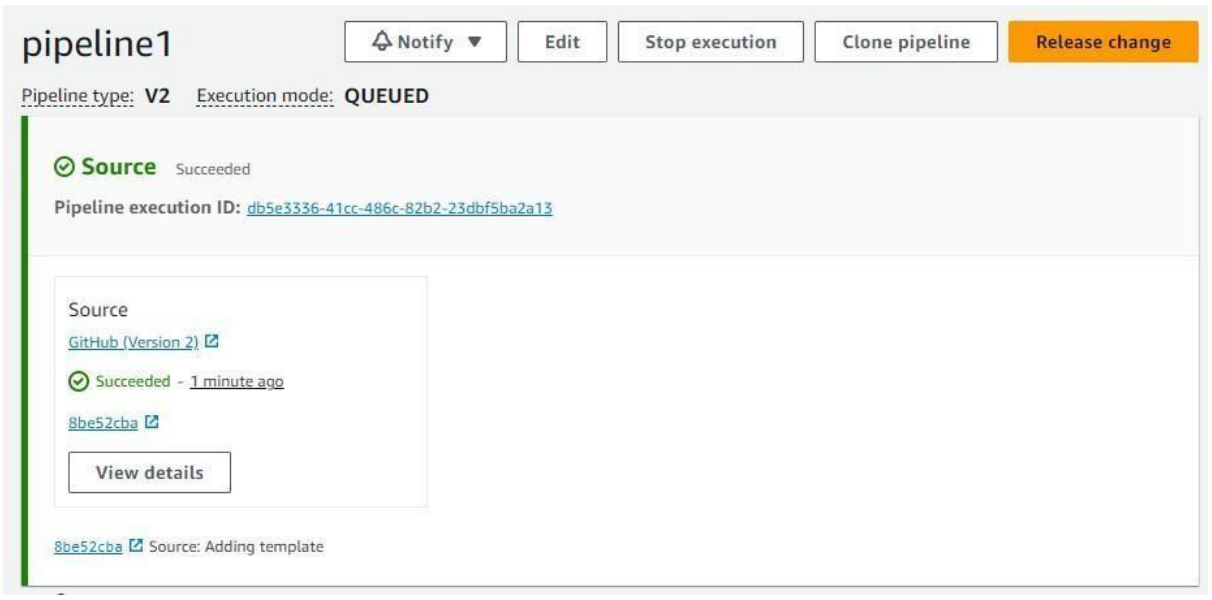
Step 23: Review the deployment details to ensure everything is set up correctly.


Deploy In progress
Pipeline execution ID: db5e3336-41cc-486c-82b2-23dbf5ba2a13

Deploy
AWS Elastic Beanstalk
In progress - Just now
View details

8be52cba Source: Adding template

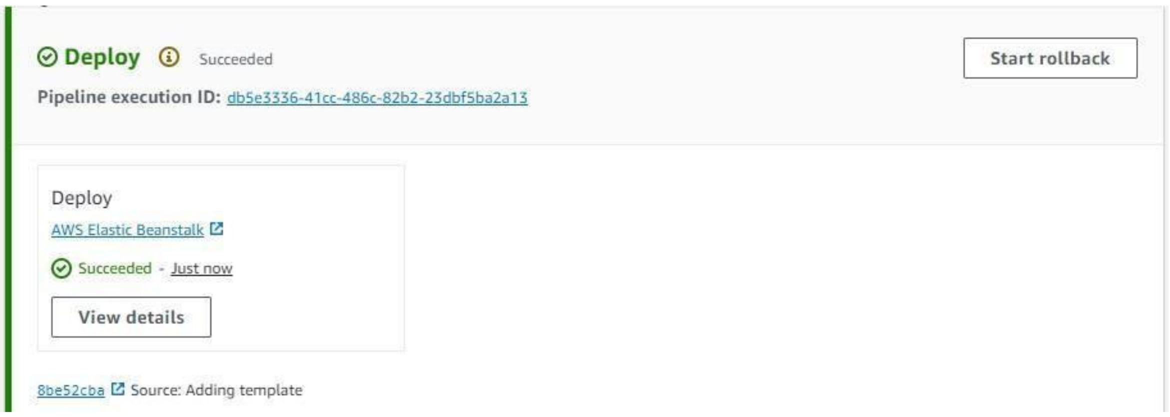
Step 24:The pipeline will be fully deployed.


pipeline1
Notify Edit Stop execution Clone pipeline Release change
Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: db5e3336-41cc-486c-82b2-23dbf5ba2a13

Source
GitHub (Version 2)
Succeeded - 1 minute ago
8be52cba
View details

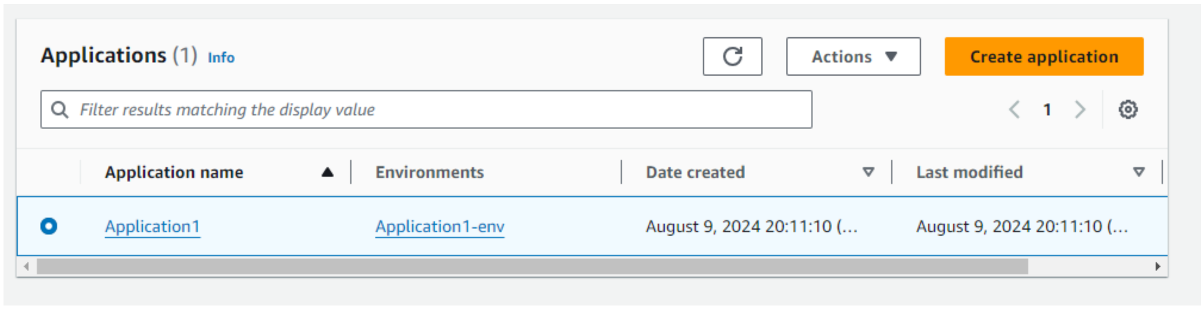
8be52cba Source: Adding template


Deploy Succeeded
Start rollback
Pipeline execution ID: db5e3336-41cc-486c-82b2-23dbf5ba2a13

Deploy
AWS Elastic Beanstalk
Succeeded - Just now
View details

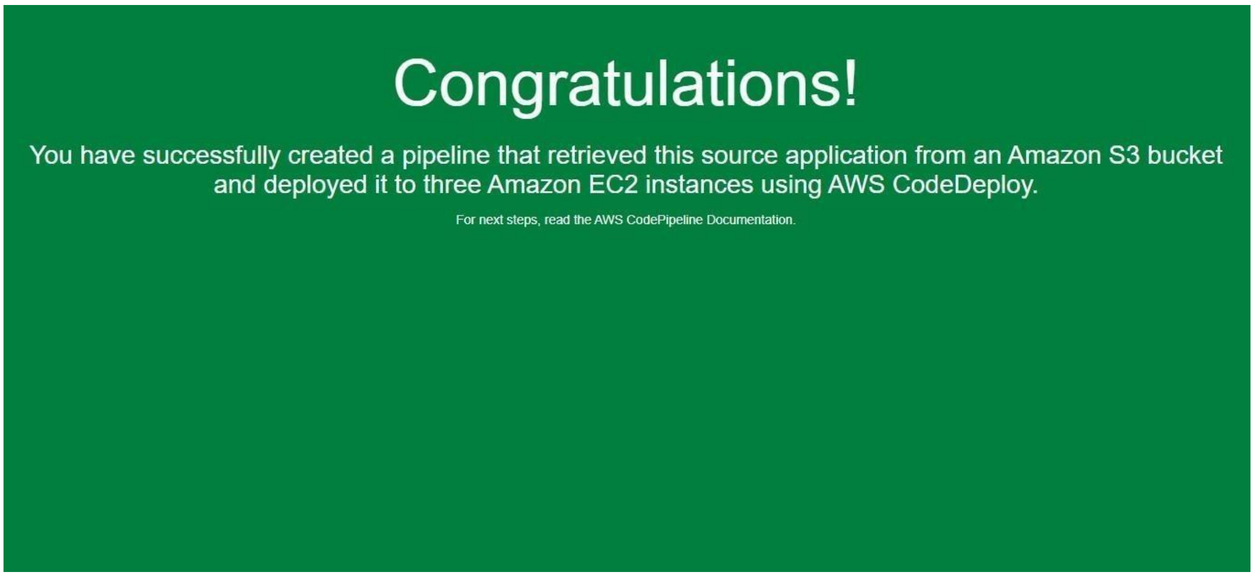
8be52cba Source: Adding template

Step 25: To view the application, go to the "Applications" section in the AWS console. Find the created application and click on the link in the environments section to access the live version.


Applications (1) Info

	Application name	Environments	Date created	Last modified
●	Application1	Application1-env	August 9, 2024 20:11:10 (...	August 9, 2024 20:11:10 (...

Step 26: The pipeline is successfully created, and the application is deployed and running as expected.


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.