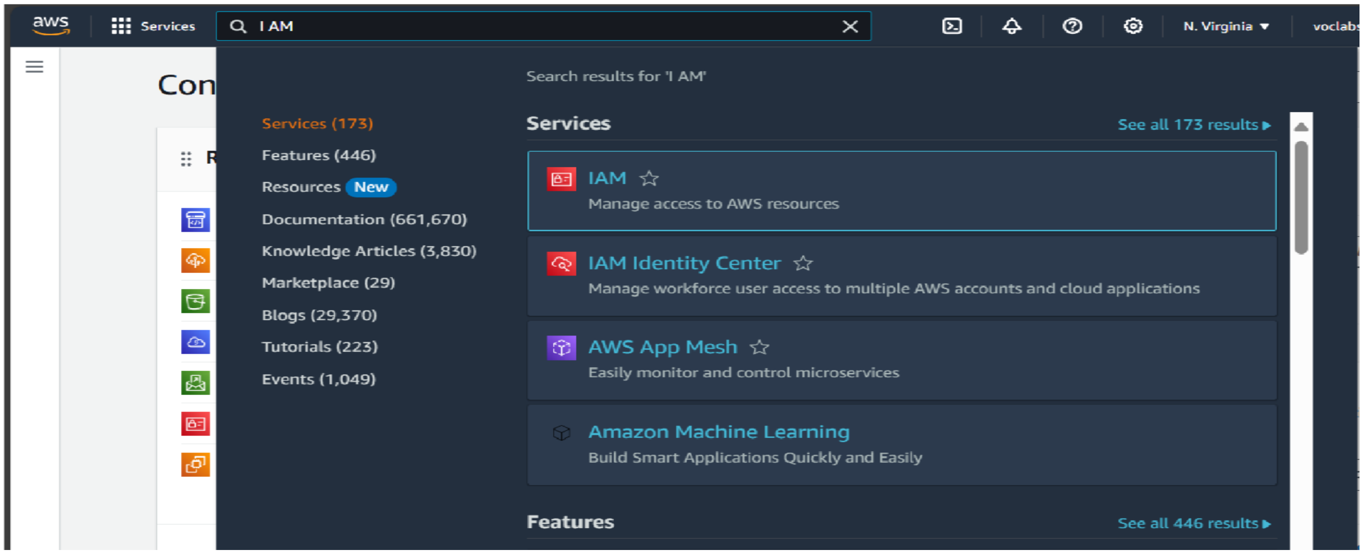
Name: Abhinav Swaminathan D15C Roll No: 01

Experiment No: 2

Step1 :- Creation of role:

1. Login to your AWS account and search for IAM



Search results for 'I AM'

Services (173)
Features (446)
Resources New
Documentation (661,670)
Knowledge Articles (3,830)
Marketplace (29)
Blogs (29,370)
Tutorials (223)
Events (1,049)

Services See all 173 results

IAM ☆
Manage access to AWS resources

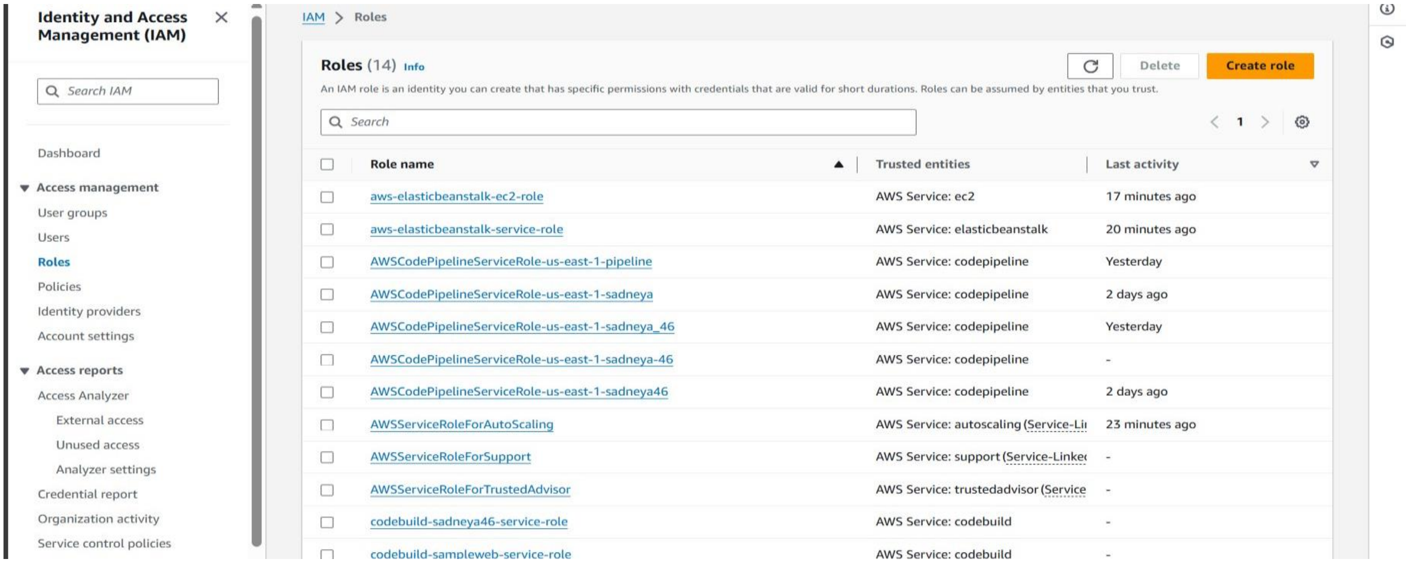
IAM Identity Center ☆
Manage workforce user access to multiple AWS accounts and cloud applications

AWS App Mesh ☆
Easily monitor and control microservices

Amazon Machine Learning
Build Smart Applications Quickly and Easily

Features See all 446 results

2. Then go into the role section and click on create role.

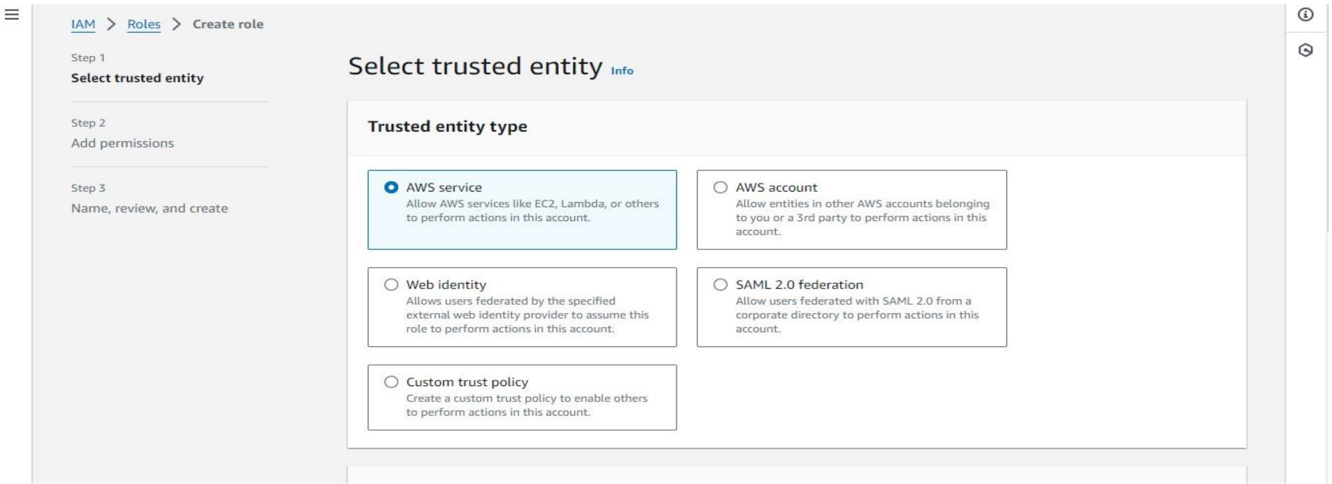


Roles (14) Info

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
aws-elasticbeanstalk-ec2-role	AWS Service: ec2	17 minutes ago
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	20 minutes ago
AWSCodePipelineServiceRole-us-east-1-pipeline	AWS Service: codepipeline	Yesterday
AWSCodePipelineServiceRole-us-east-1-sadneya	AWS Service: codepipeline	2 days ago
AWSCodePipelineServiceRole-us-east-1-sadneya_46	AWS Service: codepipeline	Yesterday
AWSCodePipelineServiceRole-us-east-1-sadneya-46	AWS Service: codepipeline	-
AWSCodePipelineServiceRole-us-east-1-sadneya46	AWS Service: codepipeline	2 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Li	23 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
codebuild-sadneya46-service-role	AWS Service: codebuild	-
codebuild-sampleweb-service-role	AWS Service: codebuild	-

3. Then select a trusted entity as AWS service.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

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

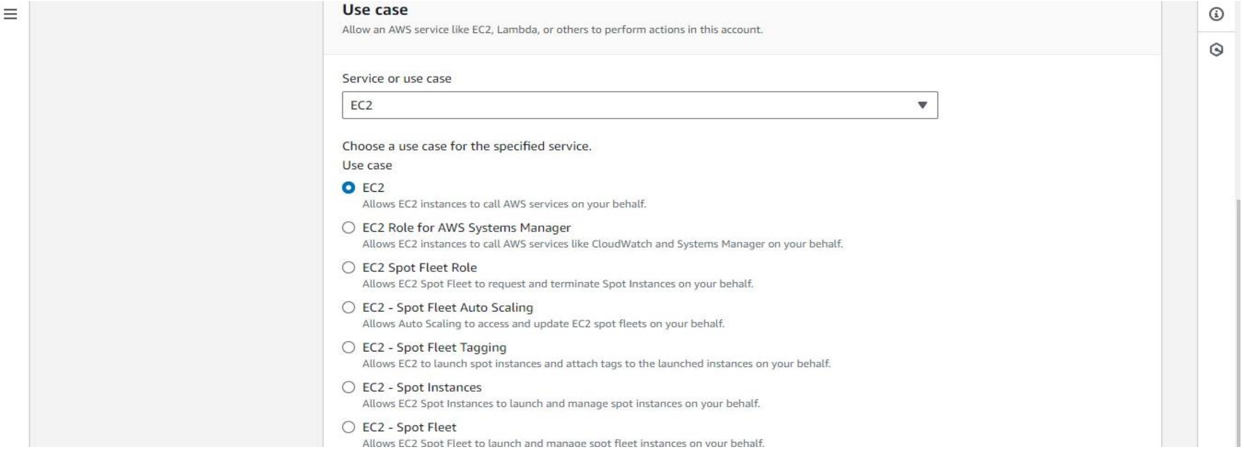
○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ 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.

4. Select use case as EC2.



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

● EC2
Allows EC2 instances to call AWS services on your behalf.

○ EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

○ EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

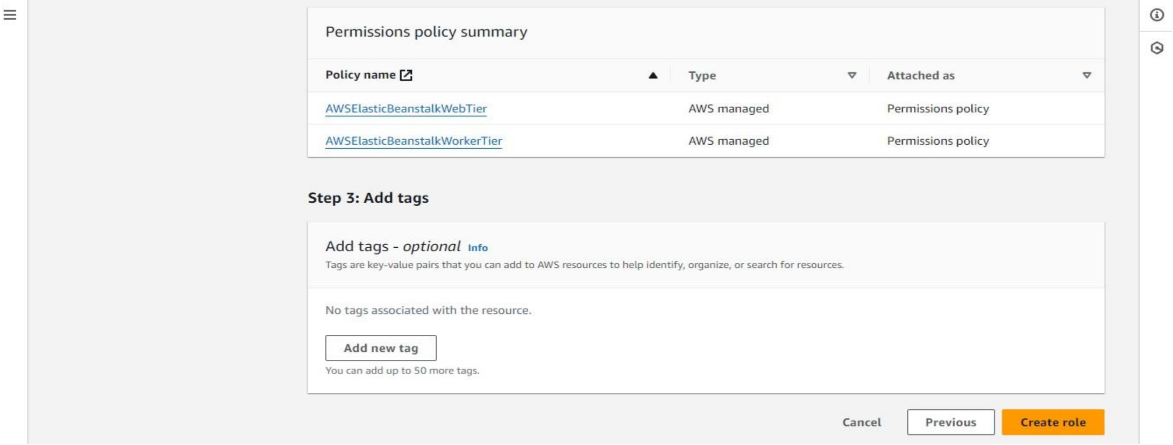
○ EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

○ EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

○ EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

○ EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

5. Select permissions as AWS Elastic Beanstalk Web Tier and AWS elastic Beanstalk worker tier.



Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

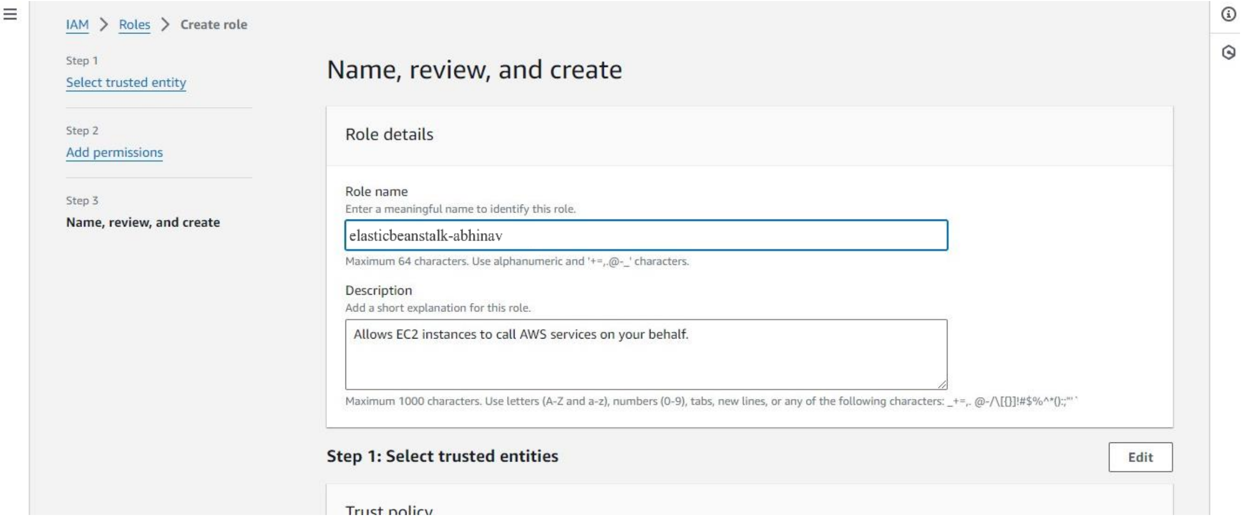
Step 3: Add tags

Add tags - *optional* Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

You can add up to 50 more tags.

6. Give a name to Role.


IAM > Roles > Create role

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Name, review, and create

Role details

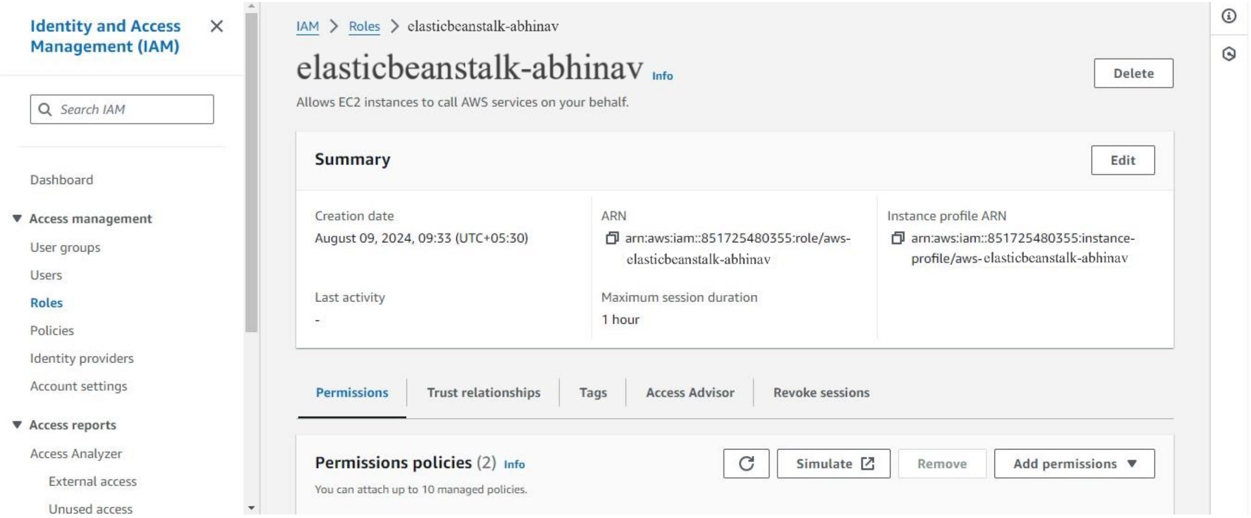
Role name
Enter a meaningful name to identify this role.
elasticbeanstalk-abhinav
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

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

Trust policy

7. Then the role gets created



IAM > Roles > elasticbeanstalk-abhinav

elasticbeanstalk-abhinav Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date: August 09, 2024, 09:33 (UTC+05:30)

ARN: arn:aws:iam::851725480355:role/aws-elasticbeanstalk-abhinav

Instance profile ARN: arn:aws:iam::851725480355:instance-profile/aws-elasticbeanstalk-abhinav

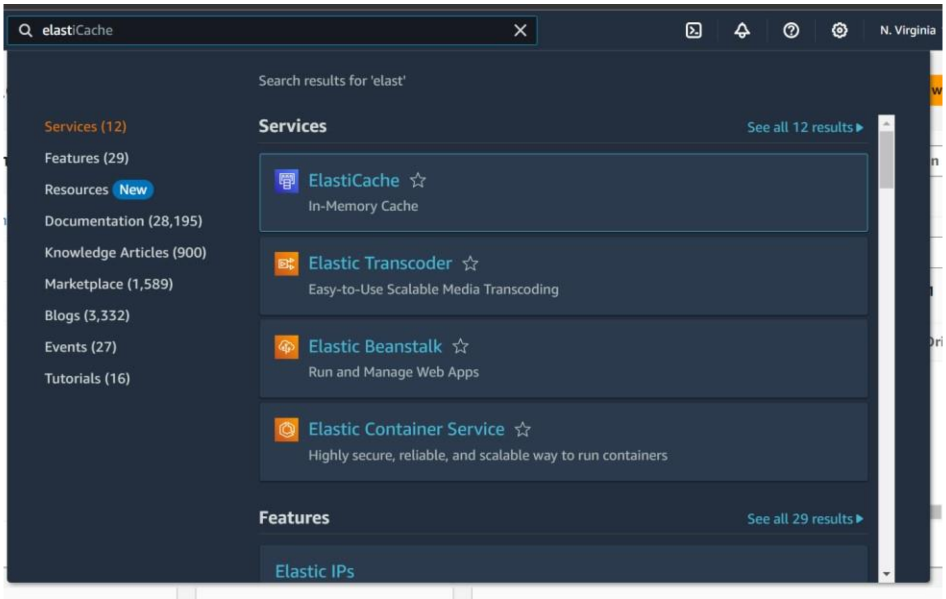
Last activity: -

Maximum session duration: 1 hour

Permissions policies (2) Info
You can attach up to 10 managed policies.

Step 2 :- Creation Elastic Beanstalk Environment

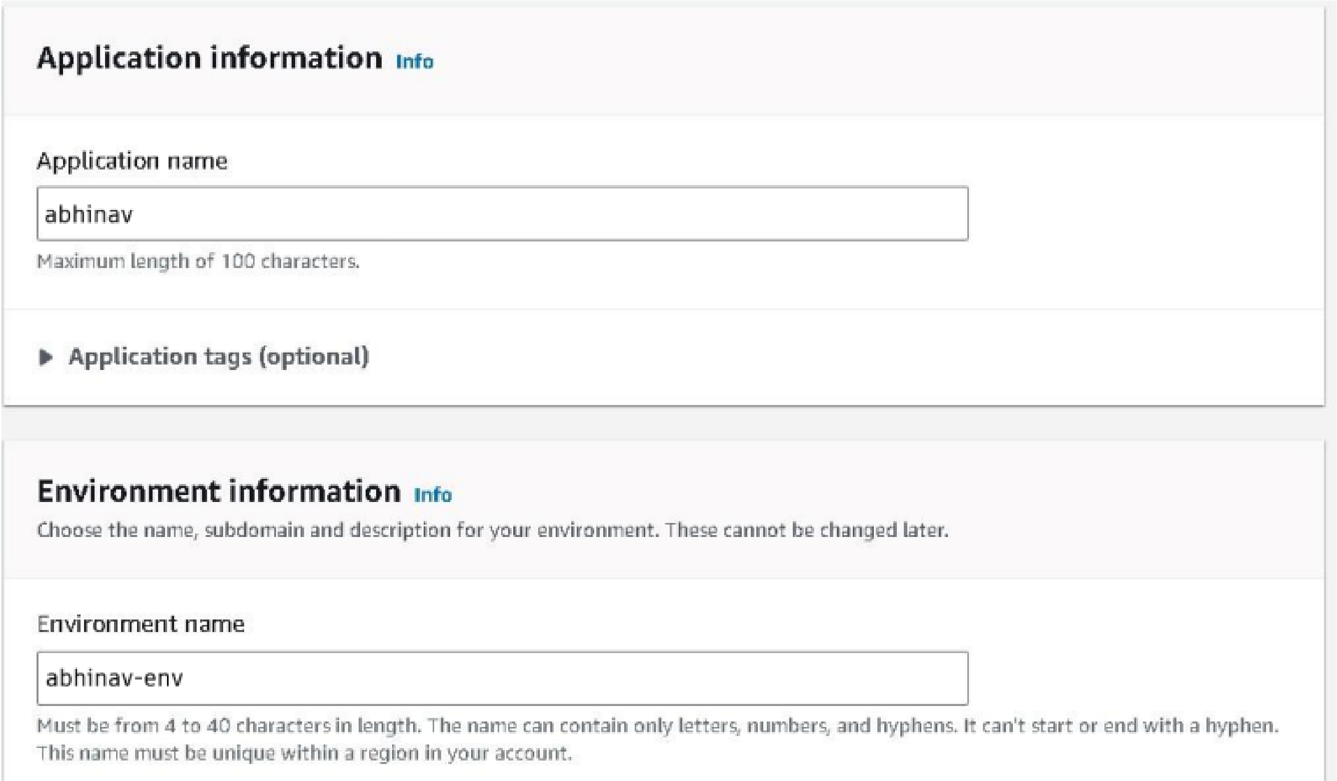
1. search for Elastic Beanstalk in the search box.



elastiCache

Search results for 'elast'

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



Application information Info

Application name

abhinav

Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

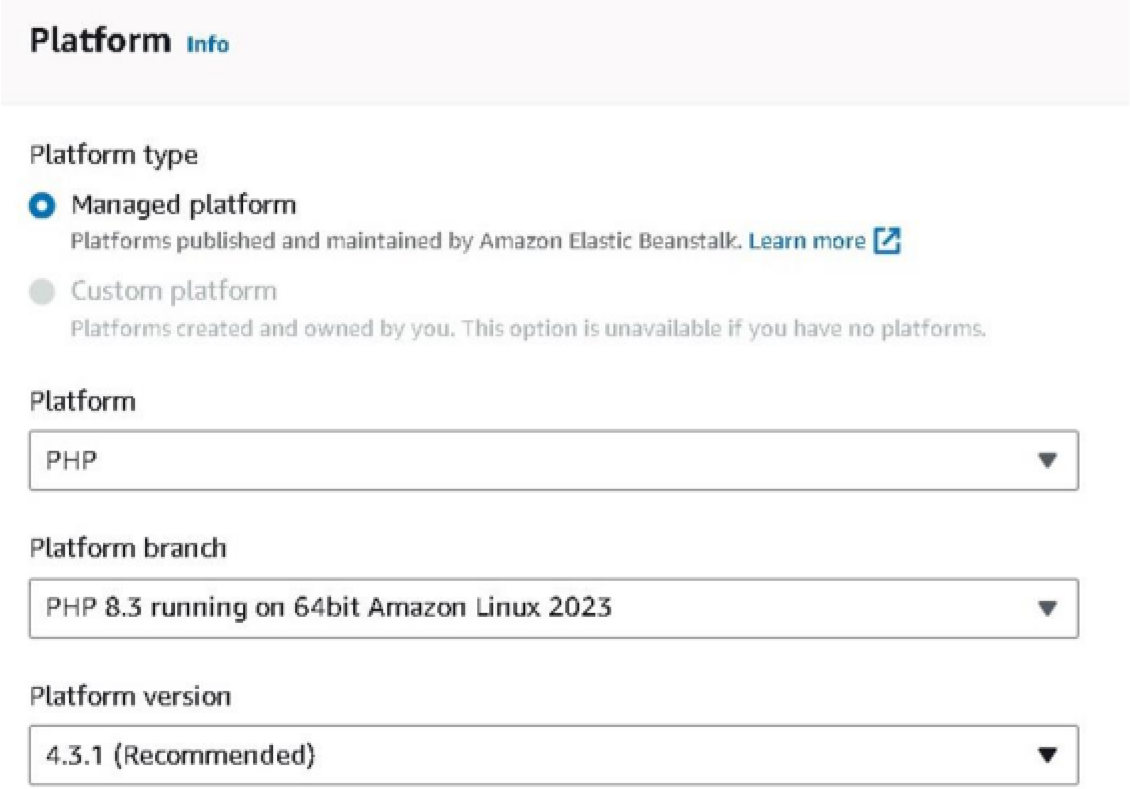
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

abhinav-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.

3. Select platform as PHP.



Platform

Platform type

● Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

○ Custom platform

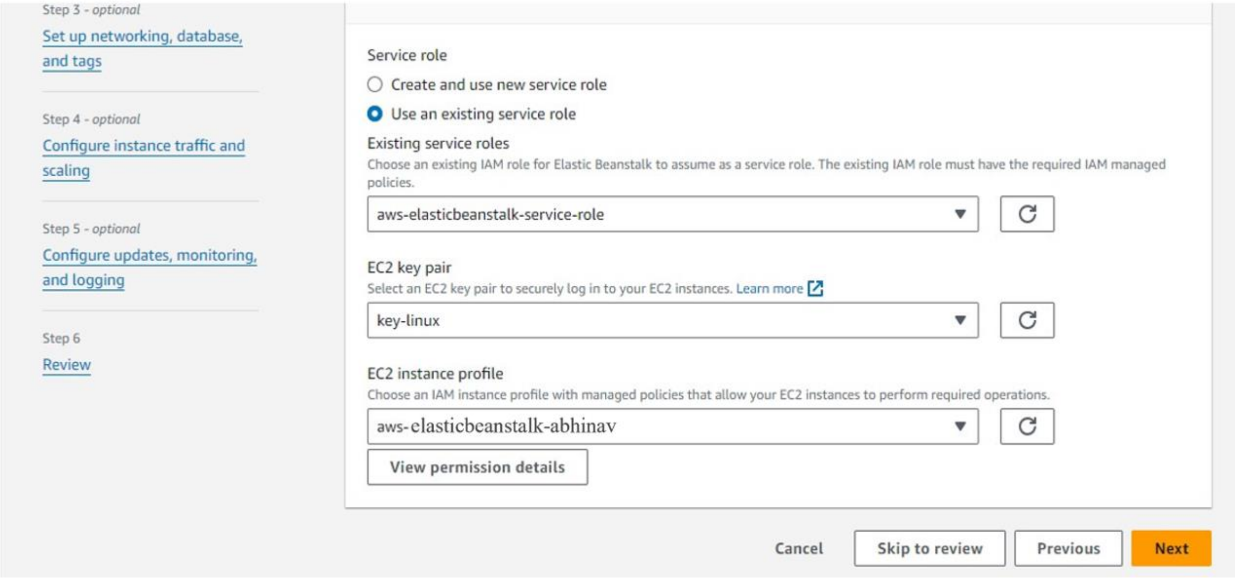
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform: PHP

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

Platform version: 4.3.1 (Recommended)

4. After clicking on next you need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Which we created in 1st part. Select role, then select key you have created then profile will be automatically selected according to role. then click on create application by keeping all the remaining settings as it is.



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 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.

aws-elasticbeanstalk-service-role

EC2 key pair

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

key-linux

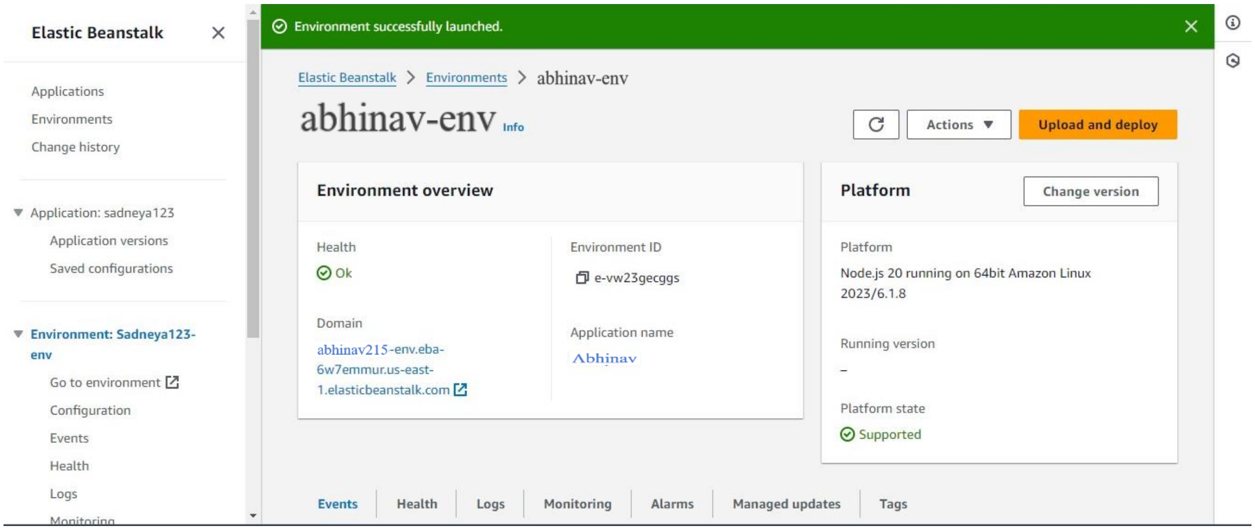
EC2 instance profile

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

aws-elasticbeanstalk-abhinav

Keep Set up networking, database and tags, Configure instance traffic and scaling, Configure updates, monitoring and logging all these default.

5. 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.



Environment successfully launched.

abhinav-env Info

Environment overview

Health: Ok

Environment ID: e-vw23gecggs

Domain: abhinav215-env.eba-6w7emmur.us-east-1.elasticbeanstalk.com

Application name: Abhinav

Platform

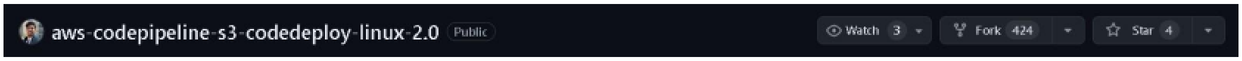
Platform: Node.js 20 running on 64bit Amazon Linux 2023/6.1.8

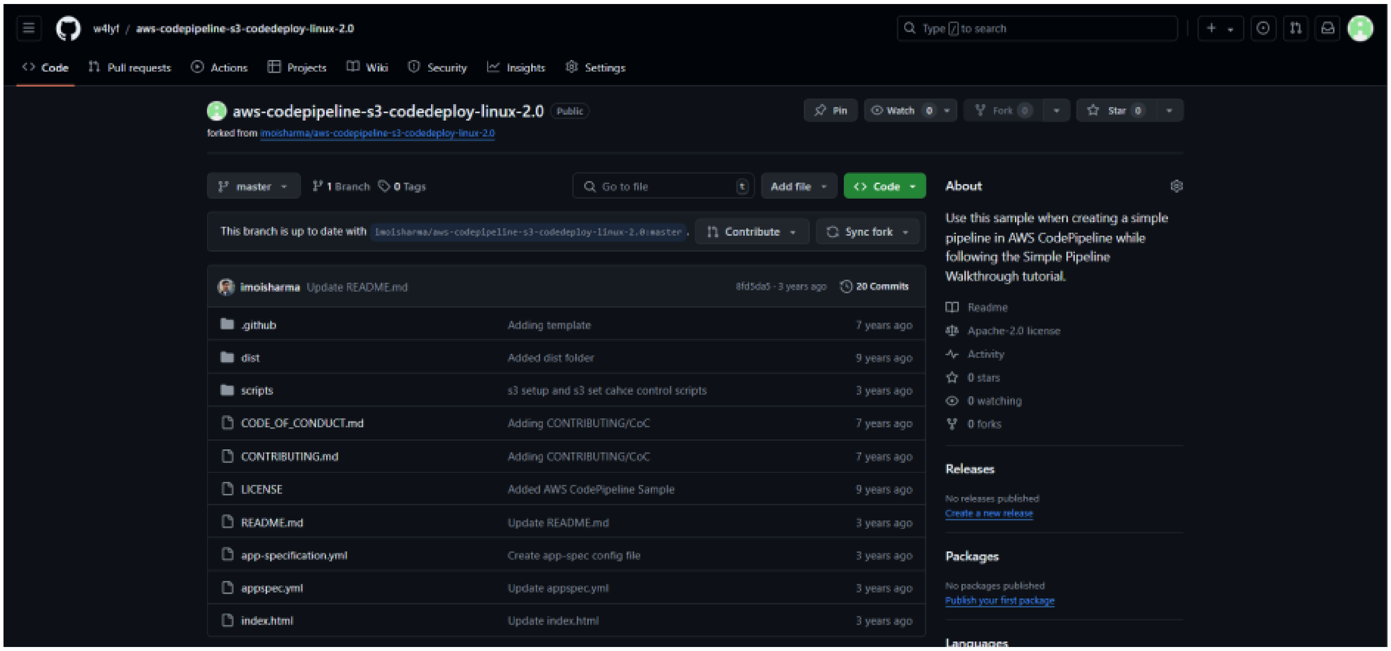
Running version: –

Platform state: Supported

Step 3: Get a copy of your sample code

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.
Go to the repository shared above and simply fork it.





forked from imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0

This branch is up to date with imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0:master .

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
CODE_OF_CONDUCT.md	Adding CONTRIBUTING/CoC	7 years ago
CONTRIBUTING.md	Adding CONTRIBUTING/CoC	7 years ago
LICENSE	Added AWS CodePipeline Sample	9 years ago
README.md	Update README.md	3 years ago
app-specification.yml	Create app-spec config file	3 years ago
appspec.yml	Update appspec.yml	3 years ago
index.html	Update index.html	3 years ago

About

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

Releases

No releases published

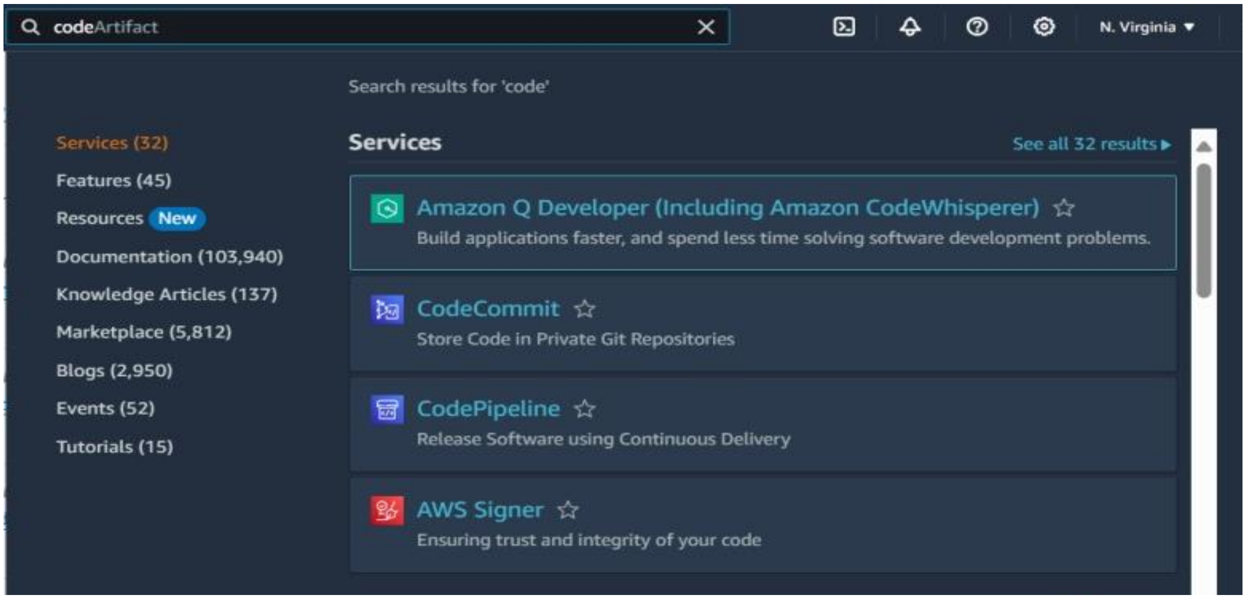
Packages

No packages published

Step 4: Creating a CodePipeline

In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.



codeArtifact

N. Virginia ▾

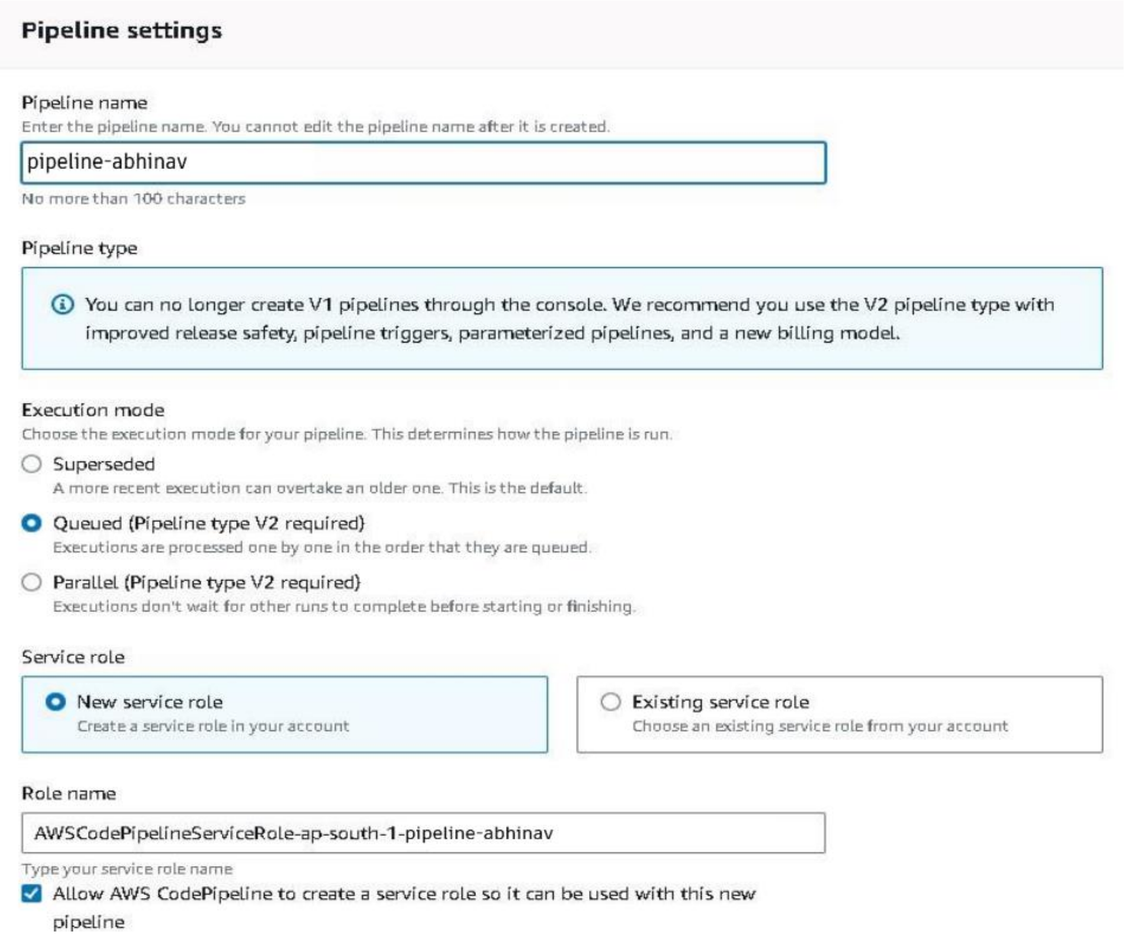
Search results for 'code'

- Services (32)
- Features (45)
- Resources New
- Documentation (103,940)
- Knowledge Articles (137)
- Marketplace (5,812)
- Blogs (2,950)
- Events (52)
- Tutorials (15)

Services — See all 32 results ▸

- Amazon Q Developer (Including Amazon CodeWhisperer) ☆ — Build applications faster, and spend less time solving software development problems.
- CodeCommit ☆ — Store Code in Private Git Repositories
- CodePipeline ☆ — Release Software using Continuous Delivery
- AWS Signer ☆ — Ensuring trust and integrity of your code

2. Give a name to your pipeline.



Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
pipeline-abhinav
No more than 100 characters

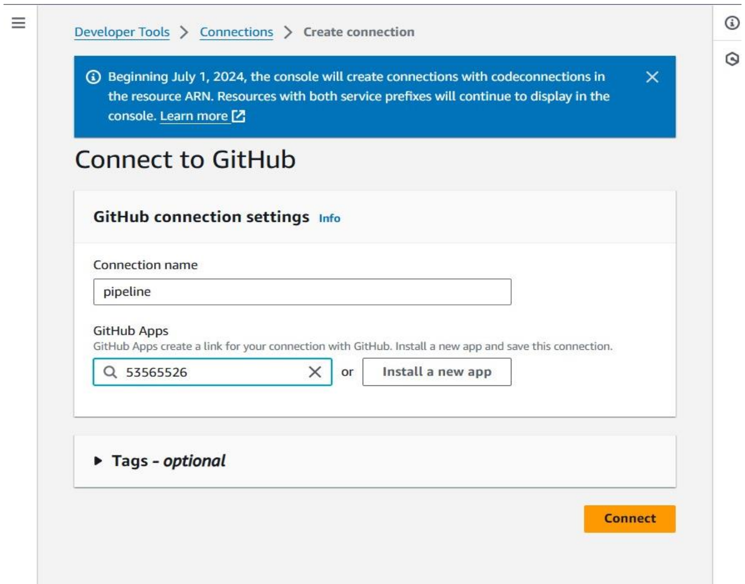
Pipeline type
ⓘ You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

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-ap-south-1-pipeline-abhinav
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

3. 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.



Developer Tools > Connections > Create connection

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

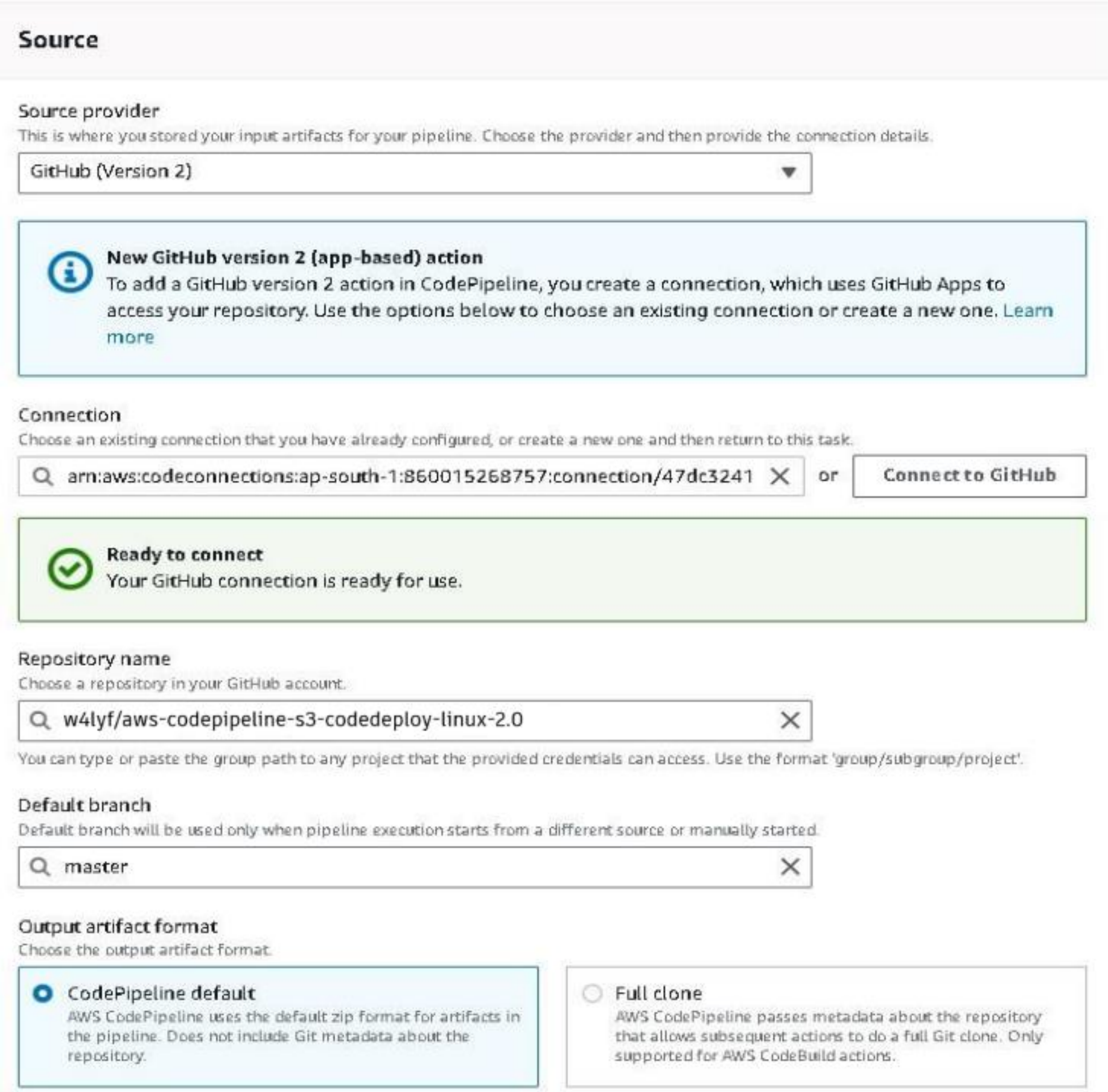
pipeline

GitHub Apps

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

53565526 or Install a new app

▸ Tags - *optional*



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:ap-south-1:860015268757:connection/47dc3241 or Connect to GitHub

Ready to connect
Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.

w4lyf/aws-codepipeline-s3-codedeploy-linux-2.0

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

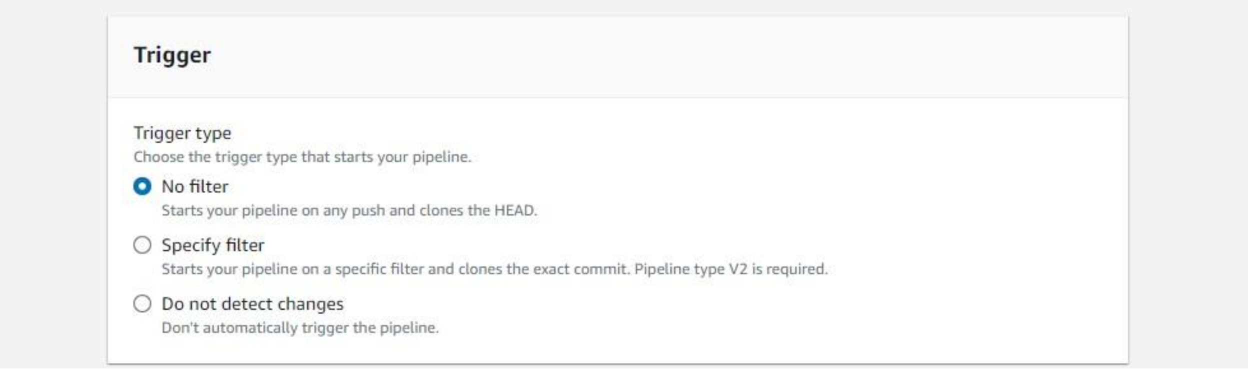
Output artifact format

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.

4. Then select trigger type none.



Trigger

Trigger type

Choose the trigger type that starts your pipeline.

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

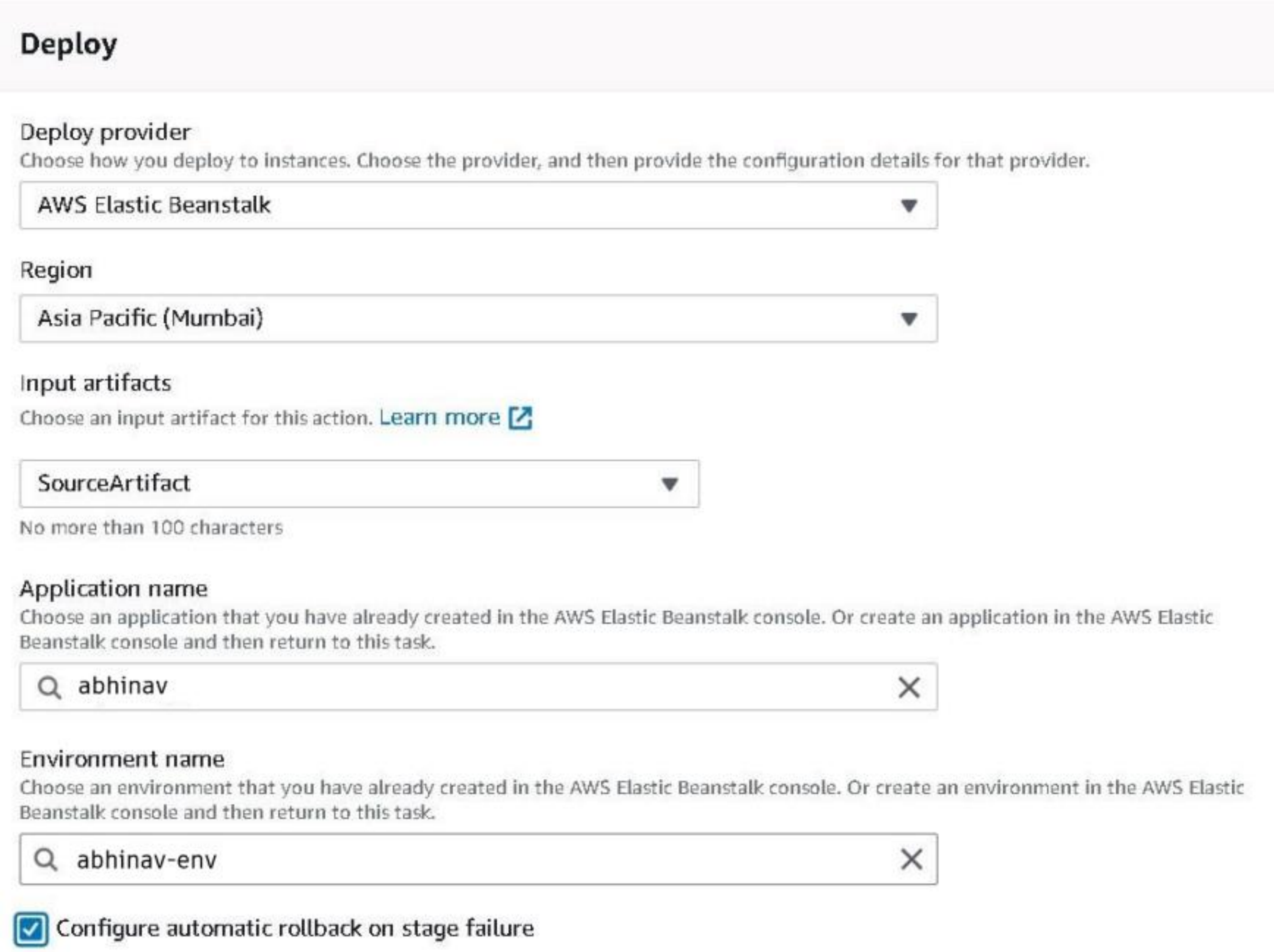
○ Specify filter
Starts your pipeline on a specific filter and clones the exact commit. Pipeline type V2 is required.

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

After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 5: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.



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

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.

abhinav

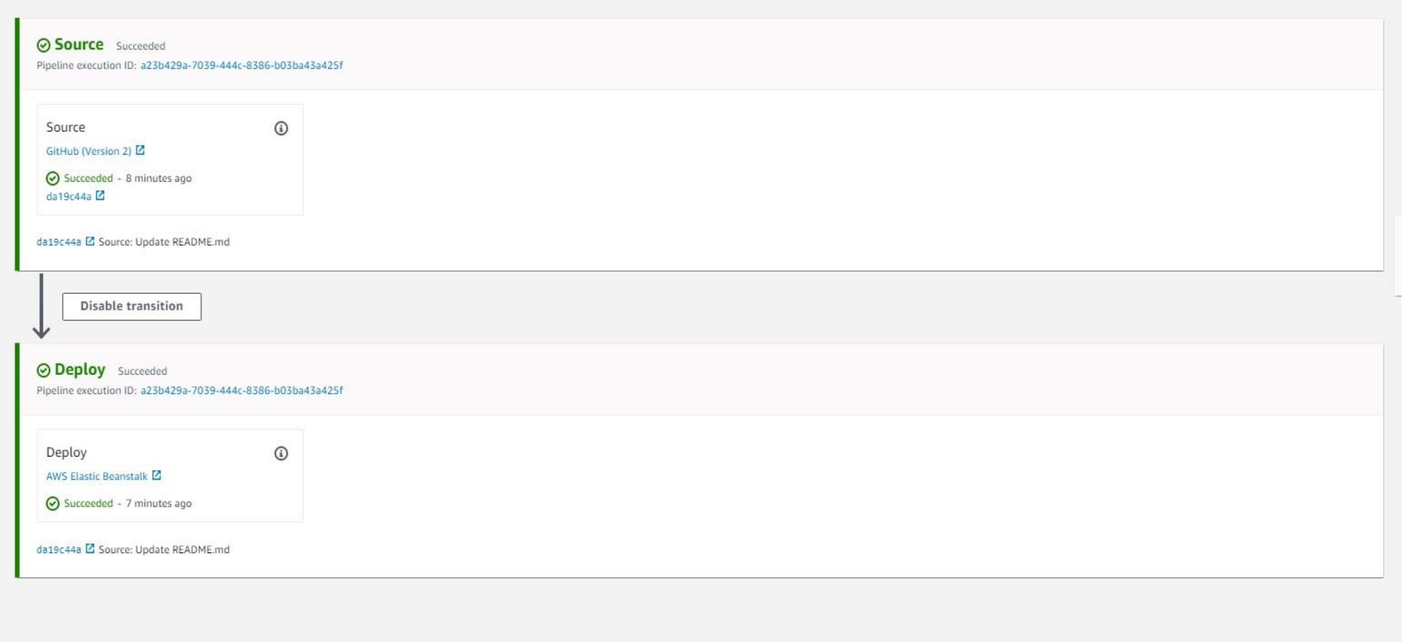
Environment name

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.

abhinav-env

☑ Configure automatic rollback on stage failure

2. Then it will give you this result on screen. i.e. deployed successfully.



Source Succeeded
Pipeline execution ID: a23b429a-7039-444c-8386-b03ba43a425f

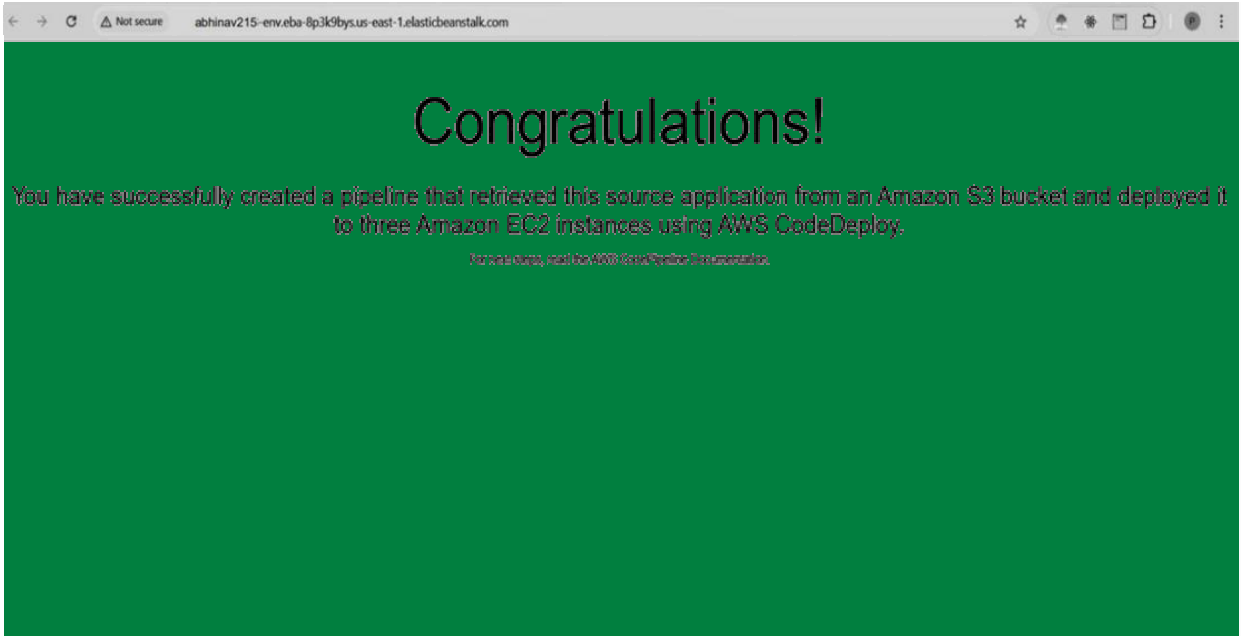
Source
GitHub (Version 2)
Succeeded - 8 minutes ago
da19c44a

da19c44a Source: Update README.md

Deploy Succeeded
Pipeline execution ID: a23b429a-7039-444c-8386-b03ba43a425f

Deploy
AWS Elastic Beanstalk
Succeeded - 7 minutes ago

da19c44a Source: Update README.md



Congratulations!

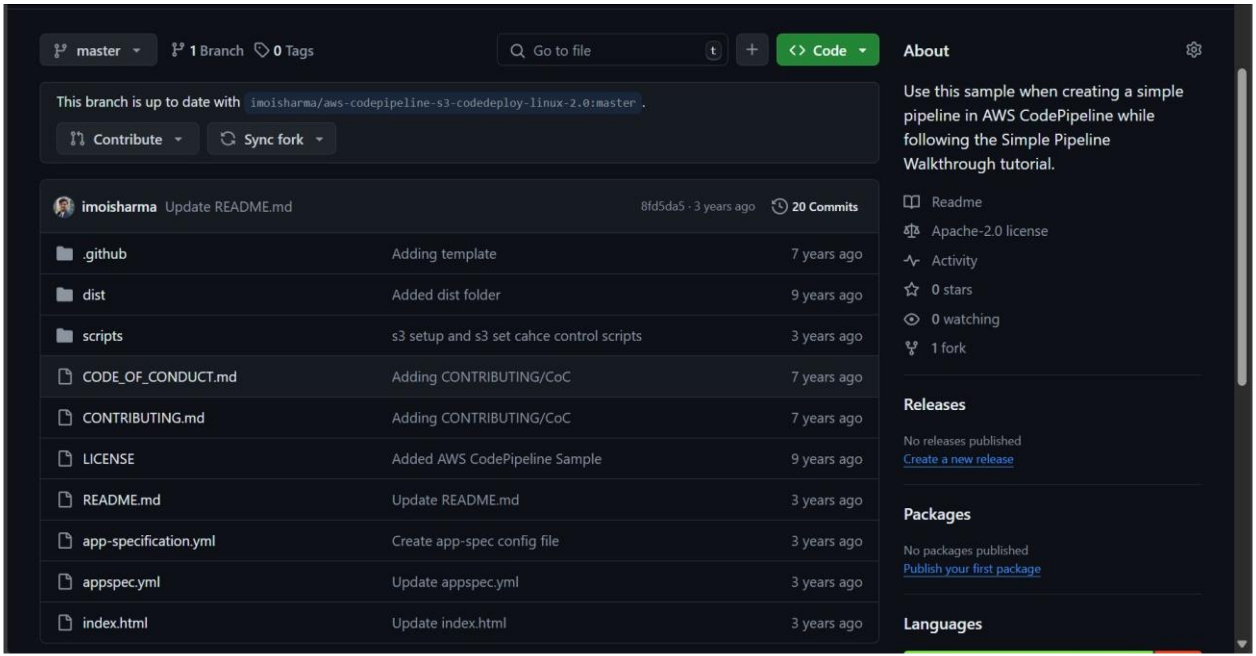
You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation.

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

Step 6: Committing changes to update app

1. In this we make some changes in the file. Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.



This branch is up to date with `imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0:master` .

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
CODE_OF_CONDUCT.md	Adding CONTRIBUTING/CoC	7 years ago
CONTRIBUTING.md	Adding CONTRIBUTING/CoC	7 years ago
LICENSE	Added AWS CodePipeline Sample	9 years ago
README.md	Update README.md	3 years ago
app-specification.yml	Create app-spec config file	3 years ago
appspec.yml	Update appspec.yml	3 years ago
index.html	Update index.html	3 years ago

About

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

Readme
Apache-2.0 license
Activity
0 stars
0 watching
1 fork

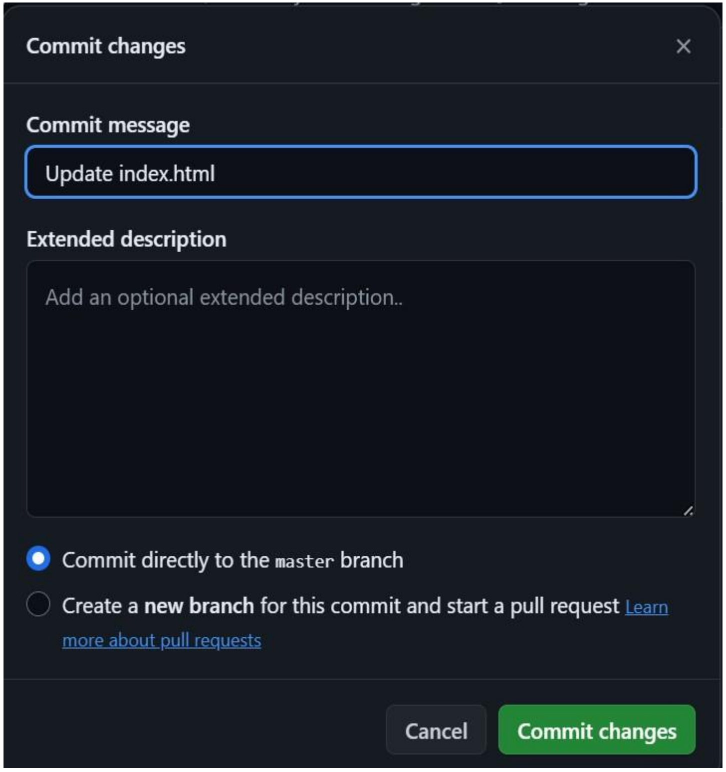
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages



Commit changes

Commit message

Update index.html

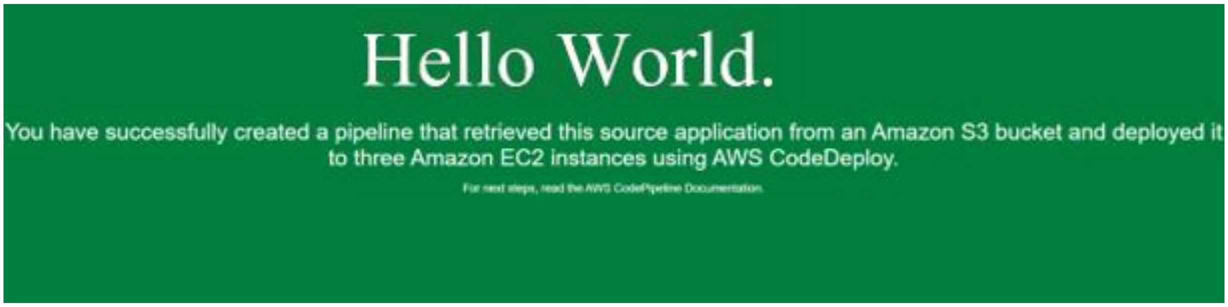
Extended description

Add an optional extended description..

● Commit directly to the `master` branch

○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

2. Then again start the deployment of the pipeline. And check the changes in the website



Hello World.

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.